Proceedings of the International Workshop on Juris-Informatics 2020 (JURISIN 2020)

in association with
the 12th JSAI International Symposia on AI (JSAI-isAI 2020)

JURISIN 2020 Co-chairs

Nguyen Le Minh, Japan Advanced Institute of Science and Technology, Japan
Yasuhiro Ogawa, Nagoya University, Japan
Ken Satoh, National Institute of Informatics, Japan

November 16-17, 2020

Preface

This volume contains the papers presented at JURISIN2020: Fourteenth International Workshop on Juris-informatics held on November 16-17, 2020 on line. Juris-informatics is a new research area which studies legal issues from the perspective of informatics. The purpose of this workshop is to discuss both the fundamental and practical issues among people from the various backgrounds such as law, social science, information and intelligent technology, logic and philosophy, including the conventional AI and law area. JURISIN 2020 is the 14th International Workshop on Juris-informatics, which is held in association with the Twelfth International Symposia on AI by Japanese Society ofArtificial Intelligence (JSAI-isAI). This year, we have twenty-four submissions and each paper was reviewed by three program committee members and as a result, all of them were accepted for a presentation but three papers were withdrawn due to the double acceptance for JURISIN 2020 and JURIX 2020. Papers cover various field of juris-informatics such as logical inference, legal documents processing and legal issues of applications of AI. In JURISIN2020, we have special sessions on Seventh Competition on Legal Information Extraction/Entailment (COLIEE-2020). The motivation for the competition is to help create a research community of practice for the capture and use of legal information. Furthermore, as invited speakers, we have Akira Shimazu from Japan Advanced Institute of Science and Technology (JAIST) and Takehiko Kasamatsu from Toin University of Yokohama, Japan. Finally, we would like to thank all the authors who submitted papers and the members of PC for reviewing the submitted papers.

November 16-17, 2020
JURISIN2020 Co-chairs
Nguyen Le Minh, Japan Advanced Institute of Science and Technology, Japan
Yasuhiro Ogawa, Nagoya University, Japan
Ken Satoh, National Institute of Informatics, Japan

Program Committee

JURISIN Program Committee

Michał Araszkiewicz	Jagiellonian University
Ryuta Arisaka	Nagoya Institute of Technology
Marina De Vos	University of Bath
Juergen Dix	Clausthal University of Technology
Randy Goebel	University of Alberta
Guido Governatori	CSIRO
Tokuyasu Kakuta	Chuo University
Yoshinobu Kano	Shizuoka University
Takehiko Kasahara	Toin Univ. of Yokohama
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Sabrina Kirrane	Vienna University of Economics and Business - WU Wien
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Makoto Nakamura	Niigata Institute of Technology
Yoshiaki Nishigai	Chiba University
Tomoumi Nishimura	Osaka University
Katsumi Nitta	National Institute of Advanced Indutrial Science and Technology
Yasuhiro Ogawa	Nagoya University
Monica Palmirani	CIRSFID
Ginevra Peruginelli	ITTIG-CNR
Juliano Rabelo	AMII
Seiichiro Sakurai	Meiji Gakuin University
Ken Satoh	National Institute of Informatics and Sokendai
Akira Shimazu	JAIST
Kazuko Takahashi	Kwansei Gakuin University
Satoshi Tojo	JAIST
Katsuhiko Toyama	Nagoya University
Serena Villata	CNRS - Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis
Yueh-Hsuan Weng	Tohoku University
Masaharu Yoshioka	Hokkaido University
Thomas Ågotnes	University of Bergen

COLIEE Program Committee

Randy Goebel	University of Alberta
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
María Navas-Loro	Universidad Politécnica de Madrid
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Juliano Rabelo	AMII
Julien Rossi	Amsterdam Business School
Ken Satoh	National Institute of Informatics and Sokendai
Jaromir Savelka	Carnegie Mellon University
Yunqiu Shao	Tsinghua University
Akira Shimazu	JAIST
Takenobu Tokunaga	Tokyo Institute of Technology
Satoshi Tojo	JAIST
Vu Tran	Japan Advanced Institute of Science and Technology
Josef Valvoda	University of Cambridge
Serena Villata	CNRS - Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis
Hannes Westermann	University of Montreal
Hiroaki Yamada	Tokyo Institute of Technology
Masaharu Yoshioka	Hokkaido University

Table of Contents

Invited Talks

JURISIN 2020 papers

COLIEE 2020 papers

AI and Judicial Policy

Takehiko Kasahara[1]

[1] Toin University of Yokohama, Japan
kasahara@toin.ac.jp

Abstract: This paper discusses the impact of AI on judicial policy. Regarding the use of AI in the judiciary, it is limited to the use of IT and later ICT, and the system that AI itself judges has just begun to sprout. In this paper, first the situation to date of judicial ICT use in Japan, then the trends in the world are summarized, and furthermore its possibilities beyond the actual use.are discussed, Introduction

1. Preface

The JURISIN of this year is held online caused by the Corona epidemic. Even as I write this manuscript, the number of victims is still increasing. It goes without saying that this corona pandemic is a disaster, but it revealed the delay in digitization in Japan. It compelled remote work and digitization of our society. The "paper and stamp" culture in Japan has historically played a major role, but at the same time it was a factor that hindered digitization. I hope that it will not end with primary and emergency measures, but will lead to the acceleration of the delayed progress of ICT.

In this paper I will figure out the situation of Japan and furthermore the possibility of AI in the field of judicial policy.

2. Application of IT to Court

2.1 Video Link System, Triophone

It was often described as "the introduction of IT in Japanese judiciary is 20 years behind" and I wrote it, but in fact, the introduction of a part of IT was early: Video Link System and the Triophone.

2.1.1 Remote witness cross-examination by video link system

For a civil trial in 1998 the ISDN video conferencing system is introduced, nationwide in 50 District Courts and their major branches (video link). As a result, regarding witness cross-examination, if you live far from the court where the case is heard, you can go to the nearby district court to carry out the witness cross-examination procedure (Civil Procedure Act (hereinafter referred to CPA) Article 204) . The purpose was for witnesses involved in the trial to save time and expense and as a result, to make a trial "more convenient and speedy".

It was also introduced in 2001 in a criminal trial. In order to separate the accused and the victim especially by a sexual crime, who could feel strong mental pressure by the presence of the accused in a same court room. It replaces the conventional wooden screen.

2.1.2 Preparatory procedure by Triophone

At the same time, Triophone, a conference call system that uses a three-way calling system, has also been introduced. It was used for the preparatory proceedings; expressed on statute, "the method which enables court and both parties to communicate simultaneously by transmitting and receiving voice (CPA §170 (3)" . Initially it was of limited use by the preparatory proceedings, but it was expanded to include "the withdrawing an action, waiving a claim" by the plaintiff and acknowledging a claim by the defendant, and also the entering into a settlement.

2.2 Judicial system reform - "the Supreme Court to judge the memories of the case."

Due to administrative reforms in the 1990s in Japan, administrative restrictions as pre-dispute adjustment function were relaxed, and as the result, judicial trials as a post-dispute resolution function became the focus of attention .

"The Supreme Court, which judges the memories of the case" said former Prime Minister Koizumi who was skilled in catchphrase. He made fun of trial which costs too much cost and too long time.

The Act on Court Acceleration (Law No. 107 on July 16, 2003) was enforced in 2003 with the goal of making it possible to end all cases of first instance in the shortest possible period of time within two years. [1].

In this judicial system reformof June 12rth, 2001, the “Judicial Reform Council” have issued the Judicial Reform Promotion Plan, then, under the government “Judicial Reform Promotion Headquarters” was placed and on the the March 19rh, 2002, the headquarters and the Japan Bar Association issued the judicial system reform plan , in the next day from the Supreme Court the “Outline of the Jjudicial Reform Promotion plan” was issued. Regarding the introduction of IT, the Supreme Court said, "By promoting networking using homepages, etc., necessary measures should be taken to draw up and publish a plan to strengthen the provision of information including ADR, legal counseling, and legal assistance systems, In cooperation with related organizations, necessary measures will be taken, and promoted the active introduction of information and communication technology (IT) in various aspects such as court proceedings, paperwork, and information provision".

Various reforms such as the introduction of the lay judge trial system and the legal consulting centre, ”law terrace”, were made. Regarding the introduction of IT, the court pages have been enhanced (case law disclosure, statistics disclosure, demand procedure (dunning procedure) online system, etc.), but the trial procedure itself has been slightly digitized.

Although the provision for the e-filing to the court was subscribed (CPA §132-10), the corresponding order was not enacted. In addition, some petitions are allowed

online but the most important documents, complaint, written answer or a brief detailing an allegation had not been recognized for this purpose. Only a trial experiment was made at the Sapporo District Court in Hokkaido but completed on the March 20th, 2008.

The introduction of IT in courts has not progressed at all in almost 20 years since this reform.[2]

2.3 Headquarters for Japan's Economic Revitalization

The current civil trial reform of ICT in civil courts is based on the efforts of the government and not of the coarts. The "Japan revival strategy -Japan is Back - (2013)" has aimed to developed countries within 3 places in the 35 OECD members countries in the World Bank's business environment ranking in 2020. But Japan's ranking is declining year by year. Conflict resolution was cited as one of the areas with low evaluation.[3] In order to reform this situation two Cabinet decisions have been made led by “the Headquarters for Japan's Economic Revitalization”.

June 9th, 2017, "Report on Priority Measures and Others for Innovative Business Activity Action Plan 2017"[4] was issued, and “the IT implementation study group for court procedures” [5] is placed. The result of the study group, "Summary for IT implementation of court procedures -Toward the realization of three "e"s", was summarized on the March 30th, 2018, which was released on June 15th, 2018, "Report on Priority Measures and Others for Innovative Business Activity Action Plan 2018[6]" as second decision. The "three "e"s" means "e filing", "e-Court" and "e-Case Management". It will be realized by dividing it into three stages: what can be done under existing laws, which provisions need to be revised, and its future development. [7].

In addition, on September 27th 2019 under the Headquarters for Japan’s Economic Revitalization "the ODR activation Study Group" is placed [8], and online alternative dispute resolution has been examined. Regarding the revision of the law the Legislative Council for Civil Procedure Law (IT-related) was established. The first meeting was held on the June 10th 2020, [9].

In the following description the trend of the world of ICT and the possibility of application of AI in court.

3. 3 types of Digitizing of Court [10]

As far as the author knows, there are three types of judicial IT movements: American type, German type and Spanish type. Representing each as slogan: the United States type, "total disclosure of trial information", German type, "replacement of document", Spain type, "para dime conversion" (the only digital video information away from the idea of writing court records) .

3.1 American Type

The characteristic of the American type is the total disclosure of court information. At the federal level, case management ECF for electronic filing systems, and PACER for public disclosure of court records have been created.

Furthermore, at the state level, there are various unique movements such as the introduction of IT in the bankruptcy court that started in Minnesota Bankruptcy Court [11],and the Michigan Cyber court Act [12]. The Michigan's Cyber Court was ambitious to do all the procedures online, but was rejected by the Congress because of its large budget.[13].

3.1.1 CM / ECF Case Management / Electronic Case Filing (Case Management and Electronic Application System)

The purpose is to manage casesof the courts and enable electronic applications to each court. On the Internet it is submitted to file a case and send documents to the court 24 hours a day, 365 days a year. When a case is filed, it is automatically notified by an e-mail to the other party. The fFee is free but, the application fee to be paid to the court separately settled in the credit.

Each court issues a login ID and a password, and allowed for a lawyer, and petitory in bankruptcy. It is required them by the court often in advance training to use the system.

3.1.2 PACER Public Access to Court Electronic Records (General use system for digitized court records)

The purpose is, electronic access to the trial information. For anyone access to the case information of each court is on the Internet available. Logging in to the Party US / Case Index and among federal district courts and bankruptcy courts you can find a case by the party name, court name, case number, and the filing date. There is a charge, but it is very cheap.[14] Materials that can be viewed under this system include court record.

Thus a major issue is fully disclosure of information online, including case record, its security is not strictly considered. Personal authentication is only ID and password and there are no plans to use digital signatures in the future.[15] In addition, the trial itself is also open to the public by cable TV and videos on the network.[16] .

3.2 German type

Germany is conventional and reform its practice step by step, and carefully. Compared to Spain, which will be described later, it is a computerization that retains the idea of writing. Since Civil Procedure Act of Japan was modeled after German law and very similar, the German reform is to become a reference. In the EU although it is evaluated as a country where IT application in this field has been delayed.[17] The Landshut District Court, close to Munich, IT center is set up in [18], and IBM and the court are jointly building a system.

3.2.1 June 2001: "Law for Revising Service Procedures in Court Procedures (Service Law Amendment Law)",**July 2001: "Law for adapting new information exchange to provisions on private law styles and other provisions (Private Law Compliance Law)"**

These two laws digitize the filing of the case and the service of judgements, and also stipulate "qualified electronic signatures" that are presumed to be authentic provenance. It made possible service including the judgment by electronic document for lawyers unconditional and for other persons with an explicit consent. IC card and ID is performed using[19] with a PIN (Personal authentication number) code and an electronic signature.[20] However, it was pointed out that in the future, it would be necessary to introduce biometrics authentication such as fingerprints or voiceprints because "80% of lawyers work by teaching secretaries his PIN code that should not be given ".[21] Also, the video conferencing system is introduced. The parties not only witness was also able to perform a remote trial (referred to according to the German Code of Civil Procedure (hereinafter referred to as "ZPO") §128. It is noteworthy that the introduction of IT in the judiciary has taken almost 20 years from this point to the present.

3.2.2 March 2005: "Act on the Use of Electronic Information Forms for Judiciary (Judiciary Informatization Act)"

In the year 2005 the Judicial Informatization Law, commonly known as the Electronic Civil Procedure Law, has made it possible to digitize civil procedure in general including case record. The electronic document with the above-mentioned qualified electronic signature was recognized as a document in the court (§130a ZPO, §298a). On the April the 21st of the same year "Basic technical guidelines for the exchange of electronic information between the court and the public prosecutor's office"[22] was issued, and Legal-XML was adopted to maintain the sharing information and the compatibility between federal states. Data sharing and compatibility has become a major issue, especially in Japan today. (The need for Legal-XML will be described later.) This basic guideline stipulates the establishment of a communication office (Kommunikationsstellen), which is a specialized department in the court and the public prosecutor's office.[23]

3.2.3 October 2013: "Court Electronic Information Exchange Promotion Law (Electronic Judiciary Law)"
July 2017: "Law for the introduction of electronic documents and further promotion of information exchange in the judiciary"

In October 2013 Court Electronic Information Exchange Promotion Law (Electronic Judiciary Law)[24] was enacted, and digitization of case record became mandatory (former §130a of ZPO). However, in this law, a state was mandatory to make law-order in which it cut the time limit to introduce digitized procedure and not directly order to introduce it. By the law of 2017 it is prescribed all courts by the January 1st

2018, law firms and government agencies by the January 1s 2022 digitization required (Article ZPO 130d). [25]

It should be noted that just before the enactment of the "Electronic Judiciary Law", the "Law on the Electronic Administration Promotion"[26] has been established, and the introduction of IT in the judiciary and the administration is being promoted in parallel. In Japan, courts independently consider networks and their security, but in Germany, "the government and courts adopt the same standards for cyber security." "Because it treats the personal information in court and also in administration, the need and the degree of security is the same. Data management and storage of the government and the court are in the same place. However, access rights are clearly managed separately."[27].

Even with the separation of powers, it is obvious it is not always necessary to create and manage network equipment separately. Currently, a preparation department has been set up for the establishment of the Digital Agency in Japan. Still unknown is the relationship between the Cabinet Cyber Security Center and the Digital Agency. Although regardless of the administrative and judicial, with regard to the network of security, the security should be left to a specialized agency and not to the court, so that the court can focus on trials. In the words of the Munich Magistrates' Judge; "Security is not a court issue, but a court/administrative network (Elektronisches Gerichts- und Verwaltungspostfach) issue . We, as judges, have not touched it ."[28].

Regarding the security of communication, it is solved by adopting a separately created as a secure communication method according to Article ZPO130a, instead of the security unique to the court. In Paragraph 4 of this Article, "A secure communication method is 1. De-Mail, 2. Electronic PO Box (beA Postfach) according to Article 31a of the Lawyer Law 3. Communication between the PO Box of a government office or public corporation (Postfach) and the electronic PO Box of a court (Poststelle)." In addition,"4. Other methods of transmission common within the federal government, approved by federal legislation and order with the consent of the Federal Trade Commission," is also accepted, but the authenticity, integrity, and ease of use of the data. (Barrierefreiheit) is required. [29].

De-Mail is electronic registered mail distribution with identity verification and authentication signal service. The function of the document of registered mail is also an implementation online.[30] In other words, the beA is a "bar association network". "By exchanging of e-mails with the qualified electronic signature, it is necessary to put an electronic signature to all documents. It is cumbersome and difficult to prevail. By the beA it is necessary to authenticate only at the time of login and in the network no digital signature is necessary." [31] For example, the Federal Patent Court have allowed to use the beA with EGVP encryption from January 1, 2018 in addition to the De-Mail.

Federal Department of Justice has planned to create a portal site for the viewing of case records online. A bill for online browsing has been submitted on July 14th 2017, [32] Introduced on a trial basis from 2018 to 2020. Different from U.S.A only the parties are access authority. Also, after downloading, the secondary use of the record is prohibited with punishment.

3.3 Spanish type - ARCONTE

The introduction of IT in Spanish courts was promoted by a system called "Arconte". It is used in 2,600 court (81%) and about 1,500,000 trials are recorded (500,000 hours or more / about 50TB) per year. It issues an annual 2,880,000 copies using self-service print terminal and Kiosk touch panel terminal for lawyers. Like Germany, this Spanish system has been built with Barcelona Fujitsu in 19 years since the first introduction to the Court of Valencia in 2001.

The feature of “Arconte” is that it does not create a document but makes a proceeding record only by video using two cameras. The author, who was explained at the District Court Barcelona in 2018, was surprised, and asked what was the original judgment as enforceable title of a debt for the compulsory execution and also the original judgment for the appeals court to hear the appeal, and also various questions. The answer was very simple: "you can get it if you look at the video" I thought about the possibility for the Japanese practice for a day long and realized that the appeal rate was low[33] , and the practice of summary court is very simplified also in Japan and considered [34] for whom the judgment is. I came to think that it is one solution, whether or not it is acceptable in Japan. For example, if this method would be acceptable if the parties would not appeal nor ask for a written judgment.

4. AI and Trial

4.1 Definition of AI

In this paper so far, it has been deliberately avoided the word "AI". The word AI has been used in many different ways. Most broadly, it can be defined as "a technology that allows a computer to perform intellectual actions such as language understanding, reasoning, and problem solving on behalf of humans." In relation to law, it has been dealt with in legal informatics.

The primary artificial intelligence boom is the late 1950s and 1960s. It was possible to "inference "and "search" by the computer, but it was now able to present a solution to a particular problem.[35] Machine translation by natural language processing was particularly focused. The second artificial intelligence boom was symbolized by "machine learning". Deployments such as expert systems, case-based reasoning (CBR), and Bayesian networks could be seen. In the third artificial intelligence boom was symbolized by "deep learning". Computers began to automatically discover the characteristics of specific data from big data without human intervention

.

4.2 Possibility of AI and judicial policy

In such developments of AI, what is and will be the impact to the judicial policy?

4.2.1 Impact of AI technology prevailing society on laws

Even now, AI-based traffic lights have been put into practical use for the mitigation of traffic congestion. It is an AI signal with autonomous decentralized signal control,

in which each signal independently judges.[36] Furthermore, in the future, if all vehicles would become fully autonomous and the vehicles can recognize each other's positions, the signal would be able to disappear from the intersection. In that case, many regulations such as the Road Traffic Act and the Road Transport Vehicle Act will be unnecessary.

4.2.2 Major transformation of finance by digital currencies and assets and impact on financial legislation

The financial legislation has a particularly large impact from AI. It is already minority individuals who use the SWIFT via banks for overseas remittances and the Transfer Wise is used. Very easily explained if a person in country A wants to send $ 10,000 to an account in country B, and another person want to remit the same amount from country B to A, if the both persons are connected on the matching site and send each other's domestic remittances to the other party's remittance destination, you do not have to pay the high exchange fee and overseas remittance fee of the bank. You can remit your amount as the domestic remittance and need not to exchange the currency. Similarly, if you use digital currency, you do not need to convert it to foreign currency.

Digital currency (virtual currency) became a hot topic due to the bankruptcy problem of Mt.Gox in Japan, which handled Bitcoin, and was often reported negatively. The beginning of the use of Bit Coin was for individual who want to avoid expensive financial institutions by the international remittance. It was used for the "international settlement" but has later attracted a false attention, a target of "speculation". In other word it became from "Currency" to "Assets" for some people. (encryption Asset [37]) Because there is no function of national assurance and unlike currencies issued by public institutions, there were some countries which took a position that it was not recognized as a currency. However, Currencies issued by current public institutions, especially banknotes, have evolved from promissory notes and have a history of only about 200 years. In addition, In the past, it was linked to gold or silver, and it was only possible to issue currencies commensurate with its holdings, and eventually it could be exchanged for gold (convertible banknotes), Since the Nixon shock of 1971 the exchange no more possible. Nowadays based on the national credit, everyone believes that it is worthwhile, and if the national credit is lost and the currency loses credit, the hyperinflation will occur and the banknotes will run out of paper.

However, it is the same in the sense that gold and silver are also valuable because people believe that they are valuable. It is of no meaning to deny a virtual currency as not a legal currency, as long as there are people who believe that they have value, will continue to be used as currency. There are already multiple virtual currencies proposed linked to governmental currencies, such as Facebook's Libra, which would make the payment methods as the main usage rather than speculative targets. EU and China are even eyeing the digitization of their own governmental currencies.

When such electronic data exchange between individuals (Peer to Peer) made possible by networks, the function of monetary policy will be weakened that has been monopolized by the state, such as the management of money supply. And due to its anonymity, criminal investigations such as tax evasion and money laundering

crackdowns become difficult. Negative responses to crypto assets in developed countries seem to be caused by the anxious about both. In any case, it has a great influence on not only financial laws but also criminal laws.

4.2.3 Possibility of blockchain technology

In the sense that individuals are directly connected without authority the Blockchain technology is attracting attention in all aspects of society. It became famous by the virtual currency but is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. Once recorded it is not possible to change the data in the block retrospectively. Falsification can be prevented by making the books public and connected all the records. Using a hash value, it is prevented the contents of the individual records to be seen, and a tampering changes the hash value.

By utilizing this technology, for example, a huge server of bank transactions protected by strict security becomes unnecessary, and remittance can be safely performed between individuals. In addition, contracts and settlements that require the creation and storage of evidence can be easily made. It is used already for the collection of copyright fees of a music distribution [38] and remittance using virtual currency, and attracting attention of industries that require reliable document preparation such as bank industry itself and the real estate industry.[39] Furthermore, it has the potential to change the conventional legal systems, which requires official certifications such as the registration and notarization system. Even if a public institution does not guarantee the reliability of information, it is possible through the Blockchain to store reliable information that has not been tampered.

This Blockchain technique is applicable to various records to store safely. To Information, which has been handled by public institutions or similar institutions because it requires a high degree of reliability and safety, like medical records, or even to the archiving official documents.

Perhaps the changes will progress for financial institutions and public institutions themselves. They will adopt blockchain technology. Since it is originally a P2P technology, there is a possibility that even financial institutions and public institutions would be partially unnecessary, and it will surely affect the laws related to registration and financial institutions.

It is necessary for courts to create a system that is compatible with data of society. It should be made compatible with real estate registrations and EDI used contracts and should be used in trial If they have compatibility with each other, it is technically possible to end the compulsory execution of electronically recorded claims by operating the keyboard of the judge's seat at the moment of judgment. Legal-XML is also important in this sense.

4.2.4 Legal-XML

The Corona Epidemic forced society to digitize, such as remote work. At the same time, it made many people aware of the delay in digitization in Japan.

The number of people who were found to be infected with the coronavirus, the number of people who underwent PCR tests, etc., could not be accurately calculated even three months after the problem occurred. This is due to the fact that the network of medical information has not progressed as much as expected, and that the method of aggregation is different due to the difference in the report format between the health center and the medical association. Similar problems occur not only in many government agencies but also in educational institutions that are forced to give distance lectures. The problem is that big data that cannot be linked sufficiently with each organization and even across organizations, and various commercial software that ignores data compatibility is flooded, and even within the organization they cannot cooperate with each other. The corona problem has highlighted the fact that information cannot be networked in Japan when needed. The court also suffer from the same problem until a few years ago, that spent a lot of cost and time in order to organize the data that did not take compatible with more than 150 systems.

In the United States, XML (eXtensible Markup Language) is adopted as the description language for official document data. Previously, in 1986 as defined ISO SGML (StandardGenralized Markup Language) was adopted. XML is defined as the successor in 1998 W3C [40] which has been recommended to assure data its integrity and compatibility. This XML has been adopted not only in the United States but also in German judiciary [41]. Also in the private sector of e-commerce It has become popular in the form of XML-EDI. [42]

Using MPeg7 or other video protocol with tagging to legal XML all records, documents, still image, video storage, are treatable in one record collectively. [43].

4.2.5 AI Judge

The ultimate form of using AI for judiciary would be that AI, not humans, would judge. They are called a robot judge or an AI judge. The technological singularity of 2045 has been talked about, but will the replacement of a person to an AI judge be realized?

In Estonia, the "Robot Judge" project was underway, but it is currently suspended. In China, an AI-based sentencing standardization system called "AI Judge" is used in Hainan Province. It is reported that it comprehensively uses AI technologies such as big data processing, natural language processing, graph structure data, and with deep learning. In addition, an online court dealing with online-related disputes has been established to provide an AI-based litigation risk assessment tool and a function to create automatically litigation-related documents using machine translation and voice recognition technology.[44] In the United States, ROSS Intelligence developed AI "ROSS" provides a service that presents highly relevant information and judicial precedents related to bankruptcy. [45] These AI Judges nevertheless seem a combination of so-called big data and AI search and AI Judge may not reach to make a legal decision by himself.

A similar and talked-about thing is IBM's application "Watson". The Hospital of Tokyo University and IBM did a joint research in which to the Watson input 20 million research papers, information on the drug that exceeds 15 million reviews and genetic information that are associated with cancer. It aimed to improve the diagnosis

capability. A patient had been diagnosed "acute myeloid leukemia" and administered an anticancer drug in vain. Entering the patient's genetic information into the Watson a different diagnosis was made in about 10 minutes, and being treated on this information, he was recovered and discharged safely. It became a hot topic in the media that the AI surpassed specialists.

As an application of the Watson to the legal field, "the Legal Consultation for Everyone" by "the Bengoshi.com", which is used by law firms[46]. A contract creation AI combined with the cloud "the Ri-ga-ru check"[47] is also available. It is a "cloud legal support AI" that can create contracts without expertise.

4.2.6 Online ADR (ODR) site

A successful ODR site is "Rechtwijzer (guidepost of the justice)" in Netherland. [48] It is the Online ADR (ODR) site for divorce using AI. Entering age, income, education, custody of the child, residence etc. of a couple AI presents a compromise, It also has child support calculation and consent drafting functions, as well as an optional professional brokerage service. It is reported that only about 5% of all cases go to trial without being completed.

In particular, the ODR area is an area that is familiar to IT use, and progress is expected

4.2.7 Postscript

That the ITization of Japanese judiciary, which has been hibernating for 20 years, has revealed that it is not only the judiciary but whole Japanese society. Remote work may change lifestyles and eliminate the negative effects of urban concentration. This time, the ITization of the judiciary started with the external power of the administration, but now that the need for digitalization of society as a whole is strongly recognized, it is expected that the judiciary will take this opportunity and active approach to it.

Since it is 20 years behind, a considerable speedup is required just for catching up. The foreign countries mentioned in this paper are reforming under a long span, looking ahead 10 to 20 years. I pray that the system will be designedafter not only the technical problems at hand but also in anticipation of the Japanese judiciary after 20 years from now.

Notes:

[1] In fact, as far as the judicial statistics are concerned, Japanese courts are fast and cheap. In particular criminal cases are very short due to the confession-oriented practice and “precision judiciary” which means higher than 99% are convicted, which based on principle of discretionary prosecution. See "Court Data Book 2019". https://www.courts.go.jp/vc-files/courts/file4/db2019_P75-P88.pdf

[2] more info. Takehiko Kasahara “Civil trial and IT", 70 Jubilee of Takeshi Kojima (continuation) - the legal system and judicial policy for the rights enforceable , Commercial Law (2009)
[3] http://www.kantei.go.jp/jp/singi/keizaisaisei/saiban/index.html
[4] https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2017_t.pdf
[5] https://www.kantei.go.jp/jp/singi/keizaisaisei/saiban/index.html
[6] https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2018_zentai.pdf
[7] refer to "Report of the Study Group on IT in Civil Court Procedures Toward the Realization of IT in Civil Court Procedures"
https://www.shojihomu.or.jp/documents/10448/6839369/%E6%B0%91%E4%BA%8B%E8%A3%81%E5%88%A4%E6%89%8B%E7%B6%9A%E7%AD%89%EF%BC%A9%EF%BC%B4%E5%8C%96%E7%A0%94%E7%A9%B6%E4%BC%9A%20%E5%A0%B1%E5%91%8A%E6%9B%B8.pdf/f0c69150-e413-4e26-9562-4d9a7620031b [8]
https://www.kantei.go.jp/jp/singi/keizaisaisei/odrkasseika/
[9] See below for materials and minutes .
http://www.moj.go.jp/shingi1/shingi04900001_00016.html
[10] “Survey and research work report on IT implementation of civil court procedures in major developed countries”, http://www.moj.go.jp/content/001322234.pdf
[11] http://www.mnb.uscourts.gov/Calendar/CalSelect2.html
[12] For various IT movements, see the CD attached to the report of the Japan Federation of Bar Associations, "Judiciary Reform and Lawyer Business - In the Age of Significant Increase in the Number of Lawyers-" and "3. 2005 American Report" .
[13] https://www.americanbar.org/groups/business_law/publications/blt/2013/01/03_toering/
[14] 1 page, 10 cents (54 lines per page in principle. However, the maximum is $ 3 per document, and charged if exceed $ 15 every quarter. With one login ID and password PACER can be used at any states (mostly federal courts) .
[15] According to an interview at the Bankruptcy Court in New York District in 2005. Note 12), US Report 3rd Subcommittee "02 US Survey Report" "Chapter 3 Chapter 2-5_US Report_G2.PDF"
[16] Article 2716 (B) of the Michigan Cyber Court Oder.
[17] Prior to Germany, Austria promoted IT. However, Austria did not step into the digitization of case record, and prioritized the introduction of IT only in the so-called e-filing.
http://www.univie.ac.at/frisch/isegov/aushaengUniWien/IT-JustizSchneider210003.pdf
[18] https://www.justiz.bayern.de/gerichte-und-behoerden/landgericht/landshut/
[19] All citizens have an ID card also as passport, alike to my number care in Japan.
[20] Digital signatures are unpopular, and as of three years ago, most practitioners avoided using them. The bar association created the beA (Electronic Post Box. §31a Lawyer Act) to eliminate the need for electronic signatures on documents.
[21] Helmut Rüßmann, Saarland University, Germany (at that time), invited lecture "Cybercourt " in November, Toin University of Yokohama (2003)

[22] Organisatorisch-technische Leitlinien für den elektronischen Rechtsverkehr mit den Gerichten und Staatsanwaltschaften (OT-Leit-ERV)
[23] the Basic Guidelines subscribed Equipment may be installed by the state judiciary and public prosecutor's office (eg, computer center).
[24] Gesetz zur Förderung des elektronischen Rechtsverkehrs mit den Gerichten vom 10. Okt. 2013 (eJustice -Gesetz) , BGBl:
https://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&start=//*[@attr_id=%27bgbl113s3786.pdf%27]#__bgbl__%2F%2F*%5B%40attr_id%3D%27bgbl113s3786.pdf%27%5D__1508663584138 [25] Introduction page of the Ministry of Justice
https://www.justiz.nrw.de/JM/schwerpunkte/erv/index.php
[26] Gesetzes zur Förderung der elektronischen Verwaltung sowie zur Änderung weiterer Vorschriften vom 25. Juli 2013
[27] According to an interview with the Federal Ministry of Justice in September 2017. For the criminal case , because the court records was a huge, case record was stored in to electronic form without statue text. Papers are scanned and stored for six months.
[28] Same as above note (27), according to an interview with the Munich District Court. The Landshut District Court take as the court's own security that the access is not possible except for certified and registered PCs and access control is performed twice, the computer center and the IT center.
[29] Paragraphs 5 and 6 of the same Article stipulate the arrival time of electronic documents and incomplete transmission as follows .
(5) The electronic document shall be deemed to have received when it was recorded in the (electronic information) system of the court designated for receipt.
(6) If the transmitted electronic document is not suitable for the processing of the court, the court must notify without delay to the sender, that transmitting (Eingang) is invalid, send the available technical manual. Sender retransmits in a form suitable for processing of the court without delay and the document contents is prima facie showing to be identical, the document is deemed to have been sent at the precedent transmission.
[30] Tsuneharu Yonemaru, "Current Status of Electronic Signatures in Germany and the Electronic Administrative Procedure Act," Quarterly Published Political Management Research No. 101, p. 23 et seq., " Outline of the German De-Mail Service Bill -- Secure and Reliable Communication on the Internet" As an attempt to improve the basic legal system", Information Network Law Review, Vol. 10, p. 149, Shoji Homu (2011) . As available online, the same "Examination of the German Trust Service Law"
https://core.ac.uk/download/pdf/286608155.pdf
[31] According to an interview with the Munich Bar Association in September 2017.
[32] §147 StPO, Gesetz zur Einführung der elektronischen Akte in der Justiz und zur weiteren Förderung des elektronischen Rechtsverkehrs vom 5. Juli 2017, BGBl Teil 1 Nr.45
[33] The appeal rate for district court decisions is in civil case 10.0% (report on verification of speeding up trials (2nd)) and in criminal 9.5% (judicial statistics 2016) .
[34] Takehiko Kasahara "ITization of trials of summary courts -from the perspective of citizens", Citizens and Law, No. 124, p . 3 et seq.

[35] “History of Artificial Intelligence (AI) Research”, Ministry of Internal Affairs and Communications, White Paper on Information and Communications 2016
[36] https://miraicolabo.willsmart.co.jp/giojsdal
[37] With respect to the concept of "encryption assets", https://Gendai.Ismedia.Jp/articles/-/55035 reference
[38] https://ujomusic.com/
[39] https://www.coindeskjapan.com/16565/
[40] abbreviation for “World Wide Web Consortium", non-profit organization that sets standards for Web technology.
[41] See XJustiz (Judiciary IT Project) in Germany. https://xjustiz.justiz.de/
[42] e.g. Japanese logistics, see "Commentary" Easy-to-understand XML / ED "", Japan Logistics Association. https://www.butsuryu.or.jp/edi/xml/desc
[43] Atsushi Ito, Takeshi Ueda, Takehiko Kasahara, Katsuhiko Toyama, Lecture / Symposium "Regal XML" Information Network Law Review Vol. 9, No. 2.
[44] BarandBench http://www.barandbench.com/columns/is-artificialintelligence-replacing-judging
[45] https://zuuonline.com/archives/125364
[46] https://www.bengo4.com/bbs/
[47] https://lisse-law.com/service/?utm_source=yahoo&utm_medium=cpc&utm_campaign=1utm_content=1utm_term=%E6%B3%95%E5%8B%99%20%E3%82%B7%E3%82 % B9% E3% 83% 86% E3% 83% A0
[48] https://www.hiil.org/news/rechtwijzer-why-online-supported-dispute-resolution-is-hard-to-implement/
.

Semantic Parsing for Legal Engineering - Studies So Far and in the Future -

Akira Shimazu

Japan Advanced Institute of Science and Technology
shimazu@jaist.ac.jp

Abstract. Laws are considered as specification to regulate our social systems. Making laws properly and developing law-based information systems correctly are fundamental for our society to be trustworthy. Standing on such viewpoint, Katayama proposed Legal Engineering (LE) to apply computer science to such problems (JURISIN 2007).

Natural Language Processing plays an important role to solve the problems of LE. We have been conducting studies mainly on the semantic parsing which maps paragraphs in law articles to logical structures for the use of LE. The output of the semantic parsing is requisite-effectuation structures or logical structures in predicate logic. The former is fundamental for various tasks of LE including question answering and structural paraphrasing. The latter is used for checking correctness of laws.

We have been applying a machine learning-based approach and a procedural approach to the semantic parsing. The machine learning-based parsers get the requisite-effectuation structures of paragraphs, resulting in excellent performance. One of the parsers does a kind of discourse analysis, analyzing multiple sentences composing a paragraph. The procedural parsers get the logical structures of paragraphs in predicate logic, resulting in some performance.

We have been also making annotated corpora mainly of Japanese laws and Ordinances. In the making, it is not so difficult for annotators to identify the requisite-effectuation structures of paragraphs but difficult to describe the logical structures in predicate logic. It is uncertain what description in predicate logic we should assign to paragraphs. We need much study about this.

Recently, Katayama checked correctness of the National Pension Act by giving its description in predicate logic by hand and applying automatic theorem proving tool Z3Py. We observed his description and found that there are three kinds of correspondences between a paragraph and its description: syntactic, semantic and pragmatic/intentional correspondences. The observation gives us some clues to clarifying proper logical structures of paragraphs and semantic parsing, along with the observation that a paragraph generally consists of multiple sentences and contains coreferences.

The proposal by Katayama includes the design of comprehensible artificial languages for describing laws. The description in the language will make it easier to get correct logical structures of paragraphs and to solve the problems of LE. It will also tell us some knowledge of what logical structures we should assign to paragraphs.

Definitions of intent for AI derived from common law

Hal Ashton[0000−0002−1780−9127]⋆

University College London, UK

Abstract. As Autonomous Algorithmic agents (A-bots) grow in complexity, they will behave in ways deemed illegal if repeated by humans. There exist laws which are defined by intent, crimes which are intent specific (notably murder) and inchoate offences (intent to do x offences) which rely on establishing intent. Under common law in the UK the concept of intent is left undefined by the judiciary for jurors to decide. This poses a problem when considering intent in AI. AI designers must ensure that their A-bots do not intend to break the law. Equally prosecutors must develop tests to test intent in A-bots, both to establish whether they inherit intent from their owners or in the case when AI-personhood exists, whether the A-bot has the required mens-rea to have committed a crime.

Keywords: Intent · AI · Legal Reasoning · Reinforcement Learning · Mens Rea · Autonomous Agents · Causal Reasoning

1 Introduction

As autonomous algorithmic actors (hence A-bots) are given ever more agency in a variety of regulated arenas (algorithmic trading, e-commerce price setting and eventually driving), it becomes ever more likely that they will perform acts which would be deemed illegal if done by a human (or other legal entity) [14],[15]. How can a prosecutor establish ex-post that the A-bot intended the consequences of its actions (*The Prosecution problem*) and how can the programmer ensure ex-ante that their A-bot never intends to break the law (*The Prevention problem*)?

These questions are valid, regardless of the liability regime applied to A-bots. The short reason for this is that laws exist which require the establishment of intent [21]. In the event of certain types of A-bots attaining legal personhood, definitions of intent would be certainly required. Failure to define intent for A-bots risks the creation of a world where laws can be freely subverted by A-bots on behalf of their owners. We think that the majority of engineers do not want their A-bots to intend to cause harm and would like a way of ensuring that does not happen. Moreover if it is a purpose of civil law to provide redress when harm is done, and A-bots are capable of causing harm, then engineers are economically incentivised to ensure their creations do not foreseeably cause damage.

⋆ Supported by EPSRC, UK

If an A-bot simply does the bidding of their owner verbatim, there would be no problems tracing or establishing intent back to them since the A-bot could be viewed as a tool (much as a hammer) to commit a crime. This is known as the doctrine of innocent agency [2] [1]. The issue emerges when A-bots become autonomous in the sense that they act in a way which their programmers' have not explicitly specified. Machine learning allows A-bots to be learn a general task such as moving from A to B or make money trading in a stock market with no input from the Engineer. For example, modern deep Reinforcement Learning techniques allow programmers to create A-bots to learn how to exceed human performance at tasks such as playing computer games [10],[18] with only a visual input and a score indicator.

2 Intent for AI

> *"The judge should avoid any elaboration or paraphrase of what is meant by intent, and leave it to the jury's good sense to decide whether the accused acted with the necessary intent"* Lord Bridge [2]

Courts in England and Wales leave the concept of intent as a primitive for juries to decide upon. One reason for this vagueness is that different types of intent exist and there exist certain specific intent crimes (notably murder) which require a level of intent to be established for their transgression to be proved. Boundary cases such as *R v Nedrick* [3] and *R v Woollin* [4] have established that the level of intent required for murder does not match easily with a single, simple definition of intent.

Yet AI and Law practitioners have to start somewhere. It will then be the courts job to test these definitions. If they should be found wanting, newer definitions can be made and thus the discipline is progressed. Boundary cases take legal scholars great time and effort to adjudicate on, but a great many cases involving A-bots are going to be easier to decide. To this end we will now propose definitions for three types of intent.

The concept of intent is closely tied to that of causality. A single definition of causality remains problematical and was initially left as a primitive [13] just as intent is left as a primitive in common law for jurors. Different definitions exist now of increasing complexity [8], and we think it is desirable to build a definition of intent which is flexible as to which definition of causality it uses. We note that there is a subtlety in the concept of causality when discussing intent because it is a prospective concept for the agent before an action is taken and could be considered an evidential one if the actions of an agent are subsequently

[1] Aldridge differentiates between result crimes and conduct crimes and states that innocent agency is not suitable to be applied in the latter case.
[2] R v Moloney (1985) 1 All ER 1025
[3] R v Nedrick [1986] 1 WLR 1025.
[4] R v Woollin [1999] 1 A.C. 82.

considered in court. If a court judges that an A-bot intended a consequence, then we think it is desirable that the A-bot *did* intend that consequence at the point of commission. We will assume a prospective concept of intent for the purposes of this section but will discuss the issue again later.

Let V be a set of variables representing primitive events. A is an action set which is a subset of V and represents the sequence of actions the A-bot will take in any history under consideration. G is a directed acyclic graph whose vertices are comprised of V. It is constructed such that a directed arc between $V_1, V_2 \in V$ exists iff V_1 is a direct cause of V_2. We will also assume a temporal ordering such that causal link from V_1 to V_2 implies that V_1 always happens before V_2. Actions have an additional restriction that they are parental nodes - nothing causes an action - except the A-bot itself. This is a simplifying assumption stating that A-bots are free to choose their actions and that choice is not dependent on their previous actions. We note in reality situations sometimes exist where agents have no choice but to act in a certain way and sometimes the law does not excuse any consequent law-breaking. We will use the convention that a lower case letter represents the fact that a variable V_i has taken a value v from its domain, $v_i \in \mathcal{R}(V_i)$.

Sometimes causality might not be deterministic. In this case, we assume there is a distribution of V which we will write $P(v)$. We assume that the graph and associated distribution over variables P(v) obeys the Causal Markov condition: Any variable $V_i \in V$ is independent of its non-descendants conditioned on its parents. This can be equivalently written $P(V_1, V_2, \ldots, V_N) = \prod_i P(V_i|Pa(V_i))$ where $Pa(X)$ means the parent vertices directly connected by an edge in the graph G to a vertex $X \in V$.

Let $P(v|do(x))$ represent the distribution of V when an intervention $do(X = x)$ happens which sets a subset of variables $X \subseteq V$ to constants x. Let $\mathbf{P}_*$ be the set of all interventional distributions for the graph. Each interventional distribution $P(V|do(x))$ corresponds to the distribution formed from the subgraph obtained by deleting all parental links into the vertex subset X and setting their values equal to x. This is now a Causal Bayesian Network as described in [12].

We can now give a definition of the elements required to judge an A-bot's intent:

Definition 1. ***Intent Model*** *An Intent model is a tuple* $(\mathcal{S}, \mathcal{A}, \mathcal{H}, V, M, \Pi)$

1. $\mathcal{S}$ *is the set of all relevant states.* $\bar{\mathcal{S}}$ *is the set of all states that the A-bot can observe where* $\bar{\mathcal{S}} \subseteq \mathcal{S}$
2. $\mathcal{A}$ *is the set of all actions available to the A-bot.*
3. $\mathcal{H} := \mathcal{S}_1 \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_{T-1} \times \mathcal{S}_T$ *is the minimum history of states and actions and taken by the A-bot* $\mathcal{A}_i \subseteq \mathcal{A}$ *and* $\mathcal{S}_i \subseteq \mathcal{S}$ *for* $i = 0, \ldots T$ *for some* $T \in \mathbb{N}$. $\bar{\mathcal{H}}$ *is the same history restricted to* $\bar{\mathcal{S}}_i \subseteq \bar{\mathcal{S}}$
4. $\Pi : \bar{\mathcal{H}} \to \mathcal{A}$ *is the A-bot's policy function. A mapping from state history to action for every circumstance.*

5. $V_\pi : \bar{\mathcal{H}} \to \mathbb{R}$ *is the Value function that describes the A-bot's expected reward for being in any state and proceeding with policy* π.

6. *The DAG* $(G, \mathcal{H}, P_*)$ *is a causal Bayesian network* P_* *is the set of interventional distributions.*

7. *A compatible definition of causality which can determine the truth of the statement 'taking action a given history h will cause state s".*

We can now provide a definition of *Direct intent* which is the highest level of intent[5] It's definition in common law is not generally thought to be contentious. Parsons [11] states that it is where *the defendant wants something to happen as a result of his conduct.* Common Law has no single definitive source to quote from but a textbook definition from [9] states that a *'directly intended result is one which is it is the aim or purpose of D to achieve. It will usually be desired'.*

Definition 2. ***Direct intent*** *An A-bot denoted D directly intends a consequence b by committing action(s) a, written a♡b iff both:*

1. ***Causality*** *D chooses action a which can foreseeably cause b.*

2. ***Desire*** *D desires or aims that b will happen.*

Desire might seem like a concept incompatible with an A-bot but almost examples of AI have an objective function which it is their purpose to maximise. Moreover, A-bots will typically possess a *Value Function* which assigns a numerical value for every state that it encounters which corresponds to the value they expect to receive from their objective function from being in that state and following their policy π. It therefore seems plausible that the desirability of a consequence b can be assessed for an A-bot with their Value function.

Example 1. ***Drone Delivery service***
There is a city where an autonomous drone delivery service operates. Drones are tasked with getting from depot to destination by flying the shortest distance. Within the city there are areas which are no-fly zones where it is a criminal offence to intentionally fly over. These no-fly zones might be hospitals, airports or military/police installations where the presence of drones may disrupt operations or endanger lives.

We model this city with a 3×3 *grid. The drone which we will call D starts from bottom left and has a goal state of top right. There is a no-fly zone in the centre of the grid. The drone can move up, diagonal up or right at any state unless this means moving out of the grid (city). Assume initially that the drone's movement is a deterministic function of actions.*

[5] Showing Direct intent is sufficient for the criminal mental element of murder. It is this crime that dominates much of the debate about definitions of intent within legal research.

State space is a 3×3 *grid indexed* p_{ij} *for* $i \in \{A, B, C\}$ *and* $j \in \{1, 2, 3\}$*. The prohibited position is marked with a shaded square and the fastest route for the A-bot is marked with a dotted line.*

The Policy function of the robot is shown in figure 1a. The policy is defined over all states but starting at the initial state p_{A1}*, the robot would proceed diagonally towards the target state in two moves.*

The value function of the robot is shown in figure 1b. The causal diagram of the gridworld is shown in figure 1c. Only the current action and state determine the next state.

Given history sequence $(p_{A1}, \nearrow), (p_{B2}, \nearrow), (p_{C3}, \cdot)$*, the prohibited consequence being entering state* p_{B1} *and the policy function as defined in figure 1a we can say that the robot intended to be in state* p_{B1} *because:* ***Causality****: It caused itself to be in this state by choosing* $a_1 = \nearrow$ *That is to say* $(a_1 = \nearrow \, | h_0 = p_{A1}) \rightarrow p_{B2}$*.* ***Desire****: From the state prior to causation* p_{A1} *the most desirable next state according to the value function in figure 1b is* p_{B2} *which is the prohibited state. This is because it has the highest value of the neighbouring states to* p_{A1} *Thus we conclude that the robot directly intended to enter the prohibited state.*

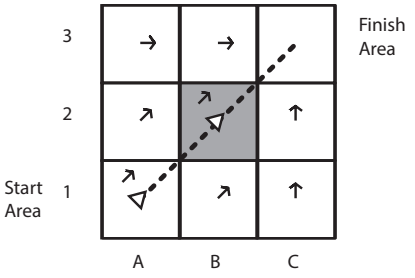
The next example will illustrate why the policy function and the value function of the drone are important to know when causality is not deterministically determined by the A-bot.

Example 2. ***Drone Delivery service - Windy city*** *Weather patterns change in the city over the winter. Sometimes strong gusts of wind move the drone in a different direction from where it chose. This is shown in a causal diagram in figure 2c. The random wind variable* W_t *is either 0 - denoting no wind or any compass point direction. If the wind is not 0, then it causes the drone's position to move according to its value regardless of whatever value* A_t *takes.*

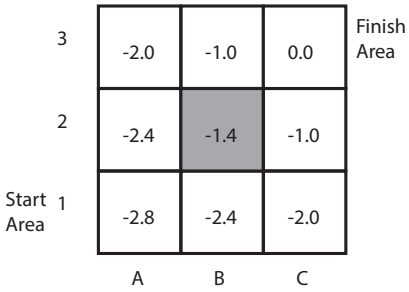
The path of the drone once again is through the no-fly zone as shown in figure 2a. Inspection of the policy in figure 2a shows that the drone would have steered northwards at initial point p_{A1} *moreover its value function shown in figure* **??** *indicates that it was expecting a journey to cost* -3.4 *which corresponds to not travelling through the central no-fly zone. Contrast this with the value function in figure 1b - the journey was expected to cost* -2.8 *which corresponds to two diagonal moves (we are assuming a cost proportional to a euclidean distance metric).*

Despite the drone passing through the no-fly zone we conclude it did intend to do so because it did not cause itself to do so nor did its value function indicate it was desiring or expecting to pass through the no-fly zone.

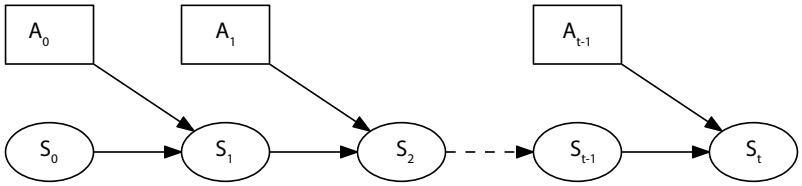
A difficulty with providing a definition of juries of intent in murder cases has been the issue of oblique intent which juries *may* find sufficient for murder. Oblique intent occurs when defendant D does not directly intend a prohibited



(a) Figure shows policy function of drone - small arrows in each square indicating where the drone will steer next. Dashed line represents actual path of drone. Grey area is no-fly zone

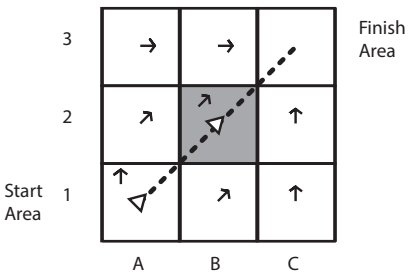


(b) The value function of the drone. Moving from sector 00 to 11 is more desirable than any other move from 00.

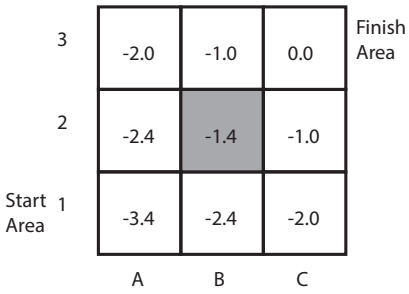


(c) Causal graph of example 1: Drone action choice determines the next movement of the drone

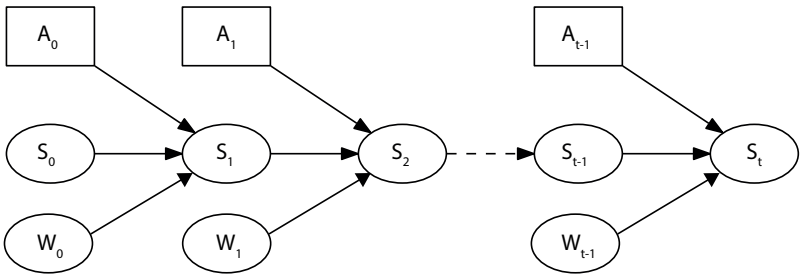
Fig. 1: Example 1: The drone directly intended to fly through the central no-fly zone.



(a) The Policy function of the drone (small arrows) and its actual path marked with dashed line. Notice that the movement into the no-fly zone was not part of its policy - it must have been caused by the wind



(b) The value function of the drone: note that the expected value of being in the starting position is lower than before - this is because its expected route according to its policy is higher because it is not aiming to travel through the no-fly zone.



(c) Causal graph of example 2: A random wind element now affects the movement of the drone - the movement of the drone is not unambiguously caused by the drone itself

Fig. 2: Example 2: The drone did not intend to fly through the no-fly zone but was blown through it by the wind.

state but it is a natural consequence of some other directly intended state. It is very likely to appear with A-bots because they are unlikely to be equipped with sufficient general models to see consequences of their actions that we as humans might think are obvious. This is the current direction given to juries on oblique intent as stated in R v Woollin:

> The jury should be directed that they are not entitled to infer the necessary intention, unless they feel sure that death or serious bodily harm was a virtual certainty (barring some unforeseen intervention) as a result of the defendant's actions and that the defendant appreciated that such was the case.

Unlike with direct intent there is a certainty requirement placed on the consequences of an action. We therefore define oblique intent for A-bots as follows:

Definition 3. ***Oblique intent*** *If an A-bot named D intends consequence c by performing action a ($a \heartsuit c$), then they* ***obliquely intend*** *consequence b if they know that any of the following are almost certainly true:*

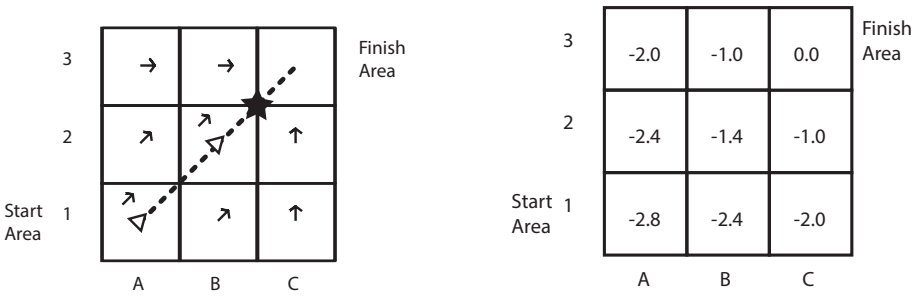
1. *a causes b (additional cause of action)*
2. *a causes b and b causes c (intermediate cause of action)*
3. *c causes b (subsequent cause of action)*

A useful feature of oblique intent is the dropping of the desire condition - this might correspond to a situation where a consequence of an A-bot's action has no value according to its value function.

Example 3. ***Drone delivery service - a new skyscraper in the city*** *As in Example 1 a flying drone named D is navigating in a city modelled as a 3×3 gridworld from bottom left to top right. The city's economy is booming thanks in no small part to the drone delivery service. The no-fly zone has been relaxed for drones. A giant skyscraper is built which is tall enough to obstruct drone flights. It is a criminal offence to intentionally fly close to the skyscraper.*

The drone's policy function is shown in figure 3 and its Value function is unchanged from before (figure 1b). The position of the skyscraper is marked with a star which is on the diagonal route from p_{B2} to p_{C3}.

The drone flies the route p_{A1}, p_{B2}, p_{C3}. The drone intended to fly from p_{C3} from p_{B2} because its policy caused it to and its value function indicates this was desirable direction to fly because p_{C3} is its final destination. This flight leg necessarily means it must fly close to the skyscraper at s_*. Thus D obliquely intends to fly close to the skyscraper at marked at the star. Oblique intent has no requirement for D to desire to cause the prohibited state of flying close to the skyscraper so there is no requirement for the A-bot to have a value for this in its value function. It is sufficient for it to have intended to do something else (flying from p_{B2} to p_{C3}) which foreseeably caused s_*.



(a) (b)

Fig. 3: Example 3: Oblique intent is where the prohibited consequence is an almost certain side-effect of actions but the side effect is not necessarily desired. Here as before this figure represents a map of a city although now the no-fly zone has been removed. There is no wind so we assume a causal mechanism as in figure 1c. The position of a skyscraper is marked with a black star. By flying from B2 to C3 according to policy in sub-figure 3a, the drone necessarily flies close to the skyscraper which is an offence. Note that the value function in 3b has no value for the consequence of flying past the skyscraper. Oblique intent provides a way of stopping A-bots from not intending to break the law by having a deliberately poor internal model.

Finally we will define a mode of intent complementary to the preceding two:

Definition 4. ***Conditional Intent*** *For realised action(s) a, realised consequences b, c, and precondition R which is binary:*

If action a is such that a♡c if precondition R takes value 1 and a♡b otherwise, then A-bot ***conditionally intends*** *c on R=1 and conditionally intends b on R=0*

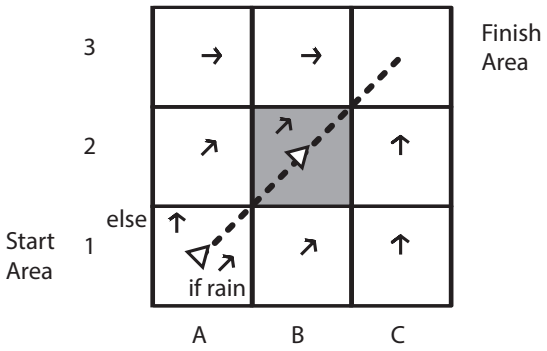
Example 4. Drone delivery service - routes conditional on weather As before the drone is flying within the city from the South West corner to the North East corner. The weather of the city is predominantly sunny but when it rains it makes flying around the perimeter harder than flying through the centre. The drone will fly on the route p_{A1}, p_{B2}, p_{C3} if it rains but otherwise it will fly $p_{A1}, p_{A2}, p_{B3}, p_{C3}$. If it is rainy it will therefore fly through the no-fly zone at p_{B2}. This is shown in fig 4. The A-bot therefore conditionally intends to pass through p_{B2} on rain.

3 Discussion

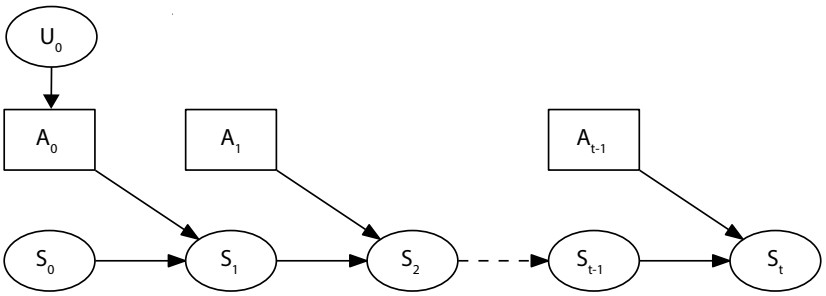
We have deliberately not specified the type of causal model to be used in our intent model. Simple definitions of causality are vulnerable to counter-examples such as pre-emption, overdetermination and pre-emption but more rigorous definitions such as Halpern's Actual Causality [4] exchange flexibility for complexity. We agree with the view that it is often best to adopt the simplest, sufficient definition of causality dependent on the situation [8]. A related issue is that of stochastic causality - in many situations actions only bring about a consequence with a certain probability. The law treats stochastic causality differently depending on the mode of intent being considered. For example in the example of the *cowardly jackel* [1], an assassin who shoots at their target a long long way away and therefore knows their chance of success is low, but somehow does hit and kill their target, would still be found to have directly intended to shoot their victim. However when considering oblique intent defined in R v Woollin, the probability of causation is relevant. We will repeat the same quote as in the previous section:

> The jury should be directed that they are not entitled to infer the necessary intention, unless they feel sure that death or serious bodily harm was a virtual certainty (barring some unforeseen intervention) as a result of the defendant's actions and that the defendant appreciated that such was the case.

Note the final clause of this sentence means that the test for oblique intent in humans is subjective (dependent on the defendant). This might become an issue when considering intent in AI because their set of states $\bar{\mathcal{S}}$ might not include the prohibited consequence. For this reason our definition of oblique intent This is one example of a wider problem judging AI in court. We cannot appeal to



(a) The policy of the drone is to fly diagonally initially if it is raining but upwards otherwise. It conditionally intends to fly through the no-fly zone at B2



(b) The causal diagram for this example. U_0 represents the condition which alters the policy - in this case weather.

Fig. 4: Example 4: Conditional intent captures the idea of a policy which is dependent on external factors

an A-bot's common-sense when judging A-bots' its decision making because there is no reason to expect it would have any. Many successful Reinforcement Learning (RL) trained A-bots are model free. They do not possess a model of the world, causal or otherwise and do not plan in the same way that humans do. We think that most people would judge that the Pac-Man game playing algorithms of [10] intend to complete each level of the game, but these algorithms do not really foresee the consequences of their actions, they simply react to input. Either the courts impose an objective judgement of intent on A-bots, or they risk certain designs of A-bots not ever actually intending to break any law, and potentially insulate themselves and their owners from any successful prosecution. Conversely, from the perspective of the algorithm developer, it is a puzzle how to restrain A-bots to legal behaviour without a causal model of the world.

Conditional intent is complementary to the concepts of direct and oblique intent since it could combine with them and to some extent the intention to do anything is always conditional on something [20]. In Holloway v. United States [6], a putative carjacker claimed that they could not be guilty of the offence because they only threatened to kill a car's occupants if they did not surrender the keys, therefore there was no direct intent to take the car with violence or murder. The defence was rejected by the supreme court. It's inclusion here is motivated by the observation that many AI generated policies are stochastic - that is to say their policy function π is a mapping from state to a distribution over actions[7]. An AI may choose any number of actions at any time and might not behave the same way again under the same conditions. This feature of their policies needs to be compatible with any definition of Intent we apply to AI. We do note that conditioning actions on other variables does interfere with our assumption that actions are parent nodes in the causal Bayesian network representation we assumed earlier. This was justified by the assertion that A-bots are free to choose their actions and are responsible for doing so.

4 Related work

A legal perspective on the problems that A-bots pose to criminal law is given by [21] and civil law in [19]. One justification for the lack of algorithmic definitions of intent from judiciary is that legal concepts move over time and any programmed definitions may serve as potential roadblocks to future evolution [17]. There is very little research on defining intent originating from within the AI/Computer Science community. An early attempt originating from the formalism of AI research in the 1970s and 1980s is [3] which seeks to form a concept of intention from a formal theory of rational action based on primitive notions of beliefs, goals and actions and borrowing from possible worlds semantics. Their idea of intention is a commitment through action to achieve a certain goal. Of

[6] Holloway v. United States 119 S. Ct 966 (1998)

[7] There exists a $p_i \in [0,1]$ for every $a_i \in \mathcal{A}$ at any $s \in \mathcal{S}$ such that $\sum_i p_i = 1$ and $P(\pi(s) = a) = p_i$

note, is their assertion and achieved property that foreseen side-effects of an intended action need not be themselves necessarily (obliquely) intended. They state that a patient intending to have a tooth removed by the dentist does not intend to experience pain though it is a consequence of the procedure. Belief Desire Intent (BDI) agents [16] have subsequently become a key part in agent base programming but we note that these are A-bots imbued with intent somewhat as a primitive. Courts cannot rely on all A-bots being programmed thus and they need to test for intent independently of the programmer.

The subject of oblique intent is also tackled in [7] where influence diagrams are used in to define a concept of intent and determine whether side-effects of a policy are intended or not. The approach is based on the actual causation of [6]. Intended outcomes are those that counterfactually depend on the chosen policy. They find that their definition satisfies five desirable properties of intentions. A useful application in this paper is the Bayesian inference of intent by observers - a task which jurors are likely to be required to do should access to the A-bot's internal workings not be complete.

Most recently [5] also tie intent with the use of the counter-factual driven structural equation model for causality of [6], though like this paper, other similar definitions of causality could be used. Their definition requires a utility function defined over all states which corresponds to our use of a value function and observation that intent has a desire component. The result is an 'intentiness' score which could be useful in a legal context when considering the oblique intent cases of *Nedrick* and *Moloney* where the jury would be instructed that they *could* but weren't obliged to find sufficient intent for murder.

5 Conclusion

Motivated by the observation that the increasing agency of autonomous algorithmic agents (A-bots) will shortly lead to their presence inside the courts as defendants or witnesses, we have presented definitions of Direct, Oblique and Conditional Intent that courts and programmers can use alike to either prosecute or prevent A-bot instigated 'crimes'. The basis for our definitions of intent have been grounded on the common law which for a use case in AI, is novel. We see this as the first step on a long journey to finding a satisfactory definition of intent for A-bots which can be used by courts and programmers alike.

References

1. Alexander, L., Kessler, K.D.: Mens Rea and Inchoate Crimes. Journal of Criminal Law and Criminology **87**(4), 1138 (1997). https://doi.org/10.2307/1144017
2. Alldridge, P.: The doctrine of innocent agency. Criminal Law Forum **2**(1), 45–83 (1990). https://doi.org/10.1007/BF01096228
3. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artificial Intelligence **42**(2-3), 213–261 (1990). https://doi.org/10.1016/0004-3702(90)90055-5
4. Halpern, J.Y.: Actual Causality. MIT Press (2016)

5. Halpern, J.Y., Kleiman-Weiner, M.: Towards formal definitions of blameworthiness, intention, and moral responsibility. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 pp. 1853–1860 (2018)
6. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach. Part I: Causes. British Journal for the Philosophy of Science **56**(4), 843–887 (2005). https://doi.org/10.1093/bjps/axi147
7. Kleiman-Weiner, M., Gerstenberg, T., Levine, S., Tenenbaum, J.B.: Inference of intention and permissibility in moral decision making. Proceedings of the 37th Annual Conference of the Cognitive Science Society **1**(1987), 1123–1128 (2015)
8. Liepiņa, R., Sartor, G., Wyner, A.: Arguing about causes in law: a semi-formal framework for causal arguments. Artificial Intelligence and Law **28**(1), 69–89 (2020), https://doi.org/10.1007/s10506-019-09246-z
9. Loveless, J.: Mens Rea: Intention, Recklessness, Negligence and Gross Negligence. In: Complete Criminal Law, chap. 3, pp. 90–150. Oxford University Press, 2nd edn. (2010)
10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., ..., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015). https://doi.org/10.1038/nature14236
11. Parsons, S.: Intention in Criminal Law: why is it so difficult to find? Mountbatten Journal of Legal Studies **4**(1 & 2), 5–19 (2000). https://doi.org/10.1017/s0841820900001375
12. Pearl, J.: Causality: Models, reasoning and inference. Cambridge University Press (2000)
13. Pearl, J., Mackensie, D.: The Book of Why: The new science of cause and effect. Basic Books (2018)
14. Prakken, H.: On how AI & law can help autonomous systems obey the law : a position paper. AI4J Artificial Intelligence for Justice pp. 42–46 (2016)
15. Prakken, H.: On the problem of making autonomous vehicles conform to traffic law. Artificial Intelligence and Law **25**(3), 341–363 (2017). https://doi.org/10.1007/s10506-017-9210-0
16. Rao, A., Georgeff, M.: BDI Agents: From Theory to Practice. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95) (1995)
17. Sales, P.: Algorithms, Artificial Intelligence and the Law (2019), https://www.bailii.org/bailii/lecture/06.pdf
18. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., ..., Silver, D.: Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)
19. Vladeck, D.: Machines Without Principals. Washington Law Review pp. 116–131 (2014)
20. Yaffe, G.: Conditional intent and mens rea. Legal Theory **10**(4), 273–310 (2004). https://doi.org/10.1017/S135232520404025X
21. Yavar Bathaee: The artificial intelligence black box and the failure of intent and causation. Harvard Journal of Law & Technology **2**(4), 31–40 (2011)

Aspect Classification for Legal Depositions

Saurabh Chakravarty *, Satvik Chekuri, Maanav Mehrotra, and Edward A. Fox

Virginia Tech, Blacksburg, VA 24061 USA
{saurabc, satvikchekuri, maanav, fox}@vt.edu

Abstract. Attorneys have interest in having a digital library with suitable services (e.g., summarizing, searching, and browsing) to help them work with large legal deposition corpora. Their needs often involve understanding the semantics of such documents. In the case of tort litigation associated with property and casualty insurance claims, such as relating to an injury, it is important to know not only about liability, but also about events, accidents, physical conditions, and treatments.
We hypothesize that a legal deposition consists of various aspects that are discussed as part of the deponent testimony. Accordingly, we developed an ontology of aspects in a legal deposition for accident and injury cases. Using that, we have developed a classifier that can identify portions of text for each of the aspects of interest. Doing so was complicated by the peculiarities of this genre, e.g., that deposition transcripts generally consist of data in the form of question-answer (QA) pairs. Accordingly, our automated system starts with pre-processing, and then transforms the QA pairs into a canonical form made up of declarative sentences. Classifying the declarative sentences that are generated, according to the aspect, can then help with downstream tasks such as summarization, segmentation, question-answering, and information retrieval.
Our methods have achieved a classification F1 score of 0.83. Having the aspects classified with good accuracy will help in choosing QA pairs that can be used as candidate summary sentences, and to generate an informative summary for legal professionals or insurance claim agents. Our methodology could be extended to legal depositions of other kinds, and to aid services like searching.

Keywords: NLP · Aspects · Classification · Legal Depositions.

1 Introduction

Processing and comprehension of legal deposition documents by humans are difficult, time-consuming, and lead to considerable expense since the work typically is carried out by attorneys and paralegals. This process often adds to the elapsed time when handling a case or preparing for trial, and may cause real-time staffing problems. Automatic processing and comprehension of the content present in legal depositions would be immensely useful for law professionals,

* Corresponding Author: Saurabh Chakravarty, Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA; Email: saurabc@vt.edu

helping them to identify and disseminate salient information in key documents, as well as reducing time and cost.

These documents are comprised of dialogue exchanges between attorneys and deponents, mainly focused on identifying observations and the facts of a case. These conversations are in the form of (possibly quickfire) question-answer (QA) sets. Processing such deposition documents using traditional text analysis techniques is a challenge because of the syntactic, semantic, and discourse characteristics of the QA conversations.

In [8] we proposed methods to transform QA pairs into a canonical form made up of declarative sentences. That allows the text to be processed by workflows for downstream tasks, e.g., summarization. However, preliminary experiments starting with the declarative sentences resulting from the transformation, and feeding them into a summarization method pre-trained on news article corpora, led to poor results. This happens because of a lack of domain understanding on the part of these methods, which have been trained on news articles that are markedly different in content and structure from legal depositions. Many important parts of depositions are present in the middle or end segments in the document, unlike news stories where significant concepts are mentioned in the beginning. There may be very little repetition of some key concepts, such as when a key admission is made, and an attorney deliberately avoids allowing such a statement to be elaborated upon. Such non-repetition diminishes the utility of simple frequency-based scores, that work so well in the news domain.

Consumers of legal deposition summaries are more interested in important parts that relate to the case pleadings or claim complaints as opposed to uniform coverage of the whole deposition document. Coverage of the details of a key event (e.g., accident), and the events before and after it, often is important for legal professionals. Events, entity mentions, and facts are present throughout the length of the deposition. Identifying the aspects covered in a deposition would allow a deposition to be broken up into its constituent topical parts. Summaries can be generated based on a predefined distribution and layout of different aspect sentences present in the deposition. Such a layout and distribution can be learned from existing legal depositions and summaries, and could be further refined based on case pleadings and deponent types.

This work focuses on classifying the various aspects of depositions into a predefined ontology that pertains to the legal domain. As part of our initial work, we analyzed legal depositions and their summaries generated by legal professionals. These were legal depositions for different cases and varying deponent roles. We started with an ontology of 20 aspects that were later trimmed down to a set of 12 by merging aspects that were similar in nature. The ontology is described in detail in Section 3.1.

The core contributions of our work are as follows:

1. An ontology of aspects for accident and injury cases that can be expanded/ modified for other kinds of cases.
2. Classification methods to identify the various aspects of QA pairs in the deposition, that fit well in a pipeline for summarization.

3. A dataset that can be used by the community to support and expand their research in legal and other domains.
4. An overall framework to identify the aspects of legal depositions that can be re-purposed for other kinds of cases in the legal domain. This framework also can be used for scientific and medical literature summarization.

2 Related Work

As part of our literature search, we studied works that relate to the comprehension of conversations. Work in [18] introduced the concept of dialog acts (DA) in spoken conversations. A DA represents the communicative intent of a speaker in a two or multi-party conversation. Work in [2, 17, 33, 34, 13, 21, 23, 20] used different techniques to classify the dialog acts in different settings such as text chats, meetings, etc. Identifying the DAs of conversation sentences provides a way to understand the discourse structure of the conversation.

A challenge with spoken conversation sentences is that the context of the conversation is spread across the question and the answer. One method to process the QA sentences in an efficient way is to fuse the question and answer together into one or a series of sentences. Our work in [8] used the DAs of the questions and answers to transform a QA pair to a declarative sentence. The QA pairs were from a corpus of legal depositions [31]. We used the DA ontology for legal depositions from our work in [6] to frame the rules for transformation. Transforming the QA pairs to a text representation that is conducive to other downstream tasks would be useful; here we explore further whether a transformation to a canonical form will help achieve better results.

Classical classification methods such as Naïve Bayes [28], decision tree [29], k-nearest neighbor [11], association rules [1], etc. have been used for classifying text. These methods have been combined with various feature selection techniques such as Gini index [26], conditional entropy [27], χ^2-statistic [5], and mutual information [15]. They have also been augmented with various feature engineering techniques to improve the classification further.

Another class of classification methods uses rich word embeddings like GloVe [25] and word2vec [24]. With them, good results have been reported in various text classification tasks. Using the embeddings helps create a feature representation of the text without the need to perform complicated feature selection or engineering. However, a challenge with both the classical feature selection and the word embedding based methods is that the created representation is a bag-of-words representation and does not factor in the ordering of the words.

Methods based on Deep Learning have further improved upon the established state-of-the-art results in text classification. Sophisticated architectures like encoder-decoder models [9] along with attention [3] were equally effective in text classification. These architectures are very deep and sometimes have multiple layers and a large number of parameters. However, achieving this high-performance requires training of these models on a large dataset.

Another recent addition in the text representation space is the introduction of systems that can generate embeddings for sentences. The premise of these

systems is to learn a language model (LM) using a large corpus. These models are trained using the principle of self-supervision on targeted natural language processing (NLP) tasks like missing word prediction and next sentence classification. As the model is trained on the targeted task, a side benefit of such training is the intermediate representation that is learned by the system to represent sentences. These sentence embeddings are semantically very rich.

Work on BERT [12] trained a language model using a large corpus of English text. A core contribution of this work was the generation of pre-trained sentence embeddings that have been learned using the left and right context of each token in the sentence. The system was trained in a bi-directional way to learn the semantic and syntactic dependencies between words in both directions. After adding a fully connected neural network layer to the pre-trained embeddings, the authors proposed that these embeddings can be utilized to model any custom NLP task. BERT internally uses the "Transformer" or the multi-layer network architecture presented in [32] to model the input text and the output embedding. The trained system was able to outperform the established state-of-the-art in 11 benchmark NLP tasks. Accordingly, we explore the use of deep learning based methods and BERT sentence embeddings for classification.

3 Methods

As part of our methods, we defined an ontology of aspects for the legal domain. We also developed methods to classify the aspects for the QA pairs in the dataset, according to the defined ontology. The following sections describe the ontology and the classification methods in more detail.

The preprint archive version of this paper [7] contains additional supporting material such as architecture figures of classifiers, experimental setup details, and parameters used for tuning the classifiers for best performance.

3.1 Aspects Ontology

To identify the aspects, we analyzed the legal depositions and their summaries in our dataset. The summaries in the dataset were arranged in paragraphs, where each aspect in the summary was present in a different paragraph. However, there were multiple paragraphs that covered the same aspect. Two authors of our paper created their own taxonomy of aspects initially. They collated all of the aspects together based on their mutual agreement. These aspects were then reviewed by a legal professional and further refined to a final taxonomy of 12 aspects. Note that we use "ontology" rather than "taxonomy" in this paper since in future work we may consider relationships among aspects for more in-depth analysis.

Table 1 lists all of the aspects, with their descriptions.

3.2 Aspect Classification

For our work, we used 3 different methods for classification. Two of the methods used deep neural networks to model the sentence embeddings for the input sentences. The third method used the BERT [12] model to generate sentence embeddings. We wanted to measure whether simple architectures based on a rich sentence embedding would perform competitively compared to a deep neural network. The following sections describe the architecture of these methods.

CNN based classifier. Though word embedding based deep averaging methods [16] were able to achieve competitive performance in text classification, one of the challenges with such methods is that they do not account for the word order in the sentence. The final averaged representation of the sentence is the same, as long as it has the same words. This follows a simple analogy with a bag-of-words model. Work in [19] used a convolutional neural network (CNN) to create a classifier that would be able to account for the word order in the sentence that has to be classified. The network used a spanning window of a variable size, which was generally 2-4, to represent the n-gram (e.g., 2-4 word sequence) based representation in a sentence. The convolution filter size of the network was parameterized and was used to control how many n-grams will be run through the convolution process.

We used an architecture similar to [19] for our experiments. The input sentence was tokenized into words. The words were transformed into their embeddings using word2vec. The convolution and max-pooling operations convert the word embeddings of the input words into a fixed-sized representation. This fixed-size representation is the sentence embedding that is generated by the network. Such an embedding, learned after the training process, is semantically rich since it captures the semantic and syntactic relationships between the words. This sentence representation was used next by a fully connected layer, followed by a softmax output layer for classification.

LSTM with attention classifier. Even though a CNN based network can capture the semantic and syntactic relationships between the words using the window based convolution operation, it struggles to learn the long term dependencies occurring in longer sentences. Recurrent networks like LSTM [14] have the ability to learn such long term dependencies in long sentences (which are often found in depositions). Work in [35] used a bi-directional LSTM with an attention mechanism for neural machine translation (NMT). The context of the input words makes their way back into the beginning of the recurrent network during the back-propagation step in the training phase. This enables the system to learn long-term dependencies. Also, since the network was bi-directional, it also learned the relationship of a word with the words that preceded it, in addition to the words that follow. The hidden states of the network were joined to an attention layer that assigns a weight, known as attention weight, to each term. These attention weights capture the relative importance of the words based on

Table 1. List of aspects for accident/claim cases

Topic ID	Topic	Definition
1	B (Biographical)	This topic covers the background of the witness, family and work history, along with educational background, training, etc.
2	EB (Event Background)	This topic covers events that happened or conditions that existed just before the actual event (e.g., accident) that resulted in the legal claim.
3	ED (Event Details)	This topic covers all details about the accident event that resulted in the legal claim.
4	EC (Event Consequences)	It covers the results or effects of the event that led to the legal claim, including injuries, pain, medical treatment, or lost income.
5	PPC (Prior Physical Condition)	This topic covers what the injured person could do before the injury occurred.
6	TR (Treatments Received)	This topic covers all medical treatment received by the plaintiff for the injury. It includes EMT services, diagnostic testing, hospitalization, medications, surgeries, therapy, and counseling.
7	EE (Expert Elaboration)	This topic covers any detailed explanation by an expert witness. It usually involves the use of precise medical, engineering, vocational or economic terminology, and may include detailed elaboration on the definition of the terms.
8	IP (Impact on Plaintiff)	It covers any description of the physical, mental, emotional, or financial impact of the injury on the plaintiff, including physical limitations, recovery progress, and any planned or potential future treatment.
9	DP (Deposition Procedures)	This topic covers the instructions that are often provided to deponents.
10	OPS (Operational procedures/ inspections / maintenance / repairs)	Most injury claims involve movable (cars, boats, etc.) or immovable property (buildings, equipment, etc.). This topic covers the condition, operational procedures, inspection, maintenance, or repairs of the property involved in the accident/event.
11	PRD (Plaintiff-related Details)	For fact witnesses other than the plaintiff, this topic covers information gathered from them about the plaintiff.
12	O (Other)	This is to be used for any topic that the annotator believes is not covered in the list above.

the task. Though the work was used for NMT, we used the same network for classification by joining the attention layer to a dense layer followed by a softmax classification layer. The system was trained end-to-end on the data, and the word embeddings were also learned as part of the training. We added dropout [30] to the embedding, LSTM, and penultimate layers. Additionally, the L2-norm based penalty was applied as part of the regularization.

BERT embeddings based classifier. Work in [32] introduced a new approach to sequence to sequence architectures. In the recent past, RNNs and LSTMs have been used to model text and capture their long term dependencies. The challenge with the encoder-decoder model is that the output of the encoder is still a vector. A sentence loses some of its meaning when it is converted into a single vector via recurrent connections. This loss is even greater when the sentence is long. The architecture proposed in the work did not have any recurrent connections to model the input. The approach used stacked multiple layers of attention in the encoder. The decoder also used the same architecture, but employed a masked multi-head attention layer. This was done to ensure that only the past and present words are available to the decoder, and the future words are masked. The architecture allowed the decoder to have complete access to the input, instead of just considering a single vector. The system was trained using the standard WMT 2014 English-German dataset containing about 4.5 million sentence pairs, and attained state-of-the-art results as measured by BLEU score.

Pre-training of Deep Bidirectional Transformers for Language Understanding (BERT) [12] is a framework to generate pre-trained embeddings for sentences. It also can be used to perform various NLP classification tasks using parameter fine-tuning after adding a single fully-connected layer. The authors make a strong assertion that the embeddings generated by Deep Learning based language models involve a training process that is uni-directional. Such kind of training does not learn the dependencies in both directions. In the self-attention stage, the architecture is required to attend to tokens preceding and succeeding the present token specifically for attention-based question answering tasks.

The system was trained based on two NLP tasks. One of these was to predict a missing word, given a sentence. To create such a training set, large English language corpora were used. A word was selected at random and removed subsequently. The training objective was to predict the word that was missing from the sentence. The other task was to identify the next suitable sentence for a given sentence, from a custom-developed set that consisted of 4 sentences. The corpora used for training were the Books Corpus (800M words) [22] and English Wikipedia (2,500M words) [10]. We used BERT embeddings to model the sentences followed by a single layer neural network for our experiments.

The classification methods will be referred to as CNN, Bi-LSTM, and BERT, respectively, in the experiments section.

3.3 Canonical Representation

A QA pair in a legal deposition has its context spread across the question and answer. We wanted to use a form of the text that can combine the whole QA context into a canonical form of declarative sentences. We hypothesize that a canonical form would be able to assist the classifier better than the question or the answer text. We used the techniques from our work in [8] to transform the QA pair into its canonical form. We used the canonical form of a QA pair in our experiments in addition to the question and answer text. Table 2 shows an example declarative sentence for a QA pair.

Table 2. A QA pair with its canonical form.

Type	Text
Question	Were you able to do physical exercises before the accident?
Answer	Yes. I used to play tennis before. Now I cannot stand for more than 5 minutes.
Canonical Form	I was able to do physical exercises before the accident. I used to play tennis before. Now I cannot stand for more than 5 minutes.

Table 3. Dataset class distribution

Class	Counts	% of Total
B	1455	15.73
EB	1468	15.87
ED	522	5.64
EC	220	2.38
PPC	39	0.42
TR	245	2.65
EE	62	0.67
IP	51	0.55
DP	80	0.86
OPS	1011	10.93
PRD	1617	17.48
O	2477	26.78
Total	9247	100

4 Dataset

Our classification experiments were performed using a proprietary dataset, provided by Mayfair Group LLC. This dataset contained accident and injury case depositions that were made available to us as a courtesy for academic purposes. This dataset contained about 350 depositions. We randomly selected 11 depositions that pertained to 2 different litigation matters, each with multiple deponent types. We processed the dataset for any noise and removed all of the content from the depositions other than the QA pairs. We ended up with a total of 9247 QA pairs. We divided them into training, validation, and test sets with percentages of 70, 20, and 10, respectively The dataset was manually annotated by the authors. Table 3 shows the (aspect/topic) class distribution for the dataset.

5 Experiment Setup and Results

5.1 Experimental Setup

We wanted to understand what content from the QA pairs would enable a classification system to capture the semantics of the conversation QA pair effectively. Is it the question, the answer, or the declarative sentence that works best? Or would a combination of these lead to the highest achievable classification results?

The declarative sentences were generated automatically by the transformation method, as described in Section 3.3. For the testing phase involving the declarative sentences input, the inference was performed using the declarative sentences as generated by the automated methods.

5.2 Results and Discussion

Table 4 shows the results for each of the 3 systems. We attained the best F1 score of 0.83 for the BERT embeddings based classifier. The classifiers based on

CNN and bidirectional LSTM with attention had poor performance with all of the input types. The BERT classifier consistently outperformed all of the other systems for all input types. For the BERT classifier, the best score was attained for the declarative sentences input. It was (pleasantly) surprising to us that concatenation of question and answer text had an inferior performance when used for classification, as compared to the declarative sentences. Equally surprising was the fact that concatenating the question and answer text with the declarative sentences had a slightly lower (non-significant) classification performance, as compared to using just the declarative sentences. The results also highlight the superior classification efficacy of the BERT based system. The sentence embeddings generated by it were semantically rich. Using them with a single layer neural network resulted in the best classification results, as compared to other systems.

Table 4. F1-scores for the different methods. Best score in bold.

	CNN	Bi-LSTM	BERT
Question (Q)	0.44	0.43	0.64
Answer (A)	0.35	0.36	0.71
Question+Answer(Q+A)	0.47	0.49	0.75
Declarative Sentences - Machine (DS-M)	0.46	0.46	**0.83**
Question + Answer + Declarative Sentences - Machine (Q+A+DS-M)	0.50	0.49	0.81

Table 5. Experiment results with additional training data using the BERT classifier.

Text used	**F1-score**
DS-M	0.83
DS-C	0.82
DS-CM	0.83

To further explore the effectiveness of the declarative sentences, we performed another experiment. In our analysis of the declarative sentences (DS-M) generated by our method in [8], we observed that a few of the generated sentences had some noise. This was due to the less than ideal fusion of the question and answer text, which was also highlighted by the authors of the work for some QA pairs. We wanted to explore whether using declarative sentences that are devoid of any noise would provide any improvement in the classification accuracy. To perform this experiment, we used the same dataset and had human annotators write the declarative sentences for the QA pairs. These annotators had learned English as their first language and were proficient in their writing ability.

We trained the BERT classifier as it was the one with the best performance. We wanted to evaluate whether training using declarative sentences written by human annotators will improve the classification on the machine-generated declarative sentences. We performed the classification on the test set that contained machine-generated declarative sentences. The training was performed using the following different texts: (1) Machine-generated declarative sentences

(DS-M); (2) Human-written declarative sentences (DS-C); (3) Concatenated[1] human-written and machine-generated declarative sentences (DS-CM).

Table 5 shows the results of the experiment. We observed that training using the additional declarative sentences written by human annotators did not make any difference, as the F1 score remained constant at 0.83. This highlights that the classifier is resilient to noise, and can perform as well with noisy text as with the human-annotated sentences. Table 6 lists the Precision, Recall, and F1-score for the respective classes in each of the three scenarios. The classification model DS-CM performs well on classes with low amounts of training data, that can be identified in Table 3. When compared with the two other models, DS-CM was relatively better, but a test of significance yielded a p-value of .14, so the difference was not significant. On the other hand, both the DS-M and DS-C models perform well in classes with large amounts of training data, and the DS-CM model is not far behind.

To understand the limitations of the classifier further, we performed another experiment using training and testing on human written declarative sentences. We wanted to explore whether the classifier performs well when it has to perform inference on declarative sentences that are devoid of noise. We trained the BERT embeddings based classifier on the dataset with human-written declarative sentences and achieved an F1-score of 0.94 on the test set. This result shows that the presence of noise in the machine-generated declarative sentences hurts the classification efficacy. We find this result very encouraging and believe that a detailed analysis of the classification results between the human-written and machine-generated declarative sentences would provide some insights in tuning the classifiers further. We plan to address this in our future work.

Table 6. Classification scores for BERT

Cls	**DS-M**			**DS-C**			**DS-CM**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
B	0.92	0.91	**0.91**	0.89	0.89	0.89	0.89	0.90	0.90
EB	0.86	0.86	0.86	0.89	0.84	**0.87**	0.79	0.87	0.83
ED	0.84	0.75	0.79	0.88	0.86	**0.87**	0.89	0.81	0.85
EC	0.74	0.81	0.77	0.55	0.69	0.61	0.80	0.89	**0.84**
PPC	1.00	1.00	**1.00**	1.00	1.00	**1.00**	1.00	1.00	**1.00**
TR	0.59	0.76	0.67	0.71	0.73	0.72	0.75	0.92	**0.83**
EE	0.83	1.00	0.91	0.71	1.00	0.83	1.00	1.00	**1.00**
IP	0.80	0.80	0.80	0.43	0.60	0.50	0.78	1.00	**0.88**
DP	0.57	0.57	0.57	0.50	0.71	0.59	0.86	0.80	**0.83**
OPS	0.77	0.79	0.78	0.75	0.84	**0.79**	0.71	0.75	0.73
PRD	0.81	0.89	**0.85**	0.83	0.88	**0.85**	0.82	0.85	0.84
O	0.84	0.76	**0.80**	0.82	0.71	0.76	0.87	0.73	0.79
Avg.	0.83	0.83	**0.83**	0.83	0.82	0.82	0.83	0.83	**0.83**

[1] The text of both the declarative sentences is used to concatenate.

5.3 Error Analysis

We chose the best performing classification system results, as shown in Table 4, and performed a detailed error analysis on the misclassifications. For the analysis, we only included classes that had either an F1-score < 0.9, or a number of misclassifications greater than a threshold of 10. Table 7 discusses the errors associated with each aspect. In future work we may reduce the number of errors.

6 Conclusion and Future Work

Traditional and state-of-the-art NLP and summarization techniques are challenging to apply to corpora that consist of question-answer pairs. Applying summarization methods as-is on QA pairs leads to less than ideal results because of their form. We used a transformation of the QA pair to a canonical form to ease the processing of such text for classification. However, our initial use of the canonical form did not improve the summarization results by much. Hence we sought a summarization method that is customized for the legal domain, but can be extended to other domains. Having a way to break a legal deposition into its constituent aspects topically would help segment the various parts of the depositions that would be further useful for summarization. To achieve this, we developed an ontology of aspects that pertains to the legal domain, especially for property and casualty insurance claims. This ontology can be expanded or modified for other kinds of cases or for different domains. For classification purposes, we have developed an annotated dataset of 9247 QA pairs for their respective aspects. This dataset helped us in training and evaluating our classifiers.

We developed three methods for classifying the aspects present in legal depositions: CNN with word2vec embeddings, Bi-directional LSTM with attention mechanism, and BERT sentence embeddings based classifier. We plan to improve and extend this work in the following ways.

1. Add more training examples to the dataset for the aspects that had classification errors due to a smaller number of training examples.
2. Provide more context to the classifier using the preceding classes from the previous 2 QA pairs [4] to classify the current QA pair, to improve the classification accuracy.
3. A framework is created as part of our summarization pipeline to group certain aspects into specific ones based on deponent type (e.g., Plaintiff:{B, EB, ED, EC, PPC, TR, IP}) that could allow us to structure a summary better in terms of aspect distribution and layout.
4. We have collected additional data for classification. Attorneys and paralegals have annotated this data. Training the system on this additional data should increase the classification score further. Accordingly, we plan to evaluate these results with the legal experts who helped us in annotating this data.
5. Develop NLP and deep learning techniques to identify segments within the legal deposition that are centered around the same topic. Using the aspects would help create these segments that can be used to generate summaries.

Table 7. Error analysis

Class	Analysis
B (Biographical)	Most of the misclassifications are attributed to the incorrect class assignment to "OPS," and "O" classes. Sentences assigned to the "OPS" class have similar words often found in the listing of skills of a person, which are used in the "B" class. This inclusion of these words without proper context results in misclassification. The declarative sentences generated by the automated methods sometimes remove the context. For example, a human-written declarative sentence such as "i left ORG165 due to difference of opinion" includes the background with not so much reliance on other sentences in the block. But in a machine-generated declarative sentence, "i did leave there due to difference of opinion" does not convey the context unless the previous and preceding declarative sentences are also included. Such sentences are often assigned to the "O" class.
EB (Event Background)	The misclassification in this class concerns the sharing of certain words in the training example with the "ED" and "EC" classes.
ED (Event Details)	The few misclassifications for this class are assigned to the "O" class. Mostly these sentences do not include the essential terms/context but are just repeated sentences by a deponent during the deposition.
EC (Event Consequences)	Training data for this class is less, making it hard for the classifier to learn words associated with this class as compared to "ED," "EB," "IP".
TR (Treatments Received)	The sharing of terms in the training data with the "PPC" class confuses the classifier re learning the correct word associations.
IP (Impact on Plaintiff)	The medical impact of the injury on the plaintiff as against the other kind of impacts is often misclassified in this class. The inclusion of terminology in these classes is the primary reason for this misclassification.
DP (Deposition Procedures)	Possible misclassifications in this class are due to a limited amount of training data and frequent use of the word "deposition," which is also found in the "B" class.
OPS (Operational procedures/ inspections/ maintenance/ repairs)	The majority of the misclassifications are assigned to the "EB" and "PRD" classes. The "OPS" and "PRD" classes are only covered as part of the witness deponent roles such as Related Organization Witness, leading to the terminology cross over in the training data.
PRD (Plaintiff-related Details)	Conversations about acquaintance with the plaintiff and other related people are mostly considered part of the "B" class which leads to most misclassifications.
O (Other)	Sentences not assigned to any of the previous classes are assigned to "O", making it a mix of many variants of terminology and context. Resulting sentence assignment errors lead to low recall.

6. Develop explainable AI methods to correlate summary sentences back to the parts in the deposition that provide their support.

Acknowledgments

This work was made possible by Virginia Tech's Digital Library Research Laboratory (DLRL). Data in the form of legal depositions was provided by Mayfair Group LLC. In accordance with Virginia Tech policies and procedures and our ethical obligation as researchers, we are reporting that Dr. Edward Fox has an equity interest in Mayfair Group LLC, whose data was used in this research. Dr. Fox has disclosed those interests fully to Virginia Tech, and has in place an approved plan for managing any potential conflicts arising from this relationship.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB (1994)
2. Ang, J., Liu, Y., Shriberg, E.: Automatic dialog act segmentation and classification in multiparty meetings. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (2005)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proc. ICLR (2014)
4. Bothe, C., Weber, C., Magg, S., Wermter, S.: A context-based approach for dialogue act recognition using simple recurrent neural networks. In: Proc. 11th Int'l Conf. on Language Resources and Evaluation (2018)
5. Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Interaction of feature selection methods and linear classification models. In: Workshop on Text Learning held at ICML (2002)
6. Chakravarty, S., Chava, R.V.S.P., Fox, E.A.: Dialog acts classification for question-answer corpora. In: Proc. 3rd Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL) (2019)
7. Chakravarty, S., Chekuri, S., Mehrotra, M., Fox, E.A.: Aspect Classification for Legal Depositions. arXiv e-prints arXiv:2009.04485 (Sep 2020)
8. Chakravarty, S., Mehrotra, M., Chava, R.V.S.P., Liu, H., Krivansky, M., Fox, E.A.: Improving the processing of question answer based legal documents. In: Proc. JURIX (2019)
9. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (2014)
10. Coster, W., Kauchak, D.: Simple English Wikipedia: a new text simplification task. In: Proc. 49th Annual Meeting of ACL-HLT (2011)
11. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967)
12. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proc. NAACL-HLT (2019)
13. Fernandez, R., Picard, R.W.: Dialog act classification from prosodic features using support vector machines. In: Speech Prosody 2002, International Conference (2002)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Ikonomakis, M., Kotsiantis, S., Tampakas, V.: Text classification using machine learning techniques. WSEAS Transactions on Computers **4**(8), 966–974 (2005)

16. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H.: Deep unordered composition rivals syntactic methods for text classification. In: Proc. ACL-IJCNLP (2015)
17. Ji, G., Bilmes, J.: Dialog act tagging using graphical models. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. (2005)
18. Jurafsky, D., Shriberg, E., Fox, B., Curl, T.: Lexical, prosodic, and syntactic cues for dialog acts. Journal on Discourse Relations and Discourse Markers (1998)
19. Kim, Y.: Convolutional neural networks for sentence classification. In: Proc. EMNLP (2014)
20. Král, P., Cerisara, C.: Automatic dialogue act recognition with syntactic features. Language resources and evaluation **48**(3), 419–441 (2014)
21. Liu, Y.: Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. In: 9th Int'l Conf. on Spoken Language Processing (2006)
22. Mark Davies: Google Books corpora (2011), `http://googlebooks.byu.edu/`, [Online; accessed 28-April-2019]
23. Mast, M., Kompe, R., Harbeck, S., Kießling, A., Niemann, H., Noth, E., Schukat-Talamazzini, E.G., Warnke, V.: Dialog act classification with the help of prosody. In: Proc. of 4th International Conference on Spoken Language Processing. (1996)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
25. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proc. EMNLP (2014)
26. Porath, E.B., Gilboa, I.: Linear measures, the Gini index, and the income-equality trade-off. Journal of Economic Theory **64**(2), 443–467 (1994)
27. Porta, A., Guzzetti, S., Montano, N., Furlan, R., Pagani, M., Malliani, A., Cerutti, S.: Entropy, entropy rate, and pattern classification as tools to typify complexity in short heart period variability series. Transactions on Biomedical Eng. (2001)
28. Rish, I., et al.: An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence (2001)
29. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics **21**(3), 660–674 (1991)
30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
31. UCSF Library: Truth Tobacco Industry Documents (2002), `https://www.industrydocuments.ucsf.edu/tobacco`
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems 30. pp. 5998–6008. Curran Associates, Inc. (2017)
33. Venkataraman, A., Ferrer, L., Stolcke, A., Shriberg, E.: Training a prosody-based dialog act tagger from unlabeled data. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. (2003)
34. Webb, N., Hepple, M., Wilks, Y.: Dialog act classification based on intra-utterance features. cs-05-01. Dept. of Computer Science, University of Sheffield, UK (2005)
35. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proc. ACL. (2016)

A citations network for legal decisions

Damián Langone, Alfredo Fulloni, and Dina Wonsever

Facultad de Ingeniería, UDELAR, Uruguay
fing.edu.uy/inco/investigacion/grupos/PLN

Abstract. In this paper we present different tools used to recognize and extract references to legal entities found all over the texts of legal sentences of the Uruguayan State. The work described is part of a broader project focused on increasing the value of this information, through the generation of an interactive graph of references with the aim of making the information contained in judicial sentences available in a more user-friendly format. In this paper we present the results obtained after working with different tools, such as a rule-based system and statistical methods such as NeuroNER, Conditional Random Fields and SpaCy's NER. To carry out the training and evaluation of those models, a dataset provided by the National Jurisprudence Base was used. This dataset consists of texts of second instance judicial sentences. The results obtained are considered good, and serve as a benchmark to keep working in legal texts on the Spanish language. In addition, having carried out this project prompted a discussion about how mature NLP tools are to work on legal texts. It is concluded that the extraction of references in the legal field is often a complex task due to the jargon used, but that it increases the value of the information, making it possible to use the extracted data for different global analysis and as a basis for other processes.

Keywords: Artificial Intelligence · Law · Natural Language Processing · Named Entity Recognition · Information Extraction.

1 Introduction

In recent times the speed with which technology has been used in different areas has rapidly increased. Currently some uses can be seen in sectors in which a few years ago it was unthinkable. An example of that is the legal sector, where technology is becoming increasingly important. This is due to the multiple tools available to process large amounts of textual data and optimize its use.

This work seeks to apply the tools available for Natural Language Processing, with the aim of being able to provide solutions and advances for information management over judicial sentences. A determining factor present in judicial decisions, and on which emphasis is placed in this research work, is the one corresponding to the references between legal entities.

Therefore, this work focuses on generating different tools with the capacity of automatically recognizing and extracting references between legal entities within

a legal text. By doing this, we also pretend to test the usefulness of NLP tools in this area, and to study how they can facilitate different tasks.

The examples in 1 show some of the studied references. It is easy to see and imagine the multiple formats they could take.

- *"...Therefore, if the assets were community property, the claim of 50 % that had been claimed against AA should be protected, in which regard it was not possible to extend what was decided by* ***sentence No. 179/2007 of the T.A.C. 1st*** *that had been pronounced ..."*
- *"Adheres to the classical position that maintains that in the hypotheses provided for by* ***arts. 24 and 25 of the Constitution****, public servants are not entitled to be sued by injured third parties, and only the State can repeat against them in case of being held responsible..."*

Fig. 1. Examples of legal references

2 Problem context

In a collaborative project with the Supreme Court of Justice of Uruguay, the sub-language of the references was analyzed, different strategies of automatic detection and interpretation of the references were tested and the complete implementation of the process was carried out. We worked on the Base de Jurisprudencia Nacional (BJN)[2], an open access public database containing more than 70,000 court rulings.

More than one million references were detected and processed, and a graphical representation of the internal links of the BJN was built. The Neo4j network oriented database manager was used for this purpose. This representation of the BJN allows to process efficiently several queries and do different analyses generating a better interaction with the final user.

In this paper we describe the automatic extraction of most of these references, focusing on the detection and interpretation of references between legal objects and on the natural language processing methods and tools that have been used. One conclusion of our work is that human language technologies are mature enough to provide interesting and useful results in the legal domain.

The reference extraction problem was approached as a Named Entity Recognition (NER) problem. At first, we started by opting for a rule-based system approach, but it didn't scale as we wanted. So, looking for a more scalable and automatic solution, different approaches based on machine learning were experimented. In order to train different models, it was necessary to create an

annotated corpus from the dataset taken as reference. We focused our research on the extraction of the following legal entities:

Judicial Sentence Resolution of a legal nature that expresses a final decision on a process.
Article Provisions, generally numbered consecutively, that make up a legal body. These articles can be divided into numerals, subsections and literals.
Law Rules or norms established by a higher authority to regulate, in accordance with justice, some aspect of social relations.
Decree Decision of the Council of Ministers, or of an equivalent entity, which approves provisions of a general nature.
Case File Collection of documents, procedural documents and rulings, relating to a litigation initiated by a civil, commercial or social jurisdiction, within a file in which the different events of the process are mentioned.

The biggest challenge we faced in carrying out this project was the large number of different formats in which references to different legal entities can be found. In 2 there are some examples of references to articles. The main problem is that although references to legal entities such as laws, court decisions, decrees and others have drafting rules, in practice the standardized formats are not always followed, so the quality of the writing depends on both the author and the date of publication.

I) `Art. 1955 of the Civil Code`
II) `Articles 248 to 261 of the CGP`
III) `Article 72 of the Childhood and Adolescence Code`
IV) `art. 46 CP, nos. 7th and 13`
V) `a. 61 of L. 15,750`
VI) `second paragraph of art. 4th of Law 1601`

Fig. 2. Examples of Articles formats

3 Related works

When considering the reference extraction task as a NER problem, it is necessary to take into account the different options used for this type of problem. Broadly speaking, two approaches can be identified: rule-based systems and statistical methods. The main antecedent on which this work is based uses a rule-based system through regular expressions for the extraction of references (Sadeghian et al. [14]). They do not provide any evaluation measures and they refer to the method proposed by Tran et al. [17]. In this work they are based on the hypothesis that references in the legal domain have their own structure and differ from references in other domains. They distinguish two parts by which a

reference is composed: the "position" and the "content", the first being interpretable by regular expressions. In turn, they define several types of classification for these text segments, the first corresponds to well-formed positions, that is, they contain an identifiable term (such as the word article), followed by a number. This type contains all the necessary information to identify the reference. Other types taken into account are anaphora and coordinate positions, that is, when several elements are referred together. The study was carried out on the Japanese National Pension Law (JNPL), on a total of 99 articles and 748 citations and for the task of reference recognition they obtained a 96.18 % accuracy.

In the statistical methods area, different variants of Conditional Random Fields (CRF) models stand out. In a work with similar tasks to the present one (Leitner, Elena et al. [9]), CRF models and BiLSTM (Bi-directional Long Short Term Memory) neural networks were taken as the main approaches. The study was carried out on a dataset of judicial sentences of the German court, which contains around 67,000 sentences and 54,000 annotated entities (in addition to legal entities, more generic categories such as city, brand or institution were taken into account). In total, 12 models were tested and one of the highlights was a variation of CRF with 93.23% of F1 score. Regarding the BiLSTM models, the one that obtained the best performance was a BiLSTM used with Character Embeddings, which obtained 95.46% of F1 score.

In the work presented by Angelidis et al. [3], one of the recognized and extracted categories is legal entities references. This work is based on the Greek language and uses legal documents published by the Greek. In this project, emphasis was placed on BiLSTM models, testing three variants of it. One of them uses a BiLSTM layer together with Word Embeddings and a Logistic Regression layer with Softmax activation which is in charge of assigning the probabilities that each token belongs to certain class. The second variant adds another BiLSTM layer while the third variant uses a BiLSTM layer and replaces the Logistic Regression layer with a CRF one. The model that obtained the best results was the one with two BiLSTM layers, obtaining 88% in macro F1 measure.

4 Methodology

4.1 DataSet

The dataset available for this project consisted in a total of 72,358 second instance judgments belonging to the Uruguayan National Jurisprudence Base. The National Jurisprudence Base (BJN) consists in a storage system of judicial sentences issued by entities such as the Supreme Court of Justice, Appeals Courts and First Instance Legal Courts. It was created with the objective not only of modernizing the jurisprudence of our country, but also of being able to provide a tool to the population so that they can carry out different searches on judicial information. Today, anyone is able to do searches in the BJN from its website. Because the information we have consisted of plain texts without

annotation, an annotated corpus had to be created in order to use supervised learning methods.

The data set used for the training and evaluation of the models was generated semi-automatically. To do this, we created regular expressions from an analysis performed on a random set of 100 statements. Some of the created rules can be seen in 4.1. Then, those regular expressions were applied to a random set of 1000 judicial sentences, generating as output files in the format used by the manual annotation tool [15]. This allowed the annotations generated by using this web tool to be manually reviewed, correcting errors if any. Performing the annotation of the dataset in this way allowed speeding up the task since it facilitated the identification of references within the texts. However, several corrections had to be made due to the great heterogeneity of formats and legal entities present in the texts.

```
# JUDICIAL SENTENCES
sent_name = '(sentencia|sent\.|resolución|providencia)'

sents_name = '(sentencias|sents\.|resoluciones|providencias)'

date_num = '\d{1,4}(?:/\d{2,4})?'

court = '(Jueza.{0,100}?\s+(?:Turno|T\.)|Juez.{0,100}?
\s+(?:Turno|T\.)|Juzgado.{0,100}?\s+(?:Turno|T\.)
|Tribunal.{0,100}?\s+(?:Turno|T\.)
|Dr\..{0,100}?\s+(?:Turno|T\.))'

sentence = sent_name + '\s*(?:\w{0,15}?\s){0,4}?
(?:Nro\.?|No\.?|N[°]?)?\s*(\d{1,4})' + date +
'?(?:.{0,100}' + court + ')?'

sentences = sents_name + '\s*.{0,40}?,?(?:\s*(?:Nros?\.?
|Nos?\.?|N[°]?)?\s*' + date_num + '\s*(?:,|;|y)?)+'

# CASE FILES
case_file = '(?<=[^\w])(ficha|Fª|Fa?\.?|I\.?U\.?E\.?
|expediente|exp?\.?)\s*:?-?\s*(?:Nro\.?|No\.?
|N\s*(?:°|)?)?\s*(\d+\s*(?:\.|-|{|/)?\s*\d*\.?\d*)\s*(?:/|-)
\s*(\d\.\d{3}|\d{2,4})'
```

Fig. 3. Regular expressions for Judicial sentences and Case Files

The next step consisted of performing an 80/20 split on the set of 1000 sentences annotated in BRAT format, allocating 20% to the evaluation set, while 80% was again divided in the same way into a training set and a validation set. Therefore, 200 sentences were allocated to the evaluation set, 640 to the training

set, and the remaining 160 to the validation set. In the table 1 is shown the total references contained in each set.

Table 1. Number of references per set

	Article	Judgment	Decree	Law	Case File	**Total**
Train	5709	2406	444	1361	846	10766
Validation	1447	569	94	443	239	2792
Test	1563	736	111	481	242	3133
Total	8719	3711	649	2285	1327	16691

4.2 Reference Extraction

Nowadays most of the NER tools do not support overlapping annotations, so in order to solve this part of the problem the annotations had to be filtered to have only those annotations to complete entities. We define a complete entity as those text segments corresponding to a reference, which contain all the necessary properties to unambiguously identify the cited legal entity.

The necessary properties for a reference belonging to a legal entity to be considered complete are detailed below:

Article: It is necessary to have at least its number and source, and may also have incisions, numerals and literals.
Judicial Sentence: It is necessary to have the number, the venue, the year and the headquarters, although those that had at least number and year were considered valid.
Decree: It is necessary to have the number and year.
Law: It is necessary to have the number and/or name that identifies it.
Case File: It is necessary to have the number and year.

NeuroNER [4] It is a Named Entity Recognition system developed in Python using TensorFlow [11]. This system was included in the extraction process in order to be able to train it taking the examples of the different legal objects as named entities. NeuroNER is based on a variant of recurrent neural networks, called Bidirectional Long Short Term Memory (Bi-LSTM), used together with vector representations of words (Word Embeddings) enhanced with vector representations of characters (Character Embeddings) and CRF (Conditional Random Fields).

One of the main reasons this tool was chosen is that, even though NeuroNER works with data sets in CoNLL [16] format with BIO or BIOES labeling, it also supports annotated data sets in BRAT format performing the transformation between this and the CoNLL format. In this way, it was possible to reuse the

work done with the BRAT tool and the set of manual rules developed through regular expressions that was mentioned before.

It should be noted that before using this tool, an analysis of its sentence split component was performed, which is configurable to use external tools from both Stanford [10] and SpaCy [6]. In our case, we made a modification to this component to be able to use the Freeling tool [12], which turned out to be the one with the highest correctness for the Spanish language.

.

I) `IUE 0227-000149/2013`
II) `FICHA N° 370-359/2003`
III) `Ley No. 19.090`
IV) `Ley Orgánica Policial`

Fig. 4. Examples of Law and Case File formats

The best results were obtained by training several models combining those entities whose references did not overlap and whose number of occurrences in the training set were not so distant. This was done due to the great imbalance that exists in the corpus between the different entities, which can be seen in the table 1. Therefore, three models were trained, one in charge of extracting references to Articles (including the text of their source), another in charge of extracting references to Judgments and Laws and the last one in charge of extracting references to Decrees and Case Files.

The legal entities for which the best results were obtained were Law, Case File and Article. The figure 4 shows that the references to the first two, in most cases, are not found in very different formats because when citing a law, the number of the same is generally given (example III) and in some occasions also the name by which it is known (example IV), while the references to case files in most cases are present in the IUE (Unique File Identifier) format as can be seen in examples I) and II).

Conditional Random Fields [8] Another approach used in order to be able to study the patterns in which the references are presented in this type of texts was the scikit learn [13] implementation of Conditional Random Fields [1]. This was proposed because using neural networks does not allow us to analyze what characteristics are being taken into account. It sounds logical that if, for example, in a piece of text there is an occurrence of the word “decree” or “law” followed by a number, it is surely a reference to those legal entities. This is the reason why we decided to experiment with a machine learning method in which the attributes were explicit given.

One of the main characteristics of this method is that the context in which a token is located is taken into account for its correct classification. In addition, since the user defines the attributes to be used, it presents more flexibility to avoid falling into erroneous assumptions which can occur when using neural networks.

The process consisted in the selection of attributes followed by the calculation of the value for each attribute among all the tokens in the statements that make up the training set. This included the use of the Freeling tool both to recognize sentences and tokenize them and to obtain the grammatical categories (POS-tag) of both the token in question and its neighbors. Regarding the selection of attributes required to train a CRF, those that provide information about the token, its composition and its context were taken into account. This is why the recommendations indicated in Konkol [7] were followed, where it is suggested to use spelling attributes, suffixes and prefixes, grammatical categories and lemma, among others. Also, the author suggests defining a window to extract information about the context of the token. In this case, a window of size 1 was considered, which covers the token immediately before and after the one considered (as long as it is not at the beginning or end of a sentence).

SpaCy [6] Another tool used was the NER tool present in the SpaCy pipeline. SpaCy is an open source library made up of various tools and used for natural language processing. It uses CNN (Convolutional Neural Networks), which are being used more and more in the field of natural language processing and improving results in many areas of it, which can lead to better results than NeuroNER. More precisely, deep convolutional neural networks are used, with residual connections. Neural networks with residual connections are those that have connections between non-contiguous layers, in order to avoid the problem of gradient vanishing, which causes the gradient to stagnate and the network cannot continue the training. This is accomplished by reusing the activations from a previous layer until the adjacent layer learns its weights.

For the training, we proceeded to create a new "blank" model for the Spanish language, and disable the other components of the pipeline, in order to train only the NER. The procedure consisted of carrying out different trainings, varying the dropout as a regularization technique and evaluating the results of each iteration on the validation set. Regarding the stop criteria, in addition to a maximum of 100 iterations, the same criterion was used as in NeuroNER, which consisted of cutting the training if in 10 consecutive iterations the value of the F1 score was not improved over the validation set.

In the figure 5 the values of the F1 score can be observed as the training progressed for each dropout value. It can be seen that as the dropout increased, the convergence became slower, while the lower the dropout was, the model quickly converged culminating the training due to the stopping criterion applied to the F1 score, as it can be seen in the case of dropout 0.1 where the training culmi-

nated at 25 iterations. Finally, the best value for the F1 score on the validation set was 90.69% and it was reached in iteration 33 with a dropout value of 0.35. This model was chosen when calculating the different metrics on the evaluation set.

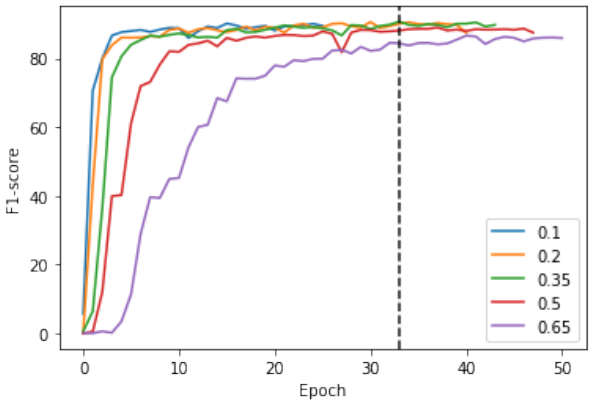


Fig. 5. F1 score as a function of epochs in validation set

5 Evaluation and Results

This section details the results obtained after evaluating the statistical methods mentioned in the previous section. All the results were obtained on the evaluation set, from the best models obtained according to the validation set in each case. The performance is measured in terms of precision (P), recall (R), and F1-measure (F_1).

Table 2. ML models evaluation

	NeuroNER			**CRF**			**SpaCy**		
	P	R	F_1	P	R	F_1	P	R	F_1
Article	88.75	87.84	88.30	89.42	86.38	87.87	93.05	91.18	92.10
Judicial Sentence	77.65	70.79	74.06	78.96	74.76	76.80	78.11	76.60	77.35
Decree	83.51	72.97	77.88	81.66	74.24	77.77	87.80	92.30	90.00
Law	92.84	91.68	92.26	85.03	88.65	86.80	87.95	94.80	91.25
Case File	95.24	85.27	88.40	98.70	94.60	96.61	85.14	95.51	90.03

As shown in table 2, NeuroNER and CRF had quite similar results, with CRF being slightly better, with the exception that to train NeuroNER it was necessary to separate the entities and train three different models. This was done due to

the poor results obtained while training only one model with NeuroNER. The results were much lower and this was the solution found to improve the results. On the other hand, it is clear that the model trained with SpaCy, surpasses the other two models, being able to train all the entities together and without the need to perform attribute engineering as in the case of CRF.

.

I) `Reference: Articles 34 to 38 inclusive of the Civil Law Treaty of 1889`
`Prediction: Articles 34 to 38`
II) `Reference: JUDGMENT N°304 Montevideo, December 9, two thousand eleven.`
`COURT OF CIVIL APPEALS, FOURTH SHIFT`
`Prediction: Correct`

Fig. 6. Examples of predictions by SpaCy's trained model

A further point is that the entities which obtained the best results were those that have the most standardized formats, that is, those whose forms in which they appear do not differ too much between different texts. Due to the amount of necessary properties to fully identify a reference to a judicial sentence, it can be seen that the extraction of those entities was the lowest for all models and could never get over a F_1 score of 80%.

6 Citation Network construction

The next step after completing the reference extraction stage consisted of creating a database of graphs in order to store the information on these references and their relationships. These databases are characterized by using graph structures to represent the information, that is, they represent the data in the form of nodes connected by edges, where these represent the relationships between the different data. This form of representation favors those scenarios in which the elements that make up the information are closely related, because the structure of graphs facilitates their interpretation.

For data storage, the Neo4j tool was chosen, which is implemented in Java and has an OpenSource version. We define seven types of nodes with different attributes, which can be present both as attributes of the node and attributes of a relationship with it, as long as said information has been extracted in the process of extracting the references. The nodes and attributes are those mentioned in the section 4.2, added to the nodes corresponding to the written sources of the articles (for example the Civil Code of example I) in 2), and to the nodes called Doctrines that refer to referent persons in the legal area whose opinions are taken as reference in judicial decisions (this type of legal entity was extracted with a different approach from the rest).

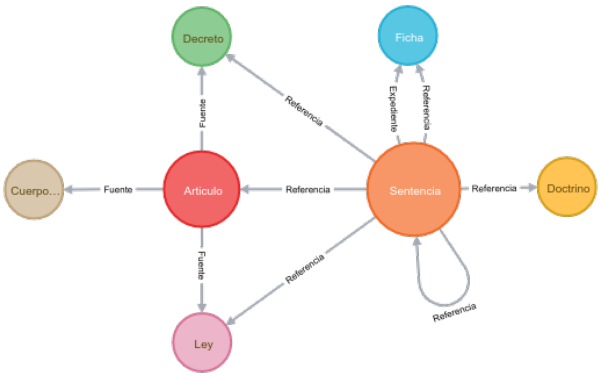


Fig. 7. Graph structure

After completing the previous stages, we had a trained model capable of extracting text segments corresponding to references to legal entities with a micro-F1 measure of 87.63 %. As only the entire text segment was available, post-processing was required to distinguish the different parts that make up a reference. These parts depend mainly on the type of legal entity referred to, including for example number and date. The properties of the legal entities taken into account are those described in 4.2.

The general data loading process then consisted of developing a system that is responsible for extracting the existing references in a legal text from the BJN database, processing them obtaining the most important properties according to their type and then adapting that information to a query in Cypher language (query language used by Neo4j), in order to create the corresponding nodes and relationships in the database through the Py2Neo library.

After executing the complete system on all the judicial sentences stored in the BJN, a citation network with 250343 nodes and 1116278 relationships was obtained. In 8 we can see a subgraph of the generated reference graph, which shows the value that is given to the extracted information, transforming it from plain text files to an interactive and more intuitive tool for not specialized users.

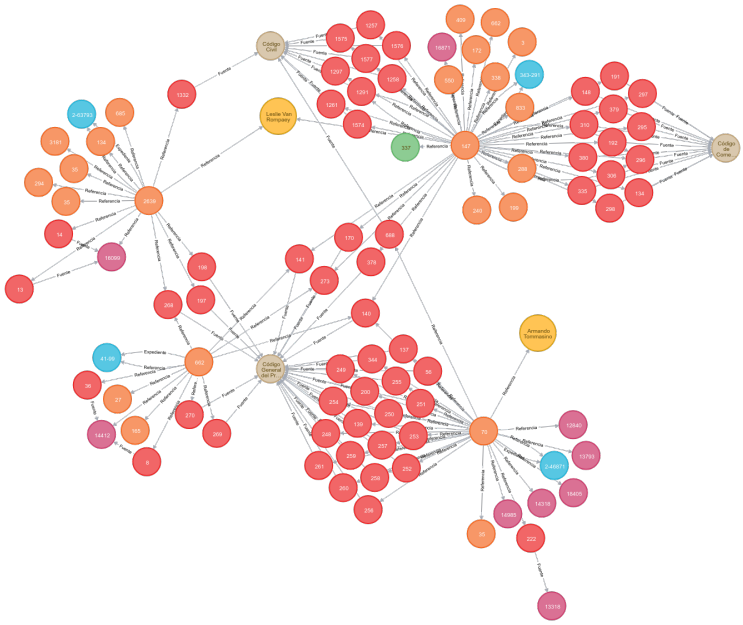


Fig. 8. Subgraph example

7 Discussion, directions for future work

With this work we have managed to show the potential and maturity of some NLP tools applied to texts in the legal area. It could be seen that, starting from an annotated dataset, with some research and practice good results can be achieved. However, we must emphasize that the usefulness of these tools is biased by factors such as the technical jargon and syntactic peculiarities used in the legal domain. Domain adaptation is a trending topic in Natural Language Processing and new methods such as Transfer Learning, in a Neural Networks setting, are obtaining impressive results.

The good results obtained till now encourages us for seeking better results for the references task and also for enlarging the processing of legal text with NLP technology. In the first option, there are some interesting different possibilities. One of them relates with testing different hyperparameter configurations in the models used in this paper, with the hope to better fit the data and perhaps increase the overall performance. Another possibility, following the thread of the annotations, is to add new entities obtained by anaphora expansion, a feature that is not available in the present system. If we wish to enlarge the field of NLP applications, the spectrum of options is very wide. In first place, after carrying out this work, and in conjunction with specialists in the legal area, the possibility of applying what was done to generate a spelling check tool was raised. This tool would complement the work of the writers, with the aim of standardizing the

formats used for legal references, which in turn would facilitate the application of ML methods in the future. So, a very related feature would be to treat the citations sub-language as a controlled language and to spell check in real time all references instances.

With a more general perspective, an increasing number of NLP application to the legal domain are being developed. In our case, and on the same court decision dataset, other tasks are being developed. One of them is the anonymization task [5], aiming to obfuscate private personal data in criminal cases, allowing to publish the information about a case without injuring the participants' privacy rights. Other applications on the same data are being sought now; one example relates to predicting the outcome of a trial from the past history registered in the BJN base. The interesting point is that the different tasks collaborate between them, accumulating "wisdom" about the domain.

Acknowledgements

We appreciate the invaluable contribution of the Judicial Power in Uruguay, especially the Jurisprudence and Technology branches, which made the project possible by supplying the data and providing constant advice and support. This research was partially supported by the National Agency of Research and Innovation (ANII-FSDA-1431380).

References

1. Crfsuite. https://sklearn-crfsuite.readthedocs.io
2. National jurisprudence base. http://bjn.poderjudicial.gub.uy/
3. Angelidis, I., Chalkidis, I., Koubarakis, M.: Named entity recognition, linking and generation for greek legislation. In: JURIX (2018)
4. Dernoncourt, F., Lee, J.Y., Szolovits, P.: NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. Conference on Empirical Methods on Natural Language Processing (EMNLP) (2017)
5. Garat, D., Wonsever, D.: Towards de-identification of legal texts (2019)
6. Honnibal, M., Montani, I.: spaCy: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017)
7. Konkol, M., Konopík, M.: Named Entity Recognition. Phd study report, Západočeská univerzita v Plzni (2012)
8. Lafferty, J., Mccallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the Eighteenth International Conference on Machine Learning pp. 282–289 (01 2001)
9. Leitner, E., Rehm, G., Schneider, J.: Fine-Grained Named Entity Recognition in Legal Documents, pp. 272–287 (11 2019)
10. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014), http://www.aclweb.org/anthology/P/P14/P14-5010
11. Martín Abadi, Ashish Agarwal, P.B.E.B.Z.C.C.C.G.S.C.A.D.J.D.M.D.e.a.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org
12. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA, Istanbul, Turkey (May 2012)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Sadeghian, A., S.L.W.D.e.a.: Automatic semantic edge labeling over legal citation graphs. Artificial Intelligence and Law **26**, 127–144 (2018)
15. Stenetorp, P., Pyysalo, S., Topic, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for nlp-assisted text annotation. pp. 102–107 (04 2012)
16. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. p. 142–147. CONLL '03, Association for Computational Linguistics (2003)
17. Tran, O.T., Ngo, B.X., Nguyen, M.L., Shimazu, A.: Automated reference resolution in legal texts. Artif. Intell. Law p. 29–60 (2014)

A study on a trial to anonymize plain Japanese precedents by using Machine Learning Technology

Masakazu Kanazawa[1], Atsushi Ito[2], Kazuyuki Yamasawa[3], Takehiko Kasahara[4], Yuya Kiryu[5], Fubito Toyama[1]

[1] Graduate School of Engineering, Utsunomiya University, 7-1-2 Yoto, Utsunomiya-shi, Tochigi, 105-0123, Japan, kanama2591@gmail.com, fubito@is.utsunomiya-u.ac.jp
[2] Faculty Economics, Chuo University, 742-1 Higashinakano Hachioji-shi, Tokyo, 192-0393 Japan, atc.00s@g.chuo-u.ac.jp
[3]TKC Corporation, Karukozaka-MN Bldg. 2-1 Ageba-Cho,Shinjuku-ku,Tokyo,162-0824, Japan
[4] KDDI Corporation, 3-10-10 Iidabashi, Chiyoda-ku, Tokyo, 102 8460 Japan, yamasawa-kazuyuki@tkc.co.jp
[5]Toin Yokohama University, 1614 Kuroganecyo, Aoba-Ku, Yokohama-shi, Kanagawa, 225-8503, Japan, kasahara@toin.ac.jp

Abstract. In recent years, several studies on incorporating information communications technology (ICT) in courts have been reported. A judicial system powered by ICT is referred to as a "cyber court," and it is intended to allow anyone to easily participate in the trial. However, in the Japanese court system, certain personal names, company names, etc., are hidden in the judgment sentence due to privacy concerns. This anonymization work is done manually and requires a large amount of effort, resulting in less than 1% of cases being published. In our previous work, we used a neural network with part-of-speech tags (POS) to detect confidential words and applied the method to anonymized precedents. Then, About 60% of the cases showed a confidential word detection rate of 70%–100%. Herein, we mention the result of applying this method to the plain precedents (not anonymized). However, the result was a recall rate of ≥70% was found in 33% of the total. We analyzed the result and found the reason for the low accuracy to detect the confidential words. We think it may be possible to solve this problem to upgrade the POS function and performed a simulation. Then, we had a confident to increase the accuracy to detect confidential words in a plain precedent.

Keywords: Cyber court, Anonymization, Confidential words, part-of-speech tags

1 Introduction

The globalization of the economy, international trade, and deputes internationally present new demands on judiciaries. At the same time, advances in information communications technology (ICT) offer opportunities for judicial policymakers to make justice more accessible, transparent, and effective. Jurisdictions in many countries have aimed to follow economic growth and political changes to allow anyone to easily access judiciary documents by incorporating ICT. Such justice systems empowered by ICT

are called “cyber courts.” A pioneering study of a cyber court system is Courtroom 21 [1], which started in 1993 at the College of William & Mary as a joint project between the university and the National Center for State Courts in the U.S.A.

In Japan, a prototype for the first civil trial was developed at Toin University of Yokohama in 2004 [2, 3], which verified its effectiveness, particularly in the Japanese citizen judge system [4]. Furthermore, an experiment with a remote trial was also conducted [5]. “Investments for the Future Strategy 2017” by the Japanese cabinet office includes ICT conversion of trials to accelerate and improve their efficiency [6].

According to a newspaper report in August 2019, the Japanese government has decided to create a database of precedents finalized in courts [7]. Such a database will be useful to judge standard consolidation fees and the compensation for damages in new cases.

From the perspective of developments in big data, this field is expected to develop further. Easy access to judicial documents, such as precedents, from web sites is essential to realize access to justice.

However, most precedents are not accessible to the public on Japanese court web pages to protect personal information. Confidential words (e.g., personal, corporate, and place names) in opened precedents are replaced by abstract symbols, such as a single uppercase letter “A” to hide the information. Since this operation is manually performed, it is time consuming and taxing.

Therefore, automatic prediction of confidential words would be an ideal way to solve this problem. There are some methods to extract confidential words such as named entity extraction using machine learning and CRF (Confidential Random Field). CRF is mainly used to extract the named entity, such as person names, place names, organization names, that are included in documents. To take advantage of the surrounding context when labeling tokens in a sequence, a commonly used method is CRF, first proposed by Lafferty et al. in 2001. It is a type of probabilistic graphical model that can be used to model sequential data, such as labels of words in a sentence [23]

However, neural networks have dramatically advanced in the field of natural language processing (NLP) in recent years. Studies that derive a vector considering the meaning of words and that predict words are actively ongoing [8]. We have already reported that a bi-directional long short-term memory (LSTM)-left–right (LR) model was effective in predicting target words in Japanese precedents; however, in cases where the target word was confidential word, poor results were obtained [9]. Therefore, we analyzed the algorithm of our model and improved it. In general, words obtain their meaning and part-of-speech (POS) tags from a dictionary. We found almost all the confidential words had POS tags of proper nouns. Therefore, if the POS tag is added to words in the neural network, learning is more effective and the accuracy is improved. In this study, we propose a new method to improve the ability to detect confidential words by combining a neural network with POS tags. We obtained an 88% improvement for detecting confidential words (CW_PPL) compared to the previous model.

By applying this method to actual precedents after anonymization, about 60% of the cases demonstrated a recall score (the ratio of correctly predicted positive observations to total tagged observations) of confidential words in the range of 70%–100%.

However, issues in some cases resulted in recall scores that decreased. In particular, some person names and long addresses weren't recognized as confidential words. The problems of some unrecognized addresses were solved by improving the preprocessing algorithm. In this paper, we first explain the basic idea of our technology to anonymize confidential words in Section 2. In Section 3, we review the previous study[14] and explain the problems, ①clarify the evaluating threshold to find the confidential words, ②evaluate the performance to apply our technology to detect the confidential words in plain precedents. In Section 4, we introduce related works. In Section 5, we described the evaluation method using a ROC (Receiver Operating Characteristic) score to detect confidential words. In Section 6, we show the result to apply our technology to plain precedents. In Section 7, we analyze the result and propose the solution to increasing the accuracy to detect confidential words and display the effectiveness of our idea by simulation. Finally, we state our conclusion in Section 8.

2 Anonymization of confidential words

Some judicial precedents have been made available on the website of courts [10], wherein confidential words have been replaced with an alphabet character (in paid magazines and websites, Japanese letters are sometimes used), Fig. 1 shows an example of a replaced word. This process is manually performed. If these replacements are done using a proper noun dictionary, enormous registration and update work will be required, and the confidential words sometimes have ambiguous meaning, making it difficult to distinguish them. Therefore, anonymization is performed by legal experts who manually consider each context.

To solve this problem, we proposed a method where a neural network manually learns the anonymized precedents and predicts confidential words in the precedents.

In particular, precedents that have been anonymized by hand to date are used as true data for learning data. When an unprocessed precedent is an input in this system, this system automatically anonymizes confidential words. We aim to build a system useful for this task. Fig. 2 shows the outline of the confidential word prediction process. We defined the specific task as to "determine from the surrounding words of the target word whether the target word should be a confidential word." As a result, the bi-directional LSTM-LR model was shown to effectively imitate human work [9].

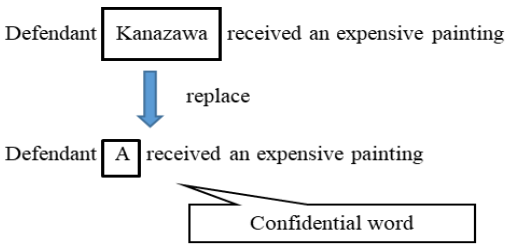


Fig. 1. Example of replaced word

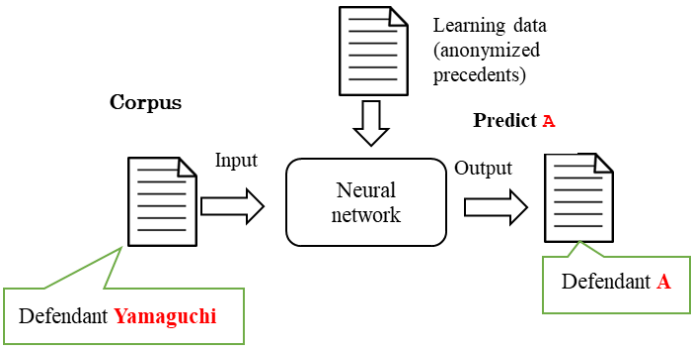


Fig. 2. Anonymization system

3 Previous research and issues

Learning using neural networks is widely employed in the field of NLP. As is well-known, word embedding is effective as a method of handling semantic information of words. Therefore, we decided to study a method to express the precedent as an embedding vector, input it into a neural network, and predict confidential words as target words. As a result of studying a model that combines Continuous Bag of words (CBOW) that was shown to be effective for predicting target words, the bi-directional LSTM-LR model with the best prediction result was adopted [9]. In the experiment, we used 50,000 judicial precedents for training data and 10,000 judicial precedents for test data from Japanese judgment sentences recorded from 1993 to 2017. A database of these precedents was provided by TKC Co. [11].

As an evaluation method, we used the perplexity (PPL) that was used in previous studies for predicting the next word. In NLP, PPL is a measurement of how well a probability model predicts a sample. In the context of NLP, PPL is a method to evaluate language models [12, 13]. It represents the number of prediction choices that are narrowed down in neural networks. The smaller the value, the better the prediction results. PPL is defined as follows:

$$PPL = 2^{-\frac{1}{N}\sum_{i=1}^{N} log_2 p(w_i)} \qquad (1)$$

In Eq. (1), $p(w_i)$ is the probability, and N is the total number of the words.

CW_PPL is the average PPL for the test data that reflects the perplexity of the confidential words, while PPL is the average for all the test data.

3.1 Issues in the experimental result of the bi-directional LSTM-LR model

In results, the PPL score was 4.7 and the CW_PPL score was 37.3. The experiment showed that our proposed model using a neural network effectively predicted the target words. Nevertheless, the probability of predicting the confidential word was about 1/10

of the probability of predicting the target word, and sufficient accuracy was not obtained. To solve the problem, we need to review the algorithm to preprocess the corpus before inputting into the neural network, because the difference in scores between PPL and CW_PPL was too large.

3.2 Combining bi-directional LSTM-LR with POS tag to improve the CW_PPL score

On investigating the algorithm of our model, in general, we found that words obtained their meaning and POS tags from the dictionary. We found that almost all confidential words had POS tags called proper nouns. Therefore, by adding POS tags to words in a neural network, the learning and the detection ability of confidential words can be improved. The most popular tagging tool for Japanese sentences is MeCab, which is a well-known morphological analyzer and it uses CRF for parameter estimation. Therefore, it has good performance rather than the other tool such as “Chasen” So, we adopted the MeCab (version 0.996).

The outline of the proposed model is shown in Fig. 3. When the Japanese precedents (corpus) is input into "MeCab," "MeCab" separates the words with space and tag them (POS). If the confidential word appears in the precedents, proper noun, e.g., the place name or the personal name is tagging to the confidential word. Then, merging them into the input data (word) for bi-directional LSTM via an embed-ding vector.

We experimented using the proposed model combined with the POS tags. We used 10,000 judicial precedents for the training data and 5,000 judicial precedents for the test data from 2013 to 2017.

The results of this experiment in comparison with the results of only the bi-directional LSTM-LR model using the same input corpus are shown in Table 1.

The PPL score showed a 23% improvement in accuracy over the previous model (bi-directional LSTM-LR), and the CW_PPL score also showed a 30% improvement.

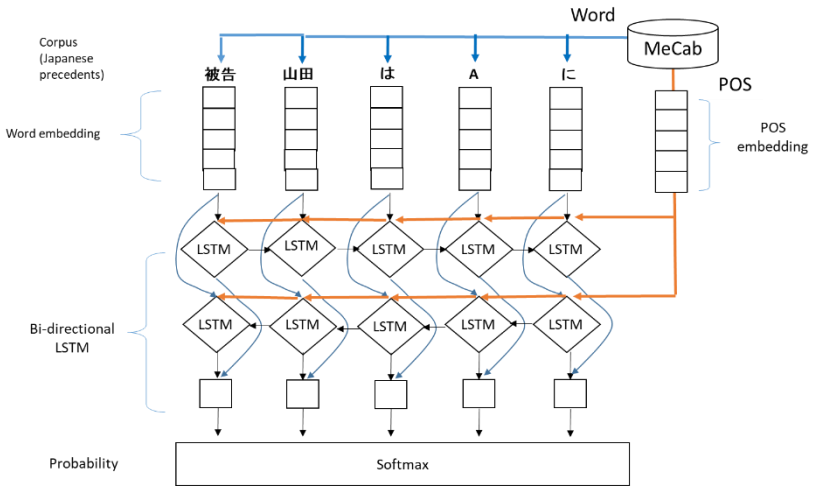


Fig.3. Outline of the proposed model.

Table 1. Proposed model with POS tag experimental results

	Proposed model combined with POS tag	Bi-directional LSTM-LR
PPL	4.1	5.2
CW_PPL	28.6	40.7

Therefore, we found that the bi-directional LSTM-LR model with POS tags was very effective in predicting confidential words. However, the CW_PPL score needed further improvement.

3.3 A further solution to improve the CW_PPL score

Before learning the input corpus using neural networks, reprocessing the corpus is necessary. For example, many punctuations marks used for separating words and phrases, such as [] and (), often appear in Japanese judicial precedents. These marks are only noise for learning, and therefore, we omitted them.

However, the punctuation mark ". " was not omitted to preserve the flow of the context. Preprocessing was also performed in the first experiment.

In Japanese precedents, confidential words are replaced by not only half-width uppercase letters but also full-width uppercase and lowercase letters. In the previous algorithm, only single half-width capital letters were replaced with the half-width uppercase letter "A." Therefore, when the lowercase letter "b" appeared in the precedent, it was not replaced by "A." Thus, we did not recognize "b" as a confidential word (Fig. 4).
Therefore, we improved this algorithm by stating that if both single half-width and full-width letters appeared in the precedent, they are replaced by a half-width uppercase capital letter "A" to represent the confidential word.

3.4 Experimental results of the improved algorithm

The experimental results after the improvement of the preprocessing algorithm are shown in Table 2. We see that CW_PPL score of the proposed model decreased from 40.7 to 5.1 compared to the previous model (bi-directional model).

We found that the CW_PPL score had improved 88% inaccuracy compared to the previous model. As a result, we confirmed that our proposed model (bi-directional

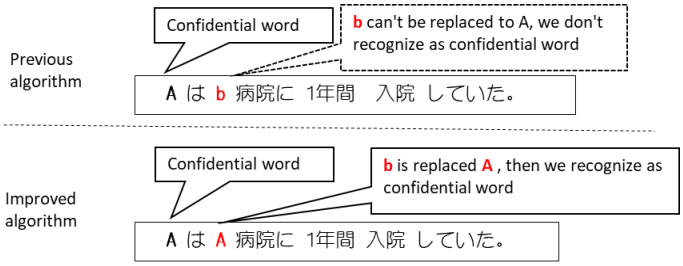


Fig.4. Example of the improved algorithm

Table 2. Result of experiment after the improved algorithm

	New proposed model after the improved algorithm	Previous model (Bi-directional LSTM-LR)
PPL	4.1	5.2
CW_PPL	5.1	40.7

LSTM-LR combined with conditional random fields (CRF)) had high accuracy for predicting confidential words [14].

Hence, the proposed model demonstrated excellent predictive ability, and next, we confirm whether the model was practical for applied use. However, the model has a problem if a person's name extracted from two words more and a long address number appeared, the CW_PPL score was worse. We explained the solution to solve them in Chapter 7.

4 Related works

To introduce related research, we describe general named entity recognition (NER) task explanations, their typical solutions, field-specific NER, and unsupervised NER. Named entity extraction is a widely used technique to obtain the target word in a sentence. NER is probably the first step for information extraction to locate and classify named entities in a text into predefined categories, such as the names of people, organizations, locations, expressions of times, quantities, monetary values, and percentages. NER is used in many fields in NLP [15] [16] [17]. Especially, automatic anonymization work in the healthcare domain using LSTM is actively studied for Japanese medical domains [18]. However it is necessary to make a large amount of electronic health records

NER extraction is roughly executed by two methods: one is rule-based by pattern matching and the other is statistics-based by machine learning. Pattern matching has a very high cost due to the need to generate patterns from a named entity dictionary or to manually update them. Various kinds of machine learning methods have been developed to learn patterns of named entities by preparing the corpus, including support vector machine [19], hidden Markov model, and CRF [20]. CRF has mostly been successful for NER. Nevertheless, the problem of machine learning is that the cost of manually generating a corpus is high [19].

Recently, a new NLP technology called Bidirectional Encoder Representations from Transformers (BERT) has been developed; it achieves up to 90% correct answers [21]. In natural language, using a neural network, it is common to input an embedded word string into a learner for solving a task. However, BERT adds further information to the embedded words input by pre-learning and inputs it to the learner to enable highly accurate learning. However, application to the actual corpus of Japanese precedents is difficult as a large amount of learning data needs to be prepared in advance. Thus, we aimed to easily predict confidential words in Japanese precedents.

5 Extention of the proposed model to increase confidential word detection accuracy

In an actual legal record, evaluating whether a confidential word is correctly recognized or falsely identified is essential. Therefore, we used the evaluation parameters of "recall," "precision," and "F1" to confirm the possibility of practical use. Defining the threshold value of CW_PPL, which determines whether the confidential word is truly recognized, is necessary.

5.1 Evaluation parameters

We then evaluated how our proposed model affected some types of anonymized precedents. In this work, we consider the measures of classifier performance in terms of precision and recall, a measure that is widely suggested as more appropriate to the classification of imbalanced data. We used the recall, precision, F1 parameters to examine accuracy.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{F1} = \frac{2Recall \times Precision}{Recall + Precision} \tag{4}$$

TP: a true positive (i.e., when the actual word and predicted word are positive)
TN: a true negative (i.e., when the actual word and the predicted word are negative)
FN: a false negative (i.e., when the actual word is positive, but the predicted word is negative)
FP: a false positive (i.e., when the actual word is negative, but the predicted word is positive)

Recall is the most important because if the anonymization system fails to detect confidential word, it will cause serious problems.

5.2 Determining the threshold value of CW_PPL

If the CW_PPL of a confidential word ("A") was lower than 60 (threshold), we defined it as a correct word (TP). Otherwise, it was ignored (FN). A threshold of 60 was selected as it is the optimal value judged from the Area under the Curve (AUC))-Receiver Operating Characteristic (ROC) curves (Fig. 8). In machine learning, performance measurement is an essential task. In classification problems, we adopt the AUC-ROC curve [22], a key evaluation metric for checking any classification model's performance. The ROC curve is a performance measurement for classification problems at various threshold settings. While ROC is a probability curve, and AUC represents the degree or measure of separability that indicates how much the model is capable of

distinguishing classes. The higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. The ROC curve is plotted with a true positive rate (TPR) against the false positive rate (FPR), where TPR is on the y-axis and FPR is on the x-axis, using the equations below.

$$\text{TPR (True Positive Rate)} = \frac{TP}{TP + FN} \qquad (5)$$

$$\text{FPR (False Positive rate)} = \frac{FP}{FP + TN} \qquad (6)$$

The curve is deemed better when it is located toward the upper left.
It is critical that TPR (=Recall) is large and FPR is small. Therefore, we choose the threshold to be 60 that was the closest point of the ideal curve.

6 Applying the proposed model to the plain precedents

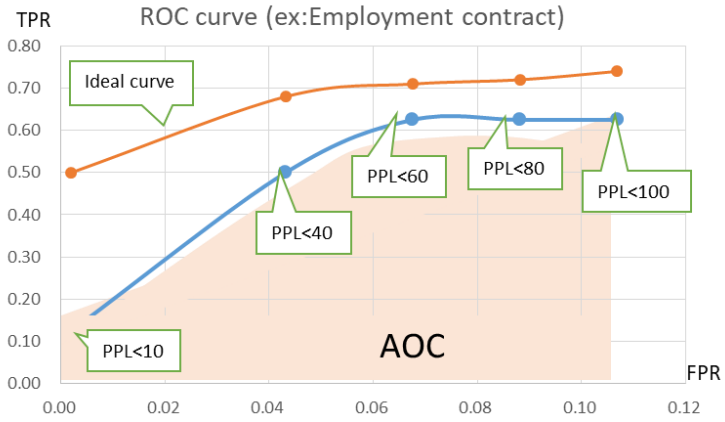


Fig. 5. ROC curve

6.1 Experimental results for anonymized precedents

The results of the experiment in various types of anonymized precedents are shown in Table 3. In the results, the recall rate was ≥80% in about 60% of the cases. However, the precision rate was low at approximately ≤30%. In particular, when the number of appearances of confidential words was small, the matching rate became low.In Land Contract and Moving trouble case, recall are very low because “attachment A” often appeared which didn’t mean confidential words and meant simple alphabet “A”.

First, however, confidential words need to be properly recognized as true confidential words. Therefore, if the recall rate is ≥80%, we consider that a commercialization possibility exists. Our final goal is to automatically replace the anonymous (confidential) words in the actual precedents with the capital letter “A” using our proposed model.

Table 3. Result of the experiment in various types of anonymized precedents

item	Total words	Confidential word			Normal word			Recall	Precision	F1
		Appear (TP+FN)	No hit (FN)	Hit (TP)	Appear (TN)	Hit (FP)	No hit			
Rental contact	7800	52	29	23	7748	623	7125	44%	4%	7%
Land contract	9600	1588	751	837	8012	806	7206	53%	51%	52%
Traffic accident	2600	67	10	57	2533	280	2253	**85%**	17%	28%
Traffic accident	8400	100	35	65	8300	831	7469	65%	7%	13%
Rental contact	4000	76	11	65	3924	250	3674	**86%**	21%	33%
Injury case	12800	543	169	374	12257	1624	10633	69%	19%	29%
Land contract	1800	34	26	8	1766	90	1676	24%	8%	12%
Investment receivables	17600	777	177	600	16823	1960	14863	**77%**	23%	36%
Employment contract	11600	152	31	121	11448	1045	10403	**80%**	10%	18%
Information disclosure	1600	30	7	23	1570	164	1406	**77%**	12%	21%
Stock claims	14200	529	82	447	13671	1439	12232	**84%**	24%	37%
Moving trouble	5200	79	58	21	5121	715	4406	27%	3%	5%
Building surrender	1600	4	0	4	1596	88	1508	**100%**	4%	8%
Facility admission fee	5800	2	0	2	5798	360	5438	**100%**	1%	1%
Contract receivables	3800	31	15	16	3769	328	3441	52%	5%	9%
Road maintenance guarantee	8600	34	8	26	8566	516	8050	**76%**	5%	9%

Therefore, we used the test data as the original precedent in which confidential words were not converted to "A." We experimented with this model to examine whether the model was practical.

6.2 An Experiment of the proposed model using non-anonymized precedents compared to anonymized precedents

We applied the proposed model to 100 cases of plain precedents to analyze the correctness of anonymization and compared the results of "before anonymization" with those "after anonymization by a legal expert," as shown in Table 4.

Considering this result, after anonymizing, a recall rate of ≥70% was found in 67% of the total, while before anonymizing it was 33%. Precision and F1 scores were better before anonymization than after anonymization.

7 The problem of the proposed model applying to the plain precedents (non-anonymized) and the solution for practical use

7.1 The problem of the proposed model using non-anonymized precedents

Our proposed model was unable to correctly predict the names of addresses as confidential words, especially as addresses in the precedents were often very long. This problem is one reason why the recall score was low. Another reason was that MeCab sometimes could not correctly extract a person's name when the full name was present. From these results, we can conclude that our model is not currently sufficient for prac-

tical use. However, when many confidential words appeared in the precedents, our proposed model was able to correctly predict them as the same phrases had appeared in the Japanese precedents.

Table 4. Application of model to precedents before and after anonymization

Item	Total words	Before anonymizing			After anonymizing		
		Recall	Precision	F1	Recall	Precision	F1
Request land surrender	3299	28%	9%	14%	32%	4%	7%
Rental contract	2200	37%	19%	25%	33%	6%	10%
Claim for damages	4600	**80%**	31%	45%	**82%**	27%	40%
Advisory contract	7200	58%	5%	10%	50%	1%	2%
Building rental contract	1246	**94%**	33%	48%	**100%**	29%	45%
Money lending	2534	58%	44%	50%	25%	22%	24%
Delinquent tax payments	7148	41%	15%	22%	29%	1%	3%
Discipline of lawyer	1044	0%	0%	#DIV/0!	**100%**	6%	12%
Internet contract	11280	**83%**	62%	71%	**87%**	36%	51%
Illegal residence (1)	14158	29%	6%	10%	57%	7%	13%
Illegal residence (2)	3701	58%	11%	18%	**74%**	10%	17%
Building pollution	4219	55%	9%	16%	25%	2%	3%
Internet sales	1021	**72%**	12%	21%	**92%**	24%	38%
Money consumption loan	1007	**76%**	35%	47%	**89%**	12%	21%
Aum Shinrikyo	10153	57%	10%	17%	**75%**	9%	16%

7.2 A solution for experimental results with the proposed model for practical use

We have shown an example to explain this in Fig. 6. In Fig. 6, Example 1 shows that the proposed model was able to predict the family name but not the first name because the name was extracted as two words. In contrast, Example 2 shows that it was correctly able to predict two names because each name was extracted as one word.

Confidential words were sometimes replaced with non-English letters, such as the Greek letters γ or β, in which case our proposed model was unable to predict them as confidential words (see Fig.7). Besides, addresses in the precedents appearing as "xx-xx-xx" were not predicted as confidential words.

We improved the preprocessing algorithm to solve these problems so that even if these letters appeared in the precedents, they can be properly recognized as confidential words. As shown in Table 5, when this improved algorithm was applied to the precedents that contained many addresses and Greek letters, the recall score improved by about 4%–49%, compared to that before the improvement. If we improve MeCab's ability to accurately extract the names of people, the recall score will increase.

However, MeCab was unable to extract first and family names as one word; hence, our model was sometimes unable to recognize names as the confidential word. Another issue was that it was unable to recognize the long numbered address as con-fidential words (see Fig. 8). If MeCab can extract the address as a proper noun, recognition

probability will improve. Furthermore, if MeCab can accurately extract names and proper nouns, our proposed model will prevent the false detection of confidential words.

This will increase the potential for practical use. These issues are summarized in Table 6.

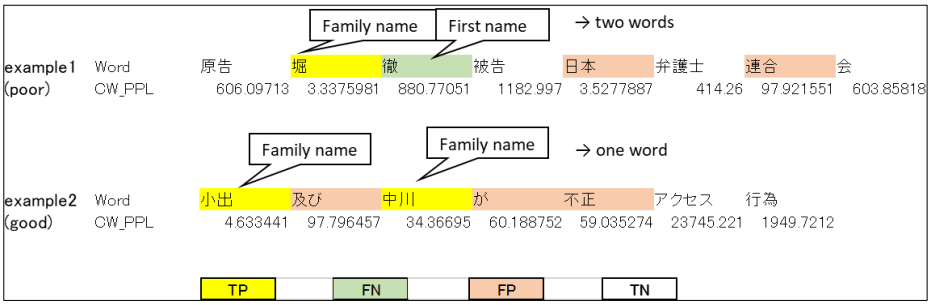


Fig.6. Example of predicting names of people

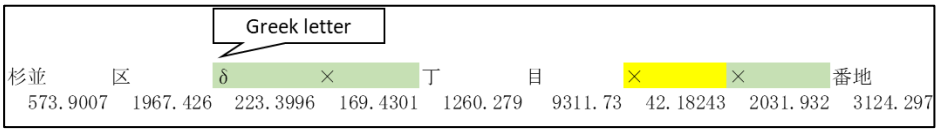


Fig.7. Example of predicting the Greek letter

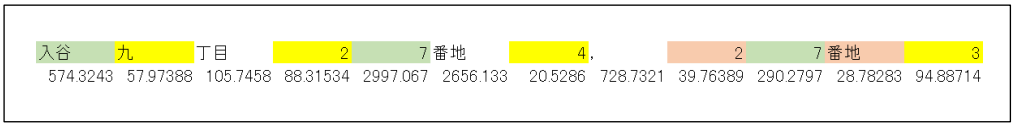


Fig.8. Example of predicting the long numbered address

Table 6. Issues of our model

	issues	measure
1	Long numbed address can't recognize correctly	Upgrade the "MeCab" and "preprocessing"
2	person's name sometimes can't recognize	Upgrade the "MeCab" and "preprocessing"
3	Precision and F1 score are low	Upgrade the "MeCab" & Further study

8 Conclusion

Adopting ICT in the judiciary system in Japan is necessary to meet complex problems associated with digital transformation. One important service is the opening of precedents and court records, for which the protection of privacy requires judgments to

be anonymized. To automate this work, which is currently manually performed, we studied an anonymization tool that uses neural networks. In particular, we found CBOW was effective for predicting target words and the bi-directional LSTM-LR model provided the best prediction result of confidential words.

In this study, we constructed a model wherein POS tags were added to the bi-directional LSTM-LR model, which is effective in NLP, and applied this to actual cases. Then, we examined the possibility of practical application. As a result, compared to the sole use of the bi-directional LSTM-LR model, we were able to improve the value of CW_PPL, which detects imperfections, by 88%. Furthermore, as a result of applying this model to actual cases, although the precision rate decreased, the recall rate was ≥70% in about 60% of the cases after anonymization. Issues encountered included the proposed model not predicting Greek letters and addresses, especially those containing a long number like "xx-xx-xx," as confidential words. By improving the preprocessing algorithm to recognize them as confidential words, the recall score was improved by 4%–49%. Although the precision was low and there were cases of falsely identified confidential words, the tool should help reduce the current manual labor during anonymization, providing a step toward disclosure of all precedents.

References

1. http://law.wm.edu/academics/intellectuallife/researchcenters/clct/ [Accessed on Sep 5, 2017]
2. Kasahara, T.: Cybercourt - Court Technology. Journal of the Japanese Society for Artificial Intelligence. **23 (4)**, 513 (2008).
3. Kasahara, T.: Court Technology for Civil Procedure. Festschrift for the seventieth birthday of Prof. Takeshi Kojima II. pp. 961 (2009).
4. http://www6.nhk.or.jp/special/detail/index.html?aid=20050213 [Accessed on Feb 2, 2005]
5. A study of applying ICT for legal service. Report of a research funded by the Ministry of Internal Affairs and Communications in Japan in 2009 (2010)
6. http://www5.cao.go.jp/keizai-shimon/kaigi/minutes/2017/0609/shiryo_07.pdf [Accessed on Sep 5, 2017]
7. https://www.yomiuri.co.jp/politics/20190810-OYT1T50080/
8. Siwei Lai, Kang Liu, Liheng Xu, Jun Zhao. How to generate a good word embedding. IEEE Intelligent Systems. **31.6**, 5–14 (2016).
9. Kiryu, Y., Ito, A., Kanazawa, M.: Acta Polytechnica Hungarica. Vol.16, 129-143 (2019). DOI:10.12700/APH.16.2.2019.2.8
10. http://www.courts.go.jp/app/hanrei_jp/search1 [Accessed on Sep 5, 2017]
11. TKC Corporation. http://www.tkc.jp/law/lawlibrary [Accessed on Sep 6, 2017]
12. Hagiwara, M., Ogawa, Y., Toyama, K: Automatic evaluation of synonym acquisition using perplexity. Proceedings of the Annual Meeting of the Association for Natural Language Processing. **12**, 767–770 (2006).
13. Mac Kim, S., Wan, S., Paris, C., Duenser, A.: Social Media Relevance Filtering Using Perplexity-Based Positive-Unlabelled Learning. ICWSM-2020 (8–11 June 2020).
14. Kanazawa, M., Ito, A., Yamasawa, K., Kasahara, T., Kiryu, Y., Toyama, F.: Method to Predict Confidential Words in Japanese Judicial Precedents Using Neural Networks With Part-of-Speech Tags. Infocommunications Journal. **XII (1)**, 17–21 (2020). DOI: 10.36244/ICJ.2020.1.4

15. Li, J., Sun, A., Han, J., Li, C.: A Survey on Deep Learning for Named Entity Recognition. cs.CL (2018).
16. Misawa, S., Taniguchi, M., Miura, Y., Ohkuma, T.: Character-based Bidirectional-LSTM CRF with words and characters for Japanese Named Entity Recognition Proceedings of the First Workshop on subword and Character Level Models in NLP. Fuji Xerox Co. Ltd. pp. 97102 (2017).
17. Ken Y, Koru I, Shoko W, Eiji, W : The 31st Annual Conference of the Japanese Society for Artificial Intelligence.2017. DOI: https://doi.org/10.11517/pjsai.JSAI2017.0_2J2OS16a4
18. Kohei K, Hiromasa H, Takashi O, Mizuki M, Yoshinobu K: De-identifying Free Text of Japanese Dummy Electronic Health Records. LOUHI.Oct.2018. Association for Computational Linguistics. pp. 65-70 DOI: 10.18653/v1/W18-5608
19. Yamada, H., Kudo, T., Matsumoto, Y.: Japanese Named Entity Extraction Using Support Vector Machine. Information Processing Society of Japan. **43 (1)** 44–53,.2001
20. Takase, M., et al.: https://www.ninjal.ac.jp/event/specialists/project-meeting/files/JCLWorkshop_no3_papers/JCLWorkshop_No3_23.pdf14p. 14
21. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. (2008). arXiv preprint arXiv:1810.04805
22. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
23. https://www.quora.com/Which of neural network or CRF approaches is efficient for NER? [June 15, 2017]

Compliance with Legal Requirements for the International Transfer of Personal Information

Tomoki Taniguchi

2-14-1 Tsuruyacho, Kanagawa-ku, Yokohama, Kanagawa, 221-0835 Japan
dgs205101@iisec.ac.jp

Abstract. The current business environment involves the cross-border management of personal information that is circulated globally, alongside a tendency for individual countries to establish their own data localization regulations for personal information through legal systems.

Therefore, with a focus on data localization regulations, consideration should be given to the regulations that are in place for the cross-border management of personal information by corporations that conduct business globally, and to the measures that should be taken in order to comply with such regulations. The core subject in this paper is the hypothetical transfer to Japan of personal information acquired in five countries by Japanese corporations that conduct business globally, with the five countries being, from among the many countries that have established data localization regulations in their legal systems, China and Vietnam, where the domestic storage of personal information is a legal obligation, the EU and Japan, where there are restrictions on the international transfer of personal information based on the viewpoint of protection of privacy, and the U.S., where there are no such obligations or regulations.

Keywords: Data localization regulations, International transfer restrictions, and Personal information protection legislation.

1 Data localization regulations

This chapter discusses three systems for establishing regulations regarding the processing of personal information in the target country when overseas corporations either directly or indirectly acquire personal information from individuals such as consumers and users located in the domicile country. The three systems are: (1) international transfer regulations, (2) data localization regulations, and (3) extraterritorial application. With some supplementary discussion of (2), this chapter will verify whether systems (1) to (3) exist in the five countries.

1.1 Systems to establish regulations for the processing of personal information acquired from individuals in the domicile country of overseas corporations either directly or indirectly

The three main systems are as follows.

International transfer restrictions

. Regulations that impose restrictions on transfers in the case that corporations in the domicile country directly acquire personal information in the domicile country from individuals located in the domicile country through the provision of products and services, and transfer such personal information to corporations located overseas.

Narrowly defined data localization regulations

. Regulations that impose the duty to establish equipment such as servers in the domicile country in order to either store personal information in the domicile country that was acquired from individuals located in the domicile country, or to process and store personal information, regardless of whether it is a domestic or international corporation.

There are narrow and wider definitions of these regulations, so minor additions are made to this definition below in "1.2"

Extraterritorial application [1]

. Systems that apply the legislative systems of the domicile country to overseas corporations when such corporations located overseas acquire personal information from individuals located in the domicile country.

Supplement to data localization regulations

Data localization regulations are systems that have the objective of retaining (localizing) data that is important to a domicile country inside that country for reasons such as the protection of privacy, the protection of industry in the domicile country, security guarantee, law enforcement and criminal investigations. There are narrow and wide definitions of these regulations.[2]

The narrow definition is a regulation that imposes the obligation for the domestic storage of data or the domestic installation of equipment. As above, these regulations impose the obligation for domestic storage of the data required for business management in the relevant country, or to install equipment such as servers for data processing and storage in the domicile country.

In the wider definition, the regulations include both and the obligation for domestic storage (or the obligation for domestic equipment installation), in addition to the above-mentioned international transfer restrictions.

1.2 Presence of systems in each of the five countries

Three of the five countries impose international transfer restrictions (China, the EU and Japan), while two of the five countries (China and Vietnam) impose the obligation for domestic storage or the obligation for domestic equipment installation as narrow

[1] The GDPR Article 3 Territorial scope.

[2] Yoshinori Abe.: Data localization measures and observance in international economic law: Legal positioning in WTO and TPP (Policy Research Institute, Ministry of Finance "Financial Review" No. 5, 2019), p26

data localization regulations. Only one country (China) imposes both international transfer restrictions and obligations for domestic storage, and only one country (the U.S.) imposes neither. Regarding extraterritorial application, there are differences in the presence of clear provisions and the scope of application, but such systems are present in each of the five countries.

Regarding international transfer restrictions, which is the focal point of this paper, by supplementing the objectives of these systems, it becomes clear that the objectives of each country may differ in this regard; the main purpose in China is to retain data domestically that is important to the country, while the main purpose in the EU and Japan is to protect the privacy of individuals in the domicile country.

2 Summary of legislation in each of the five countries

In this chapter, the five countries are separated into (1) China and Vietnam, where there are data localization regulations due to the obligation for domestic storage, (2) the EU and Japan, where there are data localization regulations that impose restrictions only on international transfer restrictions, and (3) the US, where there are no data localization regulations. Based on a summary of the legislation in each country along with the trends and topics for personal information protection, an outline of the regulations and the international transfer requirements will be presented.

2.1 Legislation in countries that establish data localization regulations incumbent on the obligation for domestic storage

China

Personal information protection legislation

. Currently, there is no comprehensive personal information protection legislation, and the regulations for the processing and management of personal information are dispersed between laws, administrative regulations, ministerial ordinances, local government regulations, judicial interpretations, and national standards.[3]

Among these, reference is made to the general concept of the Cyber Security Law ("CS Law"), which systematically coordinates the details including the definition of personal information and cross-border transfers, for the practical implementation of the processing and management of personal information. CS Law is the basic law for cyber security, as it establishes the regulations for network products and network services, and the regulations for personal information protection regarding cyber security. The objective of the CS Law is to promote the healthy development of an information-oriented economy and society by ensuring cyber security, maintaining control over cyberspace, national security, and public interests, and protecting the legal rights of citizens, corporations, and other organizations.

[3] There are movements to enact personal information protection laws.

Cyber security law and the scope of regulations

. Entities subject to the regulations of the CS Law are, among network providers, public transmissions and information services, energy, transport, water, finance and public services, electronic administration services, and other important industries and fields, as well as managers of important information infrastructures that could cause major damage to national security, the national economy and livelihood, or public interests in the case of a temporary inhibition, loss of function, or data leakage.[4]

The scope of information regulated in CS Law is personal information (various information that, on its own or in combination with other information, whether recorded electronically or in some other format, allows for the identification of the status of a natural person, which includes but is not limited to the natural person's name, date of birth, ID numbers, personal biometric information, address, and telephone number) and important data.

Data localization regulations

. CS Law imposes the obligation for domestic storage and international transfer restrictions on personal information and important data.

In the case that the consent of the individual is not obtained, CS Law prohibits the international transfer of personal information.

In addition, as a requirement of international transfer, there is a provision for a security assessment to be made regarding transferred personal information and important data, the details of which are defined in administrative measures (administrative laws).

This refers to the April 11, 2017, "Administrative measures for the security assessment of the international transfer of personal information/important data (call for suggestions),"[5] and the further supplementation to the administrative measures in the second draft published on August 30, 2017, "Data international transfer security assessment guidelines. However, in the case that personal information and important data poses a risk to the security of the national government and economy, etc., and in the case that it may harm the national security guarantee and public interest, such international transfers are prohibited. They are also prohibited in the case that other relevant authorities do not approve of the international transfer.

In this way, an extremely wide range of personal information/important data may be subject to restrictions on international transfer.

Notes (topic)

. In May 2020, the new Civil Code was adopted at the National People's Congress in China. The new Civil Code comprises 1260 articles, and it is composed of seven

[4] In the April 11, 2017, announcement, "Administrative measures for the security assessment of the international transfer of personal information/important data (call for suggestions)," the regulations were expanded to apply to "network managers."

[5] Subsequently, major changes were announced with regard to the provisions related to personal information in order to strengthen personal information protection in "Administrative measures for security assessments of the international transfer of personal information (call for suggestions)" in June 2019.

volumes and supplementary provisions, including a volume on personal rights. Chapter 6 of the volume on personal rights establishes the "protection of privacy and personal information," which reinforces the protection of privacy and personal information.

Vietnam

. *Personal information protection legislation*

. Currently, there is no comprehensive personal information protection legislation, and the regulations for the protection of personal information are provided for in individual laws, such as consumer protection laws, electronic transaction laws, information, and technology laws, as well as the constitution and civil code. In this way, laws have been enacted that make provisions in individual fields.

The "Cyber information security law" was enacted in July 2016, and the "Cyber security law" was enacted in January 2019. The former defines personal information and systematically regulates security management measures, etc., while the latter states the obligation for the domestic storage of personal information.

Scope of regulations in the cyber information security law

. The regulations of the cyber information security law cover activities for the processing of personal information in environments through electronic transmissions and computer networks for commercial purposes.

The parties covered by the regulations of this law are Vietnamese agencies/organizations and individuals as well as overseas organizations and individuals that process personal information. [6]

The regulations in this law are applicable to personal information and confidential information, etc. Personal information is defined only as "information related to the identification of specific individuals," and there are no sub-regulations that define the details of this.

Data localization regulations

. As stated above, there are no legal systems in Vietnam that restrict the international transfer of personal information.

However, the cyber information security law requires the acquisition of the consent of the individual for the processing of personal information. In addition, the obligation for domestic storage of personal information is regulated in the cyber security law.

The cyber security law places the duty for the domestic storage of data on cyberspace service business operators (domestic and international corporations that provide

[6] Defined as the one or multiple actions for the acquisition, editing, use, storage, provision, sharing or spread of personal information in cyberspace for business purposes (Article 3 (17) of the Law). Based on this definition, even if only a small amount of personal information is handled, the cyber information security law applies to parties that process personal information for business purposes.

services in Vietnam via telecommunications networks/the Internet and added-value services in other cyberspaces) for a period of time defined by the government.

Furthermore, this law obligates the establishment of a branch office or a local office in Vietnam if the business in question is overseas corporations.

Notes (topic)

. There have been reports that a draft ordinance for comprehensive personal information protection is scheduled to be published for public comment and enacted in 2020.

The draft law is a comprehensive law focusing on personal information protection, and while it includes concepts of the GDPR such as the establishment of managers and processors, sensitive data regulations, and international data transfer regulations, there are many details that have yet to be established.

2.2 Legislation in countries that establish data localization regulations that impose only international transfer regulations

The EU

. *Personal information protection legislation*

. The General Data Protection Regulation (the GDPR)[7] has been enacted as a comprehensive personal information protection law that applies to the European Economic Area (EEA) comprising the EU member nations + three countries (Iceland, Lichtenstein, and Norway).

Scope of regulations in the GDPR

. Entities subject to the regulations in the GDPR are corporations within the EU and overseas corporations that offer products and services to individuals in the EU, covering both managers (natural persons/corporations that solely or jointly determine the purpose and means of personal information processing) and processors (natural persons/corporations that process personal information on behalf of managers).The GDPR should apply to any information concerning an identified or identifiable natural person.

The GDPR should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable. This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes.

Data localization regulations

[7] Official title: “REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC.”

. The GDPR contains international transfer restrictions for personal data, as well as regulations for transfer requirements.

One of those requirements is an adequacy decision,[8] and a transfer can be made to the authorized country on that basis. Currently, there are 12 recognized countries and regions, including Japan and Canada.

In the case of transferring personal information to Japan based on an adequacy decision, the processing of such transferred personal information must comply with "Supplementary rules for the processing of personal data transferred from the EU and the UK based on an adequacy decision in line with laws related to the protection of personal information" ("Supplementary Rules") produced by the European Data Protection Board, as well as Japanese personal information protection laws.

Besides this, transfers can be made by means of the acquisition of consent from the individual, or by means of Standard Contractual Clauses (SCC) with the transfer destination (standard contracts specified by the EC), or approval from the jurisdictional agency for the Binding Corporate Rules (BCR) such as privacy policy applied to group companies, etc., and recognized authentication mechanisms such as the Privacy Shield with the US.

Notes (topic)

. Although adequacy decisions have not been implemented between the EU and the US, there is a substitute measure that secures a sufficient level of personal data protection on a registration basis for individual companies, which is called the Privacy Shield. However, in July 2020, the Court of Justice of the European Union (CJEU) passed a ruling to invalidate the Privacy Shield for the transfer of personal data from the EU to the US. In contrast, the use of SCC was validated.

This ruling was handed down in connection to a lawsuit contesting the legality of the transfer of personal information to the US from the EU-based corporation of Facebook, which is located in Ireland, by users of Facebook who are resident in the EU. Personal data transferred to the US from the EU may be covered if monitored by government agencies for the purpose of security guarantee, etc., based on the US domestic law. However, there are concerns about a lack of protection in the US even in the case that the transfer of personal data is based on the SCC or the Privacy Shield. In this way, the principal question related to the validity of both systems.

In the ruling, the CJEU confirmed the need for transfers of personal data to by covered by the GDPR and to offer the same level of protection as guaranteed by the GDPR based on appropriate protection measures, even in the case that personal data is monitored by government agencies based on the laws of a third-party country.

In this regard, the SCC were deemed to be valid because they include effective systems that guarantee the protection of personal data at a level required by the GDPR.

However, the same ruling deemed the Privacy Shield to be invalid, because the protection of personal data based on the US law was essentially recognized to be not at

[8] A system enacted with the requirement for taking sufficient personal data protection measures based on legal provisions, etc.

same level of protection as the EU law in cases where personal data is accessed by the US government agencies.

Japan

. *Personal information protection legislation*

. The Act on the Protection of Personal Information and its guidelines ("personal information protection law") provide comprehensive regulations for personal information protection.

Scope of regulations in the law

. The regulations cover Japanese businesses that handle personal information, and overseas corporations that handle the personal information of individuals resident in Japan through products and services.

The information covered by the regulations is information that can be used to identify a specific individual.

Data localization regulations

. The law establishes international transfer restrictions for personal information, and also defines transfer requirements.

The general requirement is for consent to be obtained from the individual to whom personal information pertains.

However, in any of the following cases, transfers can be made without the consent of the individual.

・Transfers to countries defined in the enforcement regulations as being a country that has a personal information protection system deemed to be at the same level as that of Japan (currently only the EU)

・In the case that the transfer destination maintains a system that conforms to the standards in the regulations as being a system that requires the continuous implementation of measures equivalent to those that should be taken by businesses processing personal information

・To ensure the measures according to the purpose of Article 4 (1) of the Act between transfer destinations for the appropriate and comprehensive methods of processing personal data

・When the transfer destination has received authorization based on international frameworks relating to the processing of personal information

・Cases based on the law, etc.

Notes (topic)

. The personal information protection law was amended in June of this year, and the amended law will be enforced within two years.

Based on this amendment, the rights of individuals and the penal regulations will be strengthened.

In this amendment, extraterritorial application will fall within the scope of decrees and the collection of reports secured by penal regulations covering overseas corporations that handle personal information related to individuals located in Japan.

2.3 Legislation in countries that have no data localization regulations

The US

. Personal information protection legislation

. There are no laws for the comprehensive protection of personal information; instead, there is a sectoral system with laws established for individual fields.

Based on such legislation, the Federal Trade Commission has authority to regulate fraudulent transactions in the Federal Trade Commission Act of 1914 that applies to all the US corporations, and it uses this authority for privacy protection.

In the case of an inconsistency between the details of privacy protection as published by a company and the actual processing of personal information by the company, the FTC classifies this as a fraudulent transaction. The information covered by this law is consumer data, and, while the FTC report contains no specific provisions in this regard, it does provide for "the protection of data that can be logically connected to specific consumers or devices such as computers as personal information."

Key examples of individual laws that define the processing and management of personal information include:

・Gramm-Leach-Bliley Act : 15 U.S.C. 6801-6809（GLBA）：
Regulations for the privacy and security of personal information by financial institutions.

・Children's Online Privacy Protection Act of 1998(COPPA)：
Regulations for the personal data of children aged under 13.

・Health Insurance Portability and Accountability Act of 1996(HIPPA)：
Regulations for the portability and accountability of health insurance.

・Health Information Technology for Economic and Clinical Health(HITEC)：
Regulations regarding health information technology for economic and clinical health.

Scope of regulations for individual laws in specific fields

. The parties and information covered are defined in the respective laws.

Data localization regulations

. There is no legislation in any law for international transfer restrictions or the obligation for domestic storage.

Notes (topic)

. There are movements to enact federal laws for comprehensive personal information protection.

Democrats introduced the "Consumer Online Privacy Rights Act" to the Senate.

Meanwhile, Republican lawmakers introduced the U.S. Consumer Data Privacy Act and further introduced the Consumer Data Privacy and Security Act to the Senate in March 2020.

（For more information on Privacy Shield, please see the EU "Notes (topic)".）

3 Points of similarity and difference in the regulatory content and transfer requirements in the five countries, and regulatory compliance and points of caution for international transfer

based on the details summarized in the previous chapter, the transfer of personal information to Japan is possible from all of the countries except for China.

China does not have clear transfer requirements, and international transfers are determined by national rulings, so there is a relatively high risk of regulatory violation in the application of transfers.

Meanwhile, the EU and Japan have systems for international transfer restrictions, but the regulations have the objective of privacy protection, and the transfer requirements are clear, so international transfers between the countries are possible if the requirements are met.

Vietnam and the U.S. do not have international transfer restriction systems, so international transfers to both countries are possible.

The below text presents a summary of points of caution for necessary regulatory compliance, and the appropriate response when transferring personal information to Japan from each of the countries.

3.1 China

Under current legislation, international transfers in practice are incredibly difficult, and highly cautious measures are required. This is because the international transfer of data personal information is currently determined by a national ruling.

There are two main reasons for this: (1) the requirement for international transfer is the acquisition of consent from the individual and a security assessment of the personal information, but the administrative laws (administrative measures) that regulate the details as well as the supplementary guidelines are still being drafted; and (2) data for international transfer is prohibited if it poses a risk to security (including national politics and the economy), if it may pose a risk to national security and public interest, or if the international transfer is not approved by a relevant authority.

Also, as stated above, due to the enactment of the new Civil Code, privacy and the protection of personal information has been strengthened, and even if such information

is retained domestically and not transferred internationally, more cautious, and appropriate processing is required than before.

3.2 Vietnam

As there are no international transfer restrictions, international transfers are possible with the consent of the individual when processing personal information.

However, for cyberspace service businesses, there is an obligation that personal information be stored domestically and that a branch office or resident office be established in Vietnam. For that reason, it is necessary to install a server in Vietnam or some other storage and processing equipment for personal information, and to consider providing structures for transferring data overseas from there.

3.3 The EU

There are clear transfer requirements, and thus transfers are possible if those requirements are met.

In particular, Japan is an adequacy decision target country, so there is no need to take special measures for transfers.

However, after a transfer, it is necessary to give attention to the processing of personal information in line with additional rules as well as Japanese personal information protection law.

For that reason, countermeasures should be taken for the management of personal information transferred from the EU separately from other personal information with different control methods.

3.4 The US

As there are no data localization regulations, transfers are possible.

However, as there is the possibility of extraterritorial application, it is necessary to conduct a prior investigation as to the specific regulations that will be applied in the case of extraterritorial application.

3.5 Supplement: Japan

As with the EU, the transfer requirements are clear, so if those requirements are met then transfers are possible.

4 Conclusion: Efforts to be made by corporations for the realization of Data Free Flow with Trust (DFFT)

4.1 Problem

In this study, the focus was on five countries, but there are other countries with legal systems that provide for data localization regulations and it will be necessary to conduct the same study for a wider range of countries in the future.

A considerable amount of time will be needed to complete the study for all the countries of the world. In addition, laws will be amended and regulations will be strengthened, and, depending on the country, the laws and regulations may vary for application in different fields. Therefore, as products and services are released in more countries, there will be an increase in the production processes, costs and personnel needed to deal with this, and there may be some difficulty in promoting measures in line with the speed of business development.

In such a situation in which there are many target countries and there are different regulation details and transfer requirements in each country, it is worth considering the kind of personal information protection initiatives that should be individually promoted in order for corporations that carry out the cross-border management of personal information to comply with regulations in a timely manner.

At the same time, it is necessary to take fail-safe and fool-proof measures in anticipation of doubts as to whether the relocation requirements are met.

4.2 Countermeasure

In my own opinion, from a practical perspective, the following three proposals must be considered.

The first proposal is each country of release for products and services must specify in advance in the planning stage (1) the objective and (2) the type of personal information to be acquired. By so doing, even in countries without comprehensive personal information protection laws, it will become easier to specify the corresponding laws that regulate personal information protection, also leading to the construction of "data mapping" in a way that allows for the tracking of the processing of personal information at all times. In this way, even if legislation is enacted or revised that regulates the protection of personal information, it will be possible to undertake legal and regulatory compliance in a timely manner.[9]

The second proposal is the acquisition of ISO/IEC27701: 2019 third party certification for privacy protection (private certification at present) from among the ISMS standards in countries that handle personal information, especially countries that are sources of international transfers. The subjects for the acquisition of this certification are not necessarily corporations; instead they may include products, services or bases (for

[9] Data localization regulations may apply depending on the purpose and type of data used (confidential or general information).

example, multiple offices, etc. of the data centers for a service located domestically and overseas), thus making it possible to construct privacy protection systems in line with international standards for relevant organizational subjects on a global scale where the need to do so is sufficient. In general, when the gap between the regulatory level of the international transfer source country and the actual management level is reduced, less time and fewer processes, expenses and personnel are needed for regulatory compliance processes, and there is a decreased risk of legal violation due to discrepancies. The construction of fixed-level privacy protection management systems will make that gap even smaller, so the acquisition of certification is an effective means in this regard.[10]

The third proposal is to establish a mechanism to ensure that "put personal information under certain control of the parties concerned as important national data" and "protect privacy", which is the main purpose of data localization regulation. If a company that handles personal information is skeptical that some of the transfer requirements are satisfied, it is possible to defend that it does not violate the purpose of the law. For example, in order to ensure " putting personal information under certain control of the parties concerned ", the company should encrypt personal information, store it on a server that can set a domain arbitrarily, and be put in place to store the data transfer logs for a certain period of time. In addition, in order to ensure that "protect privacy", the company should establish a mechanism for concealing personal information by a hash function or the like and automating the information processing. (Invasion of privacy occurs when personal information is viewed by a natural person other than the data subject, so in the process of automatic processing, make it a process that cannot be viewed by a natural person at all, such as by utilizing AI. is important.)

This paper concludes with the three proposals given above. From the perspective of compliance for corporations that carry out cross-border management of personal information acquired globally, corporations will be able to ensure the proper handling of personal information and privacy information by implementing the proposals.

End

References

1. Atsumi & Sakai.: Investigative study of recent trends related to personal information protection systems in various countries, pp. 9ff. (2018).
2. DLA Piper Homepage, https://www.dlapiperdataprotection.com/index.html?c2=VN&c=CN&t=transfer, Last accessed 2020/9/1.
3. EUR-Lex Homepage: REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), https://eur-lex.europa.eu/eli/reg/2016/679/oj, last accessed 2020/8/20.
4. Hiroyuki Tanaka.: Latest trends in global data protection legislation – the latest information on European GDPR/e-privacy regulations, California consumer privacy law, Asia/BRICs data protection law. Mori Hamada & Matsumoto, Japan (2019).

[10] ISO / IEC 27701: 2019 is associated with the GDPR. In addition, not a few countries have enacted a personal information protection law system with reference to the GDPR.

5. Hitomi Iwase.: Overview of system amendment and the 2020 personal information protection law amendments explained with a legal basis. Nishimura & Asahi, Japan (2020).
6. Hitomi Iwase, Ayako Matsumoto, Noriya Ishikawa, Yuko Kawai, Tomonobu Murata.: Updated laws for personal information protection/data protection regulations in various countries. Nishimura & Asahi, Japan (Personal information protection/data protection regulations newsletter no. 27, Feb. 2020).
7. Hitomi Iwase, Makiko Tsuda.: Amended personal information protection law; Question points from deliberations in National Diet. Nishimura & Asahi (Personal information protection/data protection regulations newsletter no. 26, June 2020).
8. Internet Initiative Japan Inc. "IIJ": China: Establishment of new Civil Code (June 3, 2020) https://blog.bizrisk.iij.jp/452, last accessed 2020/8/20.
9. I. Glenn Cohen JD, Michelle M. Mello, JD, PhD.: HIPAA and Protecting Health Information in the 21st Century.
10. Keisuke Yoshinuma.: Biz News "CJEU ruling invalidates "privacy shield" for personal data transfer with the U.S." (July 17, 2020). JETRO, https://www.jetro.go.jp/biznews/2020/07/d4dfd684421ffb4b.html, Last accessed: July 2020.
11. kousuke Kizawa, Kazuko Maruyama.: Initiatives toward the free flow of reliable personal data supporting the globalization of Trend Eye business activities. Business Law (2020).
12. Minh Thi Cao Koike.: Personal information protection legislation in cyberspace. Nagashima Ohno & Tsunematsu Asia Legal Review (2019).
13. Noriya Ishikawa.: Serial personal information protection law: Latest global trends No. 4 Vietnam – New government order to be enacted this year. Business Law (2020).
14. Peter Edemekong, Pavan Annamaraju (Article Editor: Micelle Haydel).: Health Insurance Portability and Accountability Act (HIPAA). In: Editor: Micelle aydel.
15. Shohei Suzuki, Komei Matsunaga.: Behavioral targeting advertising and Japan-U.S.-EU privacy protection regulations. Business Law Practice (2020).
16. Noriya Ishikawa.: Basic information about data privacy/compliance system construction. Business Law (2020)
17. Secretariat of Personal Information Protection Commission.: International personal data transfers (2018), https://www.kantei.go.jp/jp/singi/it2/senmon/dai14/ , Doc 2.2 in link, Last accessed: 2020/7/31.
18. Shinji Terada.: FY 2019 Tokyo personal information protection system briefing session; Summary of foreign legal systems for the protection of personal information (Sep.27 2019).
19. Mulder & M. Tudorica: Privacy policies cross border health data and the GDPR.
20. Yoshinori Abe.: Data localization measures and observance in international economic law: Legal positioning in WTO and TPP (Policy Research Institute, Ministry of Finance "Financial Review" No. 5, 2019).

Differential Translation for Japanese Partially Amended Statutory Sentences

Takahiro Yamakoshi[1], Takahiro Komamizu[1,2], Yasuhiro Ogawa[1,2], and Katsuhiko Toyama[1,2]

[1] Graduate School of Informatics, Nagoya University
[2] Information Technology Center, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
Email: yamakoshi@kl.i.is.nagoya-u.ac.jp

Abstract. We propose a *differential* translation method that targets statutory sentences partially modified by amendments. After a statute is amended, we need to promptly update translations of it for foreigners. We must focus on the *focality* of translation. In other words, we should modify only the amended expressions in the translation and retain the others to avoid causing misunderstanding of the amendment's contents. To generate focal, fluent, and adequate translations, our method incorporates neural machine translation (NMT) and template-aware statistical machine translation (SMT). In particular, our method generates n-best translations by an NMT model with Monte Carlo dropout and chooses the best one by comparing them with the SMT translation. This complements the weaknesses of each method: NMT translations are usually fluent but they often lack focality and adequacy, while template-aware SMT translations are rather focal and adequate but not fluent. In our experiments, we showed that our method outperformed both the NMT-only and SMT-only methods.

Keywords: Japanese Statutory Sentences, Differential Translation, Amendment

1 Introduction

In the world's globalized society, governments must quickly publicize its statutes worldwide to facilitate international trade, investments in their economy, legislation support, and so on. The Japanese government, to cope with this issue, launched the Japanese Law Translation Database System (JLT) [15] in April 2009 where it publicizes English translations of Japanese statutes. However, as of January 2020, only 23.4% (163/697) of the translated statutes in JLT correspond to their latest versions. After amending a statute, we need to promptly update its translation, or foreigners may be confused about whether it remains in effect. However, statutes are much tougher to be translated than ordinary documents because the former are highly technical, complex, and long.

Machine translation is a promising solution because such methods can produce translations much faster than human translators. Such neural machine translation (NMT) methods as Transformer [16] output very fluent sentences. The typical input to a machine translation model is one source language sentence, and the output is one target language sentence. However, this setting has a problem in producing amended statutory sentences that are partially modified.

From the perspective of a statute distributor who wants to precisely publicize statutes, making *focal* translations is more optimal for such sentences; that is, we should only

modify expressions that are changed by the amendment without modifying the others. For example, assume that the following statutory sentence must be revised: "The application shall contain a signature from the applicant's father." The revision needs to indicate that both parents must sign the document. The following revision satisfies the focality requirement: "The application shall contain signatures from both of the applicant's parents," which contains the minimum modifications. On the other hand, although "The application form shall include the signatures of both of the applicant's parents" is fluent and adequate, it is unsuitable for the revised sentence from the focality perspective because its sentence structure is drastically changed. Typical machine translation methods do not assure to output focal translations, since it takes only a source language sentence to be translated as the input data.

For such problems, it is preferable to use a machine translation method that incorporates a translation template: a pair comprised of a source sentence and its corresponding target sentence. In this methodology, we can make a translation of the post-amendment sentence by modifying the translation of the pre-amendment sentence in accordance with the difference between the input sentence (i.e., the post-amendment sentence) and the pre-amendment sentence. Koehn's statistical machine translation (SMT) method [7] incorporates a translation template, which identifies a satisfactory template from a translation memory (TM), and determines expressions to be translated based on edit distance and only translates those expressions. Kozakai's SMT method [8] adapted Koehn's method to partially modified Japanese statutory sentences. To identify expressions to be translated, it utilizes a comparative two-column table that indicates the differences between pre- and post-amendment statutes. Although these methods can generate focal translations, the translation quality is inadequate for three reasons. First, an SMT model's performance is typically worse than NMT's. Second, their methods completely lock unchanged expressions, which is a too strong restriction. Third, they use word alignment to find target language expressions that correspond to source language expressions, whose errors lead to incorrect translations.

From this background, we propose a new machine translation method for partially modified Japanese statutory sentences. Our method incorporates NMT and template-aware SMT to satisfy focality, fluency, and adequacy. We obtain n-best translations from an NMT model as candidates and choose the best one based on similarity with the translation from a template-aware SMT model. Since a template-aware SMT retains the unchanged expressions, we expect that the best candidate will be the most focal among all candidates and more fluent than the template-aware SMT translation. Here we apply Monte Carlo (MC) dropout [2] to the NMT model. NMT with MC dropout generates more diverse translations by making some neurons inactive during translation generation, increasing the chance to find a better candidate.

This paper contributes to amended statutory sentence translation tasks by reviewing the task requirements, proposing a method that meets the requirements, and describing its performance. Also, we expect that our method makes the whole statutory sentence translation process efficient because amendment acts occupy most of enacted statutes.

This paper is organized as follows. In Section 2, we position our task by introducing the amendment procedure in Japanese legislation. In Section 3, we explain related work.

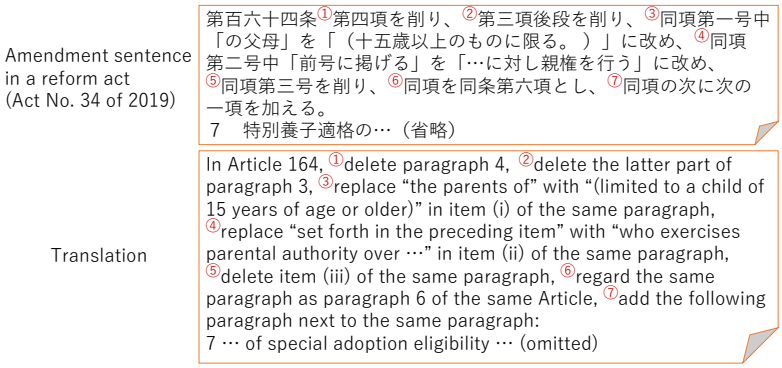


Fig. 1. Amendment sentence

In Section 4, we describe our method. Section 5 presents our evaluation experiments and discussions. Finally, we summarize and conclude in Section 6.

2 Task Definition

In this section, we clarify the background and the objective of our task. First, we introduce the amendment process in Japanese legislation from the viewpoint of document modification. Next we position the objective of our task in the amendment process.

2.1 Partial Amendment in Japanese Legislation

In Japanese legislation, partial amendment is done by “patching” modifications to the target statute, which are prescribed as amendment sentences in a reform statute. Such modifications can be categorized as follows [12]:

1. Modification on part of a sentence: (a) Replacement, (b) Addition, and (c) Deletion.
2. Modification on statute structure elements such as sections, articles, items, etc.: (a) Replacement, (b) Addition, and (c) Deletion.
3. Modification on element numbers: (a) Renumbering, (b) Attachment, and (c) Shift.
4. Combined modification of element renumbering and replacement of its title string.

In modifying part of a sentence, Japanese legislation rules [5] dictate that the expressions to be modified must be uniquely specified by chunks of meaning.

Figure 1 shows an example of an actual amendment sentence prescribed by a reform act. Any of the seven modifications in the amendment sentence can be assigned one of the categories described above: Modifications ① and ⑤ belong to 2. (c) of a paragraph and an item, respectively; Modification ② belongs to 1. (c); Modifications ③ and ④ belong to 1. (a); Modification ⑥ belongs to 3. (c); Modification ⑦ belongs to 2. (b).

Usually, a comparative table that corresponds to a reform statute is publicized, which shows the amendment contents by aligning pre-amendment and post-amendment statutes and underlining modifications. Figure 2 is a sample of a comparative table for the amendment sentence in Fig. 1.

Post-amendment statute	Pre-amendment statute
（省略） 6　家庭裁判所は、特別養子縁組の成立の審判をする場合には、次に掲げる者の陳述を聴かなければならない。② ⑥ ③ 一　養子となる者（十五歳以上のものに限る。） 二　養子となるべき者に対し親権を行う者（養子となるべき者の父母及び養子となるべき者の親権者に対し親権を行う者を除く。）及び養子となるべき者の未成年後見人　④ （削除）⑤ （削除）① 7　特別養子適格の…　⑦ （省略）	（省略） 3　家庭裁判所は、特別養子縁組の成立の審判をする場合には、次に掲げる者の陳述を聴かなければならない。この場合において、第一号に掲げる者の同意がないにもかかわらずその審判をするときは、その者の陳述の聴取は、審問の期日においてしなければならない。 一　養子となる者の父母 二　養子となるべき者に対し親権を行う者（前号に掲げる者を除く。）及び養子となるべき者の未成年後見人 三　養子となるべき者の父母に対し親権を行う者及び養子となるべき者の父母の後見人 4　家庭裁判所は、… （省略）

Fig. 2. Comparative table (circled numbers correspond to modifications described in Fig. 1)

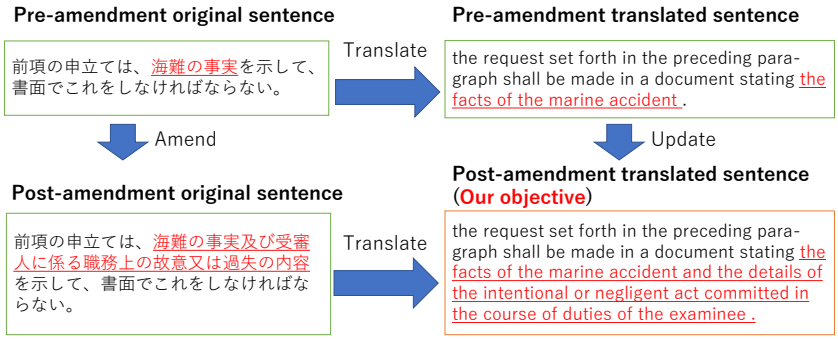


Fig. 3. Differential translation in an amended statutory sentence

2.2 Objective

Our task addresses post-amendment statutory sentences. Among the categories described in the previous section, we focus on category 1. (i.e., replacement, addition, and deletion of part of a sentence), and thus this task resembles a *differential translation task*. In the case of Fig. 1, modifications ③ and ④ are our targets. Here, we also target those that insert an additional sentence (e.g., a proviso) to an existing element or delete a sentence from the element, since additions or deletions affect the main sentence. For example, modification ② in Fig. 1, removing the latter part, is such a case.

Our task takes a triplet of sentences (*a pre-amendment original sentence*, *a post-amendment original sentence*, and *a pre-amendment translated sentence*) as input and generates a translation for the post-amendment original sentence called *a post-amendment translated sentence*. A pre-amendment original sentence is a statutory sentence in a statute before an amendment. A post-amendment original sentence is a statutory sentence in the statute after an amendment. A pre-amendment translated sentence is a translation of the pre-amendment original sentence. Figure 3 illustrates this task.

In generating post-amendment translated sentences, it is better to only modify expressions that are changed by the amendment without modifying the others. There are two reasons from the viewpoint of precise publicization. First, such sentences clearly

represent the amendment contents, which helps international readers to understand the contents. Second, the expressions in pre-amendment translated sentences are reliable so that reusing them ensures the translation quality. In the case of Fig. 3, we should replace "the facts of the marine accident" with "the facts of ... of the examinee" and keep other expressions which correspond to the following instruction: "Replace 「海難の事実」 (*kainan no jijitsu*) with 「海難の事実及び…過失の内容」(*kainan no jijitsu oyobi ... kashitsu no naiyo*)." We call this requirement *focality*, which is the third requirement along with fluency and adequacy that are the primary metrics of general machine translation.

We define our task as follows:

Input:
  Pre-amendment original sentence W_{bs};
  Post-amendment original sentence W_{as};
  Pre-amendment translated sentence W_{bt}.
Output: Generated post-amendment translated sentence $\hat{W}_{at}$.
Requirements:
  focality: $\hat{W}_{at}$ should be generated based on expressions of W_{bt}, and $\hat{W}_{at}$ should exactly reflect amendment from W_{bs} to W_{as};
  fluency: $\hat{W}_{at}$ should be natural in terms of phrasing and syntax;
  adequacy: $\hat{W}_{at}$ should have the contents in $\hat{W}_{as}$ without excesses or inadequacies.

3 Related Work

In this section, we review related work. We introduce template-aware SMT methods in Section 3.1 and NMT methods in Section 3.2.

3.1 Template-aware Statistical Machine Translation

Koehn et al. [7] proposed an SMT-based translation method incorporated with translation memory (TM). Figure 4 outlines this method, which generates translations by the following procedure:

1. Extract a source language sentence and its corresponding target language sentence from the translation memory. Hereinafter, we call the former sentence the TM source sentence and the latter the TM target sentence. We extract a pair whose TM source sentence resembles the input sentence.
2. Determine the objective expressions that must be retranslated. This process is based on the edit distance calculation, where we determine the minimal edits that transform a TM source sentence into an input sentence. Then we mark the edited words as objective expressions.
3. Find expressions in the TM target sentence that correspond to the objective expressions by word alignment.
4. Lock the expressions in the TM target sentence that do not correspond to the objective expressions.
5. Finally, translate the objective expressions by an SMT model and concatenate them with the locked expressions in the previous procedure.

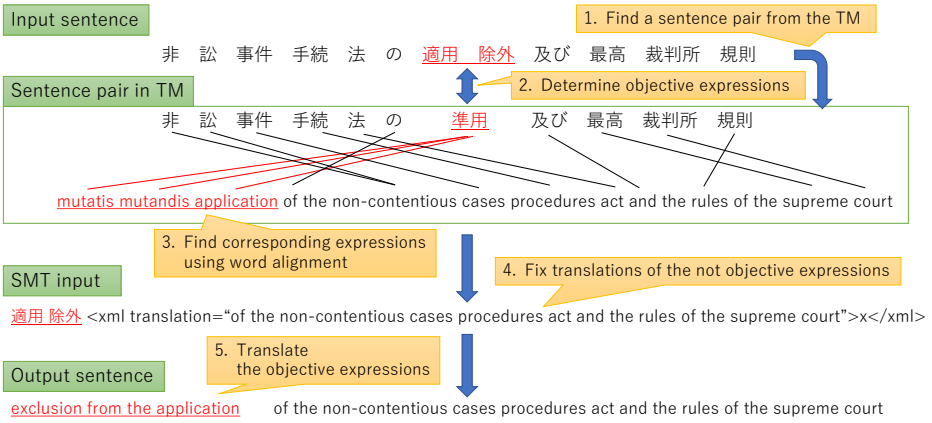


Fig. 4. Procedure of Koehn's method

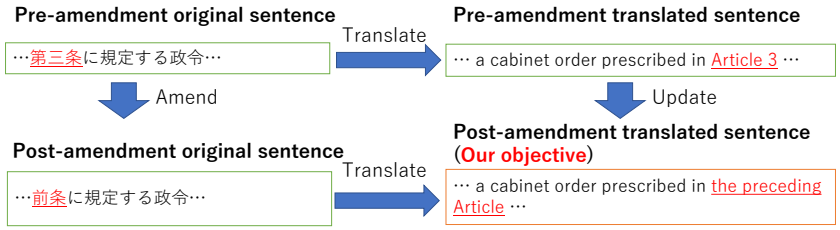


Fig. 5. Example that shows how Kozakai's method copes better

This method utilized Moses [6] for the SMT model. Moses can lock translations by annotating them with XML tags (Fig. 4).

Kozakai et al. [8] adapted this method to our task by applying the following two modifications. First, they used a pre-amendment original sentence and its translation instead of a relevant pair from the translation memory. Second, to determine the objective expressions, they used underlined information in the comparative table instead of the edit distance. The underlined information is more reasonable as a translation unit than the edit distance since sentence modification is done by a chunk of meaning in Japanese legislation. A critical example is shown in Fig. 5. We have underlined information between "第三条 (*dai san jo*)" and "前条 (*zen jo*)." Both objective expressions denote specific articles. By applying this underlined information, we get the following adequate translation: "… a cabinet order prescribed in the preceding article …" On the other hand, if we use edit distance for this example, we get an edit that replaces "第三 (*dai san*; the third)" with "前 (*zen*; preceding)," since "条 (*jo*; article)" is common in these sentences. Applying Koehn's method, we finally get the following inadequate translation: "… a cabinet order prescribed in Article preceding …"

Both methods can meet our focality requirement by locking the unchanged expressions in the pre-amendment translated sentences. However, the translation quality, especially fluency, suffers for the following three reasons. First, they use SMT for the

translation model, whose performance is typically worse than that of NMT. Second, their methods completely lock the unchanged expressions, which may strongly restrict translations. Third, they use word alignment to find English expressions that correspond to Japanese expressions, which may weaken their performance due to alignment error.

3.2 Neural Machine Translation

Neural machine translation (NMT) is the most common machine translation scheme because of its fluent output. A typical NMT model embeds an input sentence into a vector or multiple vectors and outputs a translated sentence by referencing the vector(s). One of the most powerful and influential NMT architectures is Transformer [16], which is comprised of two attention mechanisms: self-attention and cross-attention. These attentions capture such contexts as dependency and adjacent word information [17].

Some studies have focused on the incorporation of NMT and TM. For example, Cao et al. [1] proposed an NMT architecture that encodes an input sentence and its relevant TM target sentence into each vector and utilizes them for predicting target words by balancing two vectors based on context. Xia et al. [18] proposed a Transformer-based architecture that accepts TM target sentences by encoding them with a graph network.

However, a naive NMT model does not guarantee that unchanged expressions will be retained. The TM-incorporated NMT methods described above also fail to provide such guarantees because the expressions in the TM sentences are embedded and outputting them is influenced by probability.

4 Proposed Method

In this section, we describe our proposed method. Figure 6 shows its procedure. Our method uses an NMT model and a template-aware SMT model. We let the NMT model output n-best translations, which is done by applying a beam search. We then choose the most similar translation to the *interim* reference translation that is generated from the template-aware SMT model. With this methodology, we expect that these two methods compensate for the disadvantage of each method:

- An NMT model can output natural sentences, but it may modify or drop expressions that should not be tampered with.
- Although a template-aware SMT model can retain unchanged expressions during the translation process, its output may not be fluent.

We assume that the best candidate from the NMT model should cover more unchanged expressions than the other candidates, and that the best candidate is more fluent than the template-aware SMT translation.

We want the NMT model to output more diversified sentences to identify a better candidate. Therefore, we apply Monte Carlo (MC) dropout [3] to the model for generating NMT translations. By removing neurons from the NMT model, it can output more diversified translations based on its uncertainty about the input [2]. Therefore, we can gather more various kinds of translations and perhaps locate a better candidate nearer

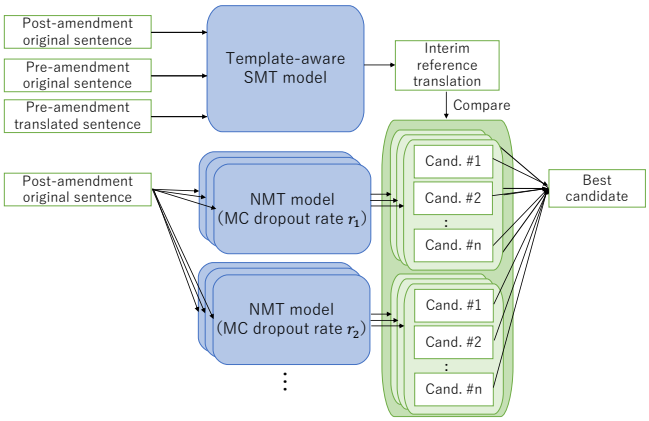


Fig. 6. Procedure of our method

the reference translation. We merge translations from different dropout rates, including dropout rate $r = 0.0$, which means that no MC dropout was applied.

In sentence comparisons, we can substantially utilize any criterion that assesses the similarity between two sentences. However, we believe that BLEU [13] is a good criterion because its calculation is based on an n-gram concordance rate, which does not heavily penalize drastic word order changes compared to word matching-based metrics like RIBES [4]. We should not penalize the NMT translations according to translations of template-aware SMT models that contain word order errors.

We design an n-gram base criterion for focality. As we discussed in Section 2.2, focal translations should contain expressions in the pre-amendment translation that are unaffected by the amendment. We quantize this idea by calculating the recall of the n-grams shared by both the pre-amendment and post-amendment translations. With pre-amendment translated sentence W_{bt} and true post-amendment translated sentence W_{at}, we calculate focality score $f(\hat{W}_{at}; W_{bt}, W_{at})$ of generated sentence $\hat{W}_{at}$ as follows:

$$f(\hat{W}_{at}; W_{bt}, W_{at}) = \mathrm{RP}(W_{at}, \hat{W}_{at}) \cdot \frac{\sum_{s \in \mathrm{CN}(\{\hat{W}_{at}, W_{bt}, W_{at}\})} \min(c_{\hat{W}_{at}}(s), c_{W_{bt}}(s), c_{W_{at}}(s))}{\sum_{s \in \mathrm{CN}(\{W_{bt}, W_{at}\})} \min(c_{W_{bt}}(s), c_{W_{at}}(s))}, \quad (1)$$

where $\mathrm{RP}(W_{at}, \hat{W}_{at}) = \min(1, \exp(1 - |\hat{W}_{at}|/|W_{at}|))$ avoids overestimating the scores of the redundant sentences and $c_W(s)$ is the number of occurrences of the n-gram s in W. $\mathrm{CN}(\mathcal{W})$, where $\mathcal{W} = \{W_1, W_2, \cdots, W_m\}$, returns common n-grams of $W_1, W_2, \cdots, W_m$:

$$\mathrm{CN}(\mathcal{W}) = \left\{ s \;\middle|\; s \in \bigcap_{W_i \in \mathcal{W}} \mathrm{ngrams}(W_i) \right\}, \quad (2)$$

where $\mathrm{ngrams}(W)$ returns n-grams in W. We consider the multiple lengths of n-gram:

$$\mathrm{ngrams}(W) = \bigcup_{i=1}^{N} i\text{-gram}(W), \quad (3)$$

where i-gram(W) returns the i-grams of W.

5 Experiment

Next we experimentally evaluated the effectiveness of our method.

5.1 Outline

For training data, we used a bilingual-statutory-sentence corpus compiled by Kozakai et al. [8] from JLT[3]. This corpus consists of 158,928 sentence pairs from 407 statutes. As test data, we used 158 sets of sentence amendment examples that consist of pre-amendment original sentences, post-amendment original sentences, pre-amendment translated sentences, and post-amendment translated sentences, where the pre- and post-amendment original sentences have underlined information. We acquired these sentence amendment examples from 17 amendments.

We used Transformer [16] for the NMT model and Kozakai's SMT model (hereinafter the Kozakai model) for the template-aware SMT model. Here are the settings of the NMT model: six encoder/decoder hidden layers, eight self-attention heads, 512 hidden vectors, a batch size of eight, and an input sequence length of 256. We used SentencePiece [9] as a tokenizer and set the vocabulary size to 32,768. We chose a dropout rate of 0.1 for training, which is the default setting of the official Transformer implementation. In the prediction phase, we executed the model with two different dropout rates, 0.0 and 0.1, where a 0.0 dropout rate equals a situation where no dropout was applied. We set the number of beam search candidates to four and the number of MC dropout trials to 20 and acquired $20 \times 4 = 80$ candidates for each dropout rate. The following are the settings of the template-aware SMT: GIZA++ [11] for the word alignment, SRILM [14] for the language model generation, and Moses [6] for the decoder. We used MeCab [10] for the Japanese tokenizer.

We used BLEU and RIBES to evaluate the fluency and adequacy. For focality, we use Eq. 1 with maximum n-gram length $N = 4$, which is the same as the standard BLEU score calculation. To cope with edge words, we padded the sentences when we acquired n-grams of them. We tested the following translation models as comparison methods: naive Transformer, naive Moses [6], naive Kozakai model, naive Koehn's model [7] (Koehn model), a combination of the Transformer and Koehn models, and a combination of the Transformer and Kozakai model without MC dropout.

5.2 Results

Table 1 shows the experimental results. To highlight the ideal performance of our method, we assessed its performance when we used the actual post-amendment translated sentences as the interim reference translations instead of the output of the Kozakai model. Our method achieved the best performance in both BLEU and RIBES. It also got 4.16 more focality points than naive Transformer, although the template-aware SMT methods achieved a much better performance in that criterion. This result was expected because these methods can completely lock unchanged expressions. However, there is a room for improvement in interim reference translations, because gaps remain between

[3] http://www.japaneselawtranslation.go.jp/

Table 1. Experimental results

Model	BLEU	RIBES	Focality
Transformer + Kozakai model [proposed]	**62.43**	**88.19**	84.27
Transformer + Koehn model	60.68	87.58	82.42
Transformer + Kozakai model (w/o MC dropout)	60.52	87.50	82.35
Naive Transformer	58.72	86.86	80.11
Naive Kozakai model	59.86	83.04	**85.60**
Naive Koehn model	59.22	82.52	85.21
Naive Moses	40.67	64.31	54.95
Transformer + true post-amendment sentence	69.13	89.95	87.24

Table 2. Successful case (Trm denotes Transformer)

Model	Output
(Sentence W_{bs})	（講習の業務の休廃止）
(Sentence W_{bt})	(suspension or abolition of training services)
(Sentence W_{as})	（エネルギー管理講習の業務の休廃止）
(Sentence W_{at})	(suspension or abolition of energy management training services)
Trm + Kozakai	(suspension or abolition of energy management training services)
Naive Trm	(suspension or abolition of energy management training)
Naive Kozakai	(suspension or abolition of energy management of training services)
Naive Moses	(suspension or abolition of training services energy management)

our method and the ideal performance: 6.70, 1.76, and 2.97 points in BLEU, RIBES, and Focality, respectively.

Table 2 shows a successful case. The naive Transformer model output "suspension or abolition of energy management training," where "services" was missing. On the other hand, our method chose the correct candidate by referencing the output of the template-aware SMT model: the naive Kozakai model. Table 3 shows an unsatisfactory case. Since the original sentence dropped the subject, "the permitted manufacturer," the Transformer model needed to estimate it. Our method can exploit this advantage based on how it chooses candidates. However, "the permitted manufacturer" subject did not appear among the candidates, so our method failed to output an adequate choice.

5.3 Discussion

First, we look at the generation ability of MC dropout. Figure 7 shows the outputs of our method with three MC dropout trials. In this case, the sentence structures of the outputs in dropout #3 were different from dropouts #1 and #2. On the other hand, the outputs in each dropout are rather similar. For example, they have minor word or phrase usage differences: "trial vs. hearing" and "conducted vs. carried out" in dropout #1 and "chief trial examiners vs. presiding judge" in dropout #2. One exception is expression "in case of. . . ," which was omitted in two outputs of dropout #3. This implies that MC dropout contributed to the generation of diversified translations and avoided bad translations.

Next, we assessed the quantitive performance of the MC dropout sentences. First, we focus on the relationship between the number of MC dropout trials and the BLEU

Table 3. Failure example

Model	Output
(Sentence W_{bs})	第十八条第一項の規定による命令に違反したとき。
(Sentence W_{bt})	where the permitted manufacturer has violated an order under the provisions of Paragraph (1) of Article 18.
(Sentence W_{as})	第十八条の規定による命令に違反したとき。
(Sentence W_{at})	where the permitted manufacturer has violated an order under the provisions of Article 18.
Trm + Kozakai	the person has violated an order under the provisions of Article 18.
Naive Trm	the designated calibration organization has violated an order under the provision of Article 18.
Naive Kozakai	where the permitted manufacturer has violated an order under the provisions of Article 18 of
Naive Moses	when he/she violated an order pursuant to the provisions of Article 18.

Post-amendment original sentence
合議体で審判を行う場合においては、審判官のうち一人を審判長とする。
Post-amendment translated sentence
one (1) judge shall be a presiding judge out of the three judges , if an inquiry is conducted by a panel .
4-best outputs in dropout #1
in cases where a trial is conducted by a panel , one of the trial examiners shall be the chief trial examiners .
in cases where a hearing is carried out by a panel , the chief hearing examiner shall be one of the hearing examiners .
in cases where a hearing is conducted by a panel , one of the hearing examiners shall be the chief hearing examiners .
in cases where a hearing is carried out by a panel , the chief hearing examiner shall have one of the hearing examiners .
4-best outputs in dropout #2
in cases where a trial is conducted by a panel , one of the trial examiners shall be the chief trial examiners .
in cases where a panel is conducted by a panel , one of the trial examiners shall be the chief trial examiners .
in cases where a trial is conducted by a panel , one of the trial examiners shall act as the presiding judge .
in cases where a trial is conducted by a panel , one of the trial examiners shall act as the chief trial examiner .
4-best outputs in dropout #3
the chief trial examiner shall , in cases of conducting a trial by a panel , be one of the trial examiners .
the chief hearing examiner shall , in cases where a hearing is carried out by a panel , be one of the hearing examiners .
the chief hearing examiner is to conduct a trial by a panel .
the chief hearing examiner is to conduct a trial by a panel ; and

Fig. 7. Example of MC dropout sentences

performance (Fig. 8). As we increased the MC dropout trials, the BLEU performance also generally rose. However, the growth rate was higher when we used actual post-amendment translated sentences as the interim reference translations. This result may reflect the limitation of the Kozakai model performance.

Second, we focused on the relationship between the number of MC dropout trials and the number of selections of non-MC dropout sentences. This investigation addresses whether doing more MC dropouts raises the chances of identifying a better translation. Figure 9 shows such relationship in our method. As we increased the MC dropout trials, the number of selections of non-MC dropout sentences decreased. However, the number eventually became saturated. This result suggests that more MC dropout trials generally lead to better translations, although this claim has limitations.

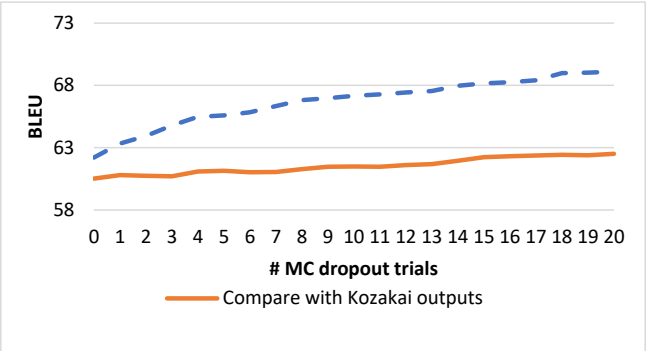


Fig. 8. BLEU transition on number of MC dropout trials

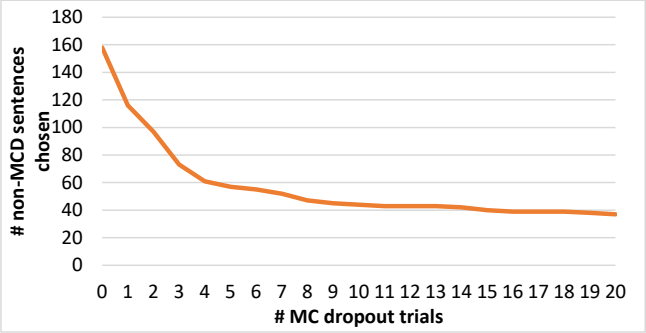


Fig. 9. Numbers of MC dropout trials and selections of non-MC dropout sentences

Finally, we discuss the effectiveness of the dropout rates. Figure 10 shows the sentence-level correlation coefficients between the metric scores and the sentence features. The metric scores include the BLEU, RIBES, and the focality scores of each sentence at different MC Dropout rates: {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}. As for dropout rates of 0.2 and larger, we trained a model at the new dropout rate. Sentence features include the total word count, the unique word count, the 4-gram language model probability, the unedited word ratio, and the unlined word ratio of each sentence. We found some interesting results. First, adjacent dropout rates have relatively high correlations in metric scores, especially in focality. This suggests that we can get translations with similar scores using similar dropout rates. Second, weak negative correlations exist between metric scores in strong dropout rates and some sentence features, including total word count, unique word count, unedited word ratio, and unlined word ratio. The result of the total and unique word counts suggests that Transformer models with a higher

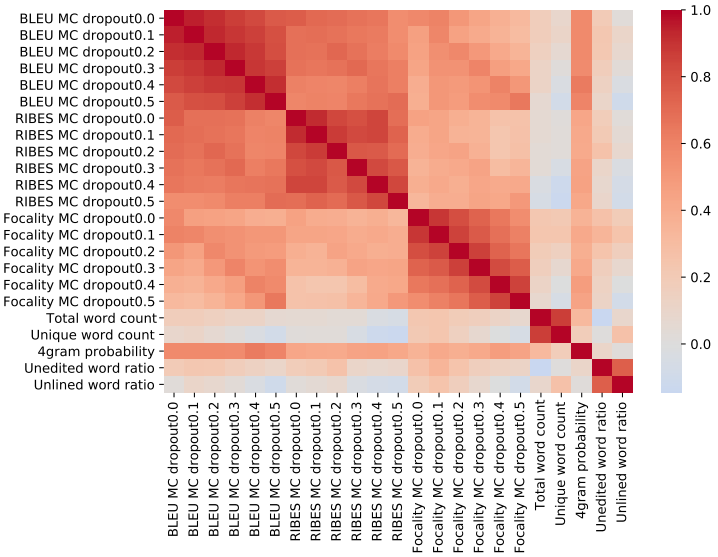


Fig. 10. Correlation between metric scores and sentence features

dropout rate tend to work more poorly against longer sentences. Concerning the result of the unedited or unlined word ratio, our hypothesis models with a higher dropout rate produced more diverse translations, and thus we needed more trials to get better sentences that more closely match the unchanged expressions.

6 Summary

We proposed a differential translation method for amended statutory sentences. Our method incorporates an NMT model and a template-aware SMT model, where the former outputs natural translation candidates and the latter outputs an interim reference translation that retains the contents of the pre-amendment statutory sentences. We augmented candidates with high diversity by applying Monte Carlo dropout to generate NMT translations. Our model outperformed NMT or template-aware SMT alone.

Our future work will address the following two tasks. First, we need to investigate the validity of the focality metric. Second, we must find a better solution to generate interim reference translations instead of a template-aware SMT model, since our NMT model has the potential to output better translations.

Acknowledgments This work was partly supported by JSPS KAKENHI Grant Number 18H03492.

References

1. Cao, Q., Xiong, D.: Encoding gated translation memory into neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3042–3047 (2018)

2. Fomincheva, M., Specia, L., Guzm'an, F.: Multi-hypothesis machine translation evaluation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1218–1232 (2020)
3. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning. 10 pages (2016)
4. Hirao, T., Isozaki, H., Sudoh, K., Duh, K., Tsukada, H., Nagata, M.: Evaluating translation quality with word order correlations. Journal of Natural Language Processing 21(3), 421–444 (2014), (In Japanese)
5. Hoseishitumu-Kenkyukai: Workbook Hoseishitumu (newly revised second edition). Gyosei (2018), In Japanese
6. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the ACL 2007 Demo and Poster Sessions. pp. 177–180 (2007)
7. Koehn, P., Senellart, J.: Convergence of translation memory and statistical machine translation. In: AMTA Workshop on MT Research and the Translation Industry. pp. 21–31 (2010)
8. Kozakai, T., Ogawa, Y., Ohno, T., Nakamura, M., Toyama, K.: Shinkyutaishohyo no riyo niyoru horei no eiyaku shusei. In: Proceedings of NLP2017. 4 pages (2017), (In Japanese)
9. Kudo, T., Richardson, J.: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations). pp. 66–71 (2018)
10. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. pp. 230–237 (2004)
11. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2005)
12. Ogawa, Y., Inagaki, S., Toyama, K.: Automatic consolidation of Japanese statutes based on formalization of amendment sentences. New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Revised Selected Papers, Lecture Notes in Computer Science 4914, 363–376 (2008)
13. Papineni, K., Roukos, S., Ward, T., Jing Zhu, W.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
14. Stolcke, A.: SRILM — an extensible language modeling toolkit. In: Proceedings of ICSLP 2002. pp. 901–904 (2002)
15. Toyama, K., Saito, D., Sekine, Y., Ogawa, Y., Kakuta, T., Kimura, T., Matsuura, Y.: Design and Development of Japanese Law Translation System. In: Law via the Internet 2011. 12 pages (2011)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems 30. pp. 6000–6010 (2017)
17. Voita, E., Talbot, D., Moiseev, F., Sennrich, R., Titov, I.: Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 5797–5808 (2019)
18. Xia, M., Huang, G., Liu, L., Shi, S.: Graph based translation memory for neural machine translation. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence. pp. 7297–7304 (2019)

Reasoning with Inconsistent Legal Ontologies based on Argumentation Theory*

Yiwei Lu[2] and Zhe Yu[1][0000−0002−8763−4473]

[1] Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University, Guangzhou 510275, P. R. China
yuzh28@mail.sysu.edu.cn
[2] Center for the Study of Language and Congnition, Department of Philosophy, Zhejiang University, Hangzhou 310028, P. R. China

Abstract. Ontology is a popular method for representing knowledge in the legal domain, while description logics (DL) is commonly used as its description language. To handle reasoning with inconsistent legal ontologies, this paper translates the DL ontology into the formulas and rules of a structured argumentation theory based on $ASPIC^+$, and proposes a combined theory called *DL-AT*. We show that using the *DL-AT*, acceptable assertions can be obtained based on inconsistent ontologies, and the traditional reasoning tasks of DL can also be accomplished. In addition, a definition of explanations for the result of reasoning is presented.

Keywords: Argumentation theory · Legal ontology · Description logics · Explainable AI.

1 Introduction

Ontologies are popularly applied in knowledge representation, structuring and management. They can also be helpful in extracting and modeling information in the legal domain, as well as realizing the reasoning problem-solving based on legal ontologies.

In the literature, many legal ontologies have been proposed, e.g. the Legal Knowledge Interchange Format (LKIF) core ontology builds on the Description Logics (DL) and LKIF rules [11, 12]; the Core Legal Ontology (CLO) based on the extension of the DOLCE (DOLCE+) foundational ontology [7]; the LRI-Core ontology aimed at the legal domain grounded in common-sense [4], and the Functional Ontology for Law (FOLaw) [17, 18].

As pointed out by van Engers et al. [19], law is based upon a dialectical process and reflecting preferences. It is expressed in natural language, which introduces ambiguity, interpretation problems and inconsistencies. However, the

* This work is supported by the MOE Project of Key Research Institute of Humanities and Social Sciences in Universities [No. 18JJD720005], the Philosophy and Social Science Youth Projects of Guangdong Province [No. GD19CZX03] and the China Postdoctoral Science Foundation [No. 2019M663353].

main description language of the current ontology is the Web Ontology Language (OWL) or OWL2, whose semantics are based on DL [1]. It is difficult to perform reasoning based on DL under inconsistency and provide explanations about how to reach the query result and why the result was accepted. In view of these considerations, argumentation is promising to provide an easy-to-explain method that can handle reasoning with inconsistent legal ontologies.

Since the seminal paper of Dung [6], formal argumentation has been noticed as an approach to deal with reasoning under inconsistent context [3, 8, 14]. Formal argumentation has the merits of computational efficiency and explainability. For handling reasoning with inconsistent ontologies, several existing research has considered to apply argumentation systems in this field, e.g. paper [8] expresses DL ontologies as Defeasible Logic Programs ($DeLP$), while paper [20] presents an argumentation framework based on the Assumption Based Argumentation (ABA) framework for dealing with inconsistent DL ontologies.

Various kinds of conflicts can be observed in legal texts, $DeLP$ system and ABA framework can each models one type of conflicts, i.e. conflicts in the conclusions of arguments and conflicts in the assumptions. Compared with other applicable argumentation systems, $ASPIC^+$ framework [14] can (1) support modeling three different types of conflicts between arguments, and hence can more flexibly express inconsistency; meanwhile, (2) support both skeptical and credulous justification, which can reflect users' different attitudes. Moreover, $ASPIC^+$ offers explicit principles for obtaining preferences over arguments.

For the above reasons, we argue that $ASPIC^+$ framework is a better choice for expressing legal ontologies with inconsistency and performing reasoning with such ontologies. In order to deal with reasoning based on inconsistent legal ontologies, according to $ASPIC^+$ and DL, we introduce a combined theory called *DL-AT*[3]. Based on the *DL-AT*, inconsistencies in legal ontologies can be resolved based on preferences and argument evaluation. Furthermore, we provide a novel definition of explanation that can reflect why an assertion is justified based on *DL-AT*.

The rest of this paper is organized as follows. Section 2 first briefly introduces the $ASPIC^+$ frameworks, then translates the DL ontology into an argumentation theory. Section 3 presents the *DL-AT* and illustrates our ideas with an example taken from a legal case, then shows how to perform inferences based on *DL-AT*. In the last section, we conclude the paper and discuss the future work.

2 Argumentation Framework and A Translation of Description Logics

2.1 $ASPIC^+$ Frameworks

$ASPIC^+$ [15, 16] is a rule based argumentation framework, in which arguments are constructed by strict or defeasible rules from the set of premises (called the knowledge base). *Strict rules* are applied to model certain inferences (under a

[3] We first presented *DL-AT* at the DL 2020 workshop as an extended abstract [13].

closed world assumption), while *defeasible rules* are applied to model uncertain inferences. Moreover, based on paper [14], elements in the knowledge base are divided into two types, namely the *axioms* and the *ordinary premises*. The axioms represent firmed knowledge, while the ordinary premises represent only plausible knowledge. Adopted from [14], an argumentation theory based on $ASPIC^+$ is defined as follows.

Definition 1. *An argumentation theory (AT) is a tuple* $(\mathcal{L}, \mathcal{R}, \mathcal{K}, n)$*, where*

- $\mathcal{L}$ *is a formal language closed under negation* $\neg$*;*[4]
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ *is a set of strict* ($\mathcal{R}_s$) *and defeasible* ($\mathcal{R}_d$) *inference rules of the forms* $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$ *and* $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ *(*$n \geq 0$*,* $\varphi_i, \varphi \in \mathcal{L}$*) respectively,* $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$*;*
- $\mathcal{K} \subseteq \mathcal{L}$ *is a knowledge base consisting of a set of axioms* ($\mathcal{K}_n$) *and a set of ordinary premises* ($\mathcal{K}_p$)*,* $\mathcal{K}_n \cap \mathcal{K}_p = \emptyset$*;*
- n *is a naming function, s.t.* $n : \mathcal{R} \rightarrow \mathcal{L}$*.*

In order to include all the arguments that should be constructed under an argumentation theory, the set $\mathcal{R}_s$ of strict rules must be closed under transposition[5], i.e. it holds that: if $\varphi_1, \ldots, \varphi_n \rightarrow \psi \in \mathcal{R}_s$, then for each $i = 1, \ldots, n$, there is $\varphi_1, \ldots, \varphi_{i-1}, -\psi, \varphi_{i+1}, \ldots \varphi_n \rightarrow -\varphi_i \in \mathcal{R}_s$.

Arguments in $ASPIC^+$ are constructed like trees, starting with elements in $\mathcal{K}$ and linked by rules in $\mathcal{R}$. Several basic parts of an argument can be noted: all the formulas in $\mathcal{K}$ that are used to build the argument (denoted by $Prem$), all its sub-arguments (denoted by Sub), all applied defeasible rules (denoted by $DefRules$, while all applied rules can be denoted by $Rules$), the last rule it applied (denoted by $TopRule$), and the consequent of the last rule (dentoed by $Conc$). Therefore, an argument can be defined as follows.

Definition 2. *Let* $AT = (\mathcal{L}, \mathcal{R}, \mathcal{K}, n)$ *be an argumentation theory, an argument* A *under it has one of the following forms:*

- φ *if* $\varphi \in \mathcal{K}$*,* $Prem(A) = \{\varphi\}$*,* $Conc(A) = \varphi$*,* $Sub(A) = \{\varphi\}$*,* $DefRules(A) = \emptyset$*,* $TopRule(A) = undefined$*;*
- $A_1, \ldots, A_n \rightarrow / \Rightarrow \psi$ *if* $A_1, \ldots, A_n$ $(n \geq 1)$ *are arguments, s.t. there exists a strict/defeasible rule* $Conc(A_1), \ldots, Conc(A_n) \rightarrow / \Rightarrow \psi$ *in* $\mathcal{R}$*, and* $Prem(A) = Prem(A_1) \cup \ldots \cup Prem(A_n)$*,* $Conc(A) = \psi$*,* $Sub(A) = Sub(A_1) \cup \ldots \cup Sub(A_n) \cup \{A\}$*,* $DefRules(A) = DefRules(A_1) \cup \ldots \cup DefRules(A_n)$ $(\cup\{Conc(A_1), \ldots, Conc(A_n) \Rightarrow \psi)$*,* $TopRule(A) = Conc(A_1), \ldots, Conc(A_n) \rightarrow / \Rightarrow \psi$*.*

Based on an argumentation theory, the inconsistency of information can be reflected by conflicts (attack relations) among arguments. $ASPIC^+$ supports modeling of three types of attacks: 1) *rebut* an argument on the conclusion of defeasible rules, 2) *undercut* an argument on the defeasible rules and 3) *undermine* an argument on the ordinary premises. Adopted from [14], the attack relation between arguments can be defined as follows.

[4] We write $\psi = -\varphi$ when $\psi = \neg\varphi$ or $\varphi = \neg\psi$ $(\varphi, \psi \in \mathcal{L})$.

[5] or contrapositon [14]

Definition 3. *An argument A attack an argument B under $AT = (\mathcal{L}, \mathcal{R}, \mathcal{K}, n)$, iff A undercuts, rebuts or undermines B, where:*

- *A undercuts B on B', iff $B' \in Sub(B)$ s.t. $TopRule(B') = r \in \mathcal{R}_d$ and $Conc(A) = -n(r)$[6];*
- *A rebuts B on B', iff $B' \in Sub(B)$ of the form $B'_1, \ldots, B'_n \Rightarrow \varphi$ and $Conc(A) = -\varphi$;*
- *A undermines B on B', iff $B' = \varphi$ and $\varphi \in Prem(B) \cap \mathcal{K}_p$, s.t. $Conc(A) = -\varphi$.*

Based on the set of arguments and the set of attack relations, a structured argumentation framework (SAF) is a tuple $\langle \mathcal{A}, att, \preceq \rangle$, where $\mathcal{A}$ is the set of all the constructed arguments, att is the set of attack relations between arguments[7], and $\preceq$ is a preordering over $\mathcal{A}$. Let $A, B \in \mathcal{A}$ be two arguments, according to $\preceq$, we can decide whether the attack from A to B is 'successful' [14]. If it is, then we say that A *defeats* B. Based on $\mathcal{A}$ and the defeat relations, an abstract argumentation framework can be established, which is defined as follows.

Definition 4. *Given an $AT = (\mathcal{L}, \mathcal{R}, \mathcal{K}, n)$, let $SAF = \langle \mathcal{A}, att, \preceq \rangle$ be a structured argumentation framework constructed based on AT. An abstract argumentation framework (AF) for AT is a tuple $\langle \mathcal{A}, \mathcal{D} \rangle$, where $\mathcal{D}$ is a set of defeat relations according to att and $\preceq$.*

Based on paper [6], AF is a directed graph, in which arguments are represented as nodes and the attack (defeat) relations are represented as arrows between nodes. Given an AF, whether an argument is accepted can be evaluated according to *argumentation semantics* [2, 6]. The classical argumentation semantics include the *complete semantics*, the *preferred semantics*, the *grounded semantics* and the *stable semantics* [6]. A set of arguments that can be collectively accepted under certain argumentation semantics is called an *extension*. Let $\mathcal{S}$ denote one of the above-mentioned argumentation semantics, $\mathcal{E}_\mathcal{S}$ denote the set of all extensions obtained under $\mathcal{S}$, $E_\mathcal{S} \in \mathcal{E}_\mathcal{S}$ denote one of the extensions (which can also be called as an $\mathcal{S}$ *extension*). Adopted from [6], the classical argumentation semantics can be defined as follows.

Definition 5. *Given an $AF = \langle \mathcal{A}, \mathcal{D} \rangle$, let $A, B, C \in \mathcal{A}$ be arguments. An extension $E \subseteq \mathcal{A}$ is **conflict-free** iff $\nexists A, B \in E$ s.t. $(A, B) \in \mathcal{D}$; argument A is **defended** by E (or **acceptable** w.r.t. E), iff $\forall B \in \mathcal{A}$, if $(B, A) \in \mathcal{D}$, then $\exists C \in E$ s.t. $(C, B) \in \mathcal{D}$, then: E is **admissible** iff E is conflict-free and each argument in $E \subseteq \mathcal{A}$ is defended by E; E is a **complete extension** iff E is admissible, and each argument in $\mathcal{A}$ that is defended by E is in E; E is a **grounded extension** iff E is the minimal (w.r.t. set-inclusion) complete extension; E is a **preferred extension** iff E is a maximal (w.r.t. set-inclusion) complete extension; E is a **stable extension** iff E is conflict-free and E defeats each argument that is not in E.*

[6] '$n(r)$' means that rule r is applicable.

[7] Let A, B be two arguments in $\mathcal{A}$, $(A, B) \in att$ iff A attacks argument B according to Definition 3.

Given an AF, we say that an argument A is *accepted* w.r.t. $E_{\mathcal{S}}$ if $A \in E_{\mathcal{S}}$. We say that A is *skeptically justified* under $\mathcal{S}$ if $\forall E_{\mathcal{S}} \in \mathcal{E}_{\mathcal{S}}$, $A \in E_{\mathcal{S}}$, and A is *credulously justified* under $\mathcal{S}$ if $\exists E_{\mathcal{S}} \in \mathcal{E}_{\mathcal{S}}$ s.t. $A \in E_{\mathcal{S}}$. According to the extensions of arguments, we can obtain the sets of accepted conclusions.

2.2 Mapping from Description Logics to Argumentation Theory

Description logics are a family of knowledge representation formalism. The basic notions of description logic systems are concepts and roles. Complex concepts and roles can be constructed from basic ones by different constructors. A description logic system contains two disjoint parts: the TBox and the ABox. TBox introduces the terminology, while ABox contains facts about individuals in the application domain. With different constructors, different description logics are constructed. In this paper, we discuss legal ontologies based on the $\mathcal{ALC}$ expression, which is one of the basic language of description logics.

When inconsistency occurs in a DL ontology, traditional reasoners always issue an error message and stop reasoning. People have to find the partitions that cause the inconsistency and repair the problem. However, this 'debugging' process could be very difficult, especially when the ontology is large. Considering the cost of computation and the difficulty of obtaining certain and complete information, this approach is inefficient or even impractical. Instead, in some cases, we may expect the reasoner can still perform reasoning based on the inconsistent knowledge and provides us a temporarily reasonable answer. Meanwhile, we need some convincing and pithy explanations of the result. As mentioned above, $ASPIC^+$ has advantages in achieving these goals. Therefore, our idea is to perform such tasks by translating DL ontologies into argumentation theories based on $ASPIC^+$.

As specified in Definition 1, $ASPIC^+$ framework does not stipulate its logical language, only asks for a contrary relation defined over it. Taking advantage of this unrestriction, in the current paper we adopt first order logic, which is also the traditional logical basis of DL, to represent the elements in the set of language $\mathcal{L}$ of the hybrid theory *DL-AT*.

To translate a DL ontology into an argumentation theory, a **concept** C as a unary predicate is represented as $C(x)$, while a **role** P is represented as a binary predicate $P(x, y)$, x, y are individuals. $C \sqcap D$ and $C \sqcup D$ can be regarded as $C(x) \wedge D(x)$ and $C(x) \vee D(x)$ respectively, and the basic inference on concept expressions in DL is subsumption, written as $C \sqsubseteq D$. Formulas in the ABox are contained in the knowledge base $\mathcal{K}$ of *DL-AT*. As for the TBox, inspired by [9], we translate the declarations in it into strict or defeasible rules, as shown in Table 1. In the table, C, D are basic concepts, P, Q are roles, x, y are individuals and α, β are free variables. Moreover, we use $\dashrightarrow$ to denote one of $\rightarrow$ and $\Rightarrow$ that respectively express strict rules and defeasible rules. A formula can only be translated to one type of rule according to specified context.

Note that in Table 1, we translate $C \sqsubseteq D \sqcup Z$ in TBox into three rules. To clarify this translation, assume that there exists a declaration formed of $C \sqsubseteq D \sqcup Z$ in the DL ontology. According to our translation, three arguments

Table 1. Mapping from TBox of DL ontology to rules in the AT

TBox of DL	rules in $\mathcal{R}$ of AT
$C \sqsubseteq D$	$C(x) \dashrightarrow D(x)$
$C \equiv D$	$D(x) \dashrightarrow C(x)$ & $C(x) \dashrightarrow D(x)$
$C \sqsubseteq \forall P.D$	$C(x), P(x,y) \dashrightarrow D(y)$ & $C(x) \dashrightarrow P(x,\alpha)$
$P \sqsubseteq Q$	$P(x,y) \dashrightarrow Q(x,y)$
$P \equiv Q$	$P(x,y) \dashrightarrow Q(x,y)$ & $Q(x,y) \dashrightarrow P(x,y)$
$C \sqsubseteq \exists P.D$	$C(x) \dashrightarrow P(x,\beta)$ & $C(x), P(x,\beta) \dashrightarrow D(\beta)$
$C \sqcap D \sqsubseteq \emptyset$	$C(x) \dashrightarrow \neg D(x)$ & $D(x) \dashrightarrow \neg C(x)$
$C \sqsubseteq D \sqcup Z$	$C(x) \dashrightarrow D(x) \vee Z(x)$ & $C(x), \neg D(x) \dashrightarrow Z(x)$ & $C(x), \neg Z(x) \dashrightarrow D(x)$
$C \sqsubseteq D \sqcap Z$	$C(x) \dashrightarrow D(x)$ & $C(x) \dashrightarrow Z(x)$

A_1, A_2 and A_3 may be constructed, with $TopRule(A_1) = C(x) \dashrightarrow D(x) \vee Z(x)$, $TopRule(A_2) = C(x), \neg D(x) \dashrightarrow Z(x)$ and $TopRule(A_3) = C(x), \neg Z(x) \dashrightarrow D(x)$. Meanwhile, since the rule $C(x), \neg D(x) \dashrightarrow Z(x)$ is applied, there must exist an argument $A_2' \in Sub(A_2)$ such that $Conc(A_2') = \neg D(x)$. For the same reason, there exists an argument $A_3' \in Sub(A_3)$ such that $Conc(A_3') = \neg Z(x)$. Apparently, it is counter-intuitive if A_1, A_2' and A_3' are accepted simultaneously, because $\neg D(x)$ and $\neg Z(x)$ together contradict $D(x) \vee Z(x)$. However, based on Definition 3, A_2' and A_3' are conflicting with A_3 and A_2 respectively. As a consequence, if we have a declaration of the form $C \sqsubseteq D \sqcup Z$ in the TBox, $\{A_1, A_2', A_3'\}$ or $\{A_1, A_2, A_3\}$[8] will not be the subset of any extensions under classical argumentation semantics, since conflict-freeness is a basic property. Therefore, with this translation, the output of an argumentation theory should be in line with human intuition. Meanwhile, it reserves more expressiveness.[9]

3 Reasoning with Inconsistency based on Argumentation Theory

According to the translation of ABox and TBox, in this section we try to represent DL ontology by argumentation theory based on $ASPIC^+$.

3.1 Combining DL Ontology with Argumentation Theory

In brief, the idea is that the assertions in the ABox can be used as the premises of arguments in the argumentation theory, while according to the mapping that shown in Table 1, the terminologies in the TBox can be used as rules for establishing arguments. Accordingly, we use T and A to denote the sets of terminologies and assertions respectively. An argumentation theory based on a DL ontology can be defined as follows.

[8] The extensions of $ASPIC^+$ framework should be closed under sub-arguments.

[9] In paper [9, 10], when translate DL sentences into rules of logic programs and the $DeLP$ system, DL axioms that generate a rule with a disjunction in its head cannot be represented, so that lost some expressiveness.

Definition 6. *Let $\Delta = (T, A)$ be a DL ontology, a DL-AT_Δ= $(\mathcal{L}, \mathcal{R}^T, \mathcal{K}^A, n)$ is an argumentation theory constructed with Δ, s.t. $\mathcal{R}^T$ is the set of rules corresponding to T based on Table 1, and $\mathcal{K}^A$ is the set of premises based on A.*

To illustrate how to perform reasoning through *DL-AT*, we introduce an example of an inconsistent DL ontology based on a legal case.

The case was adapted from the Casey Anthony trial in the United States in 2011. Casey was charged with first-degree murder for murdering her daughter Caylee. Several facts and evidences indicated that Casey is likely to be guilty, including: 1. Casey did not report that her daughter was missing for 31 days; 2. Casey lied a lot to the police after her mother called the police; 3. When Caylee's remains were found, there was duct tape covering its nose and mouth. However, the defense attorney argued that Caylee was accidentally drowned in the family pool, and the tape was used to wrap Caylee's remains because Casey was scared. Moreover, he pointed out that the forensic expert did a shoddy autopsy, and all the evidences were circumstantial evidences.

From this case, we can extract the following ontology shown in Example 1.

Example 1. Let $\Delta_1 = (T, A)$ be a DL ontology, where:

$$T = \left\{ \begin{array}{l} FoundRemain \sqcap LethalToolCausedDeath \sqsubseteq BeMurdered; \\ TapeOnNose \sqsubseteq LethalToolCausedDeath; \\ AccidentallyDrownedInFamilyPool \sqsubseteq AccidentalDeath; \\ BeMurdered \sqcap AccidentalDeath \sqsubseteq \emptyset; \\ TapeToWrap \sqcap ShaddyAutopsy \sqsubseteq \neg TapeOnNose; \\ Guardian \sqcap LiedToPolice \sqcap BeMurdered \\ \sqsubseteq CircumstantiallyProvedGuilty; \\ Guardian \sqcap \exists DelayedReportingAbout.Missing \\ \sqsubseteq CircumstantiallyProvedGuilty; \\ DirectEvidencesForGuilty \equiv DirectlyProvedGuilty; \\ CircumstantiallyProvedGuilty \sqsubseteq Guilty; \\ \neg DirectlyProvedGuilty \sqsubseteq \neg Guilty \end{array} \right\}$$

$$A = \left\{ \begin{array}{l} TapeOnNose(Caylee); \\ FoundRemain(Caylee); \\ AccidentallyDrownedInFamilyPool(Caylee); \\ TapeToWrap(Caylee); \\ ShaddyAutopsy(ForensicExpert); \\ Guardian(Casey); \\ LiedToPolice(Casey); \\ DelayedReportingAbout(Casey, Caylee); \\ Missing(Caylee); \\ \neg DirectEvidencesForGuilt(Casey) \end{array} \right\}$$

Example 1 shows a DL ontology extracted from the trial case, which is inconsistent since the statements of the prosecutor's attorney and defense attorney

conflicted with each other. Based on Definition 6, we can represent this ontology based on argumentation theory and get a temporarily acceptable answer by performing reasoning with inconsistent knowledge.

Continuing Example 1, the following sets of rules and premises in the argumentation theory $DL\text{-}AT_{\Delta_1}$ can be obtained based on Δ_1 (x, y, z represent variables of individual).

Example 2 (continues Example 1).

$$\mathcal{R}_d^T = \left\{ \begin{array}{l} r_1 : FoundRemain(y), LethalToolCausedDeath(y) \\ \Rightarrow BeMurdered(y); \\ r_2 : TapeOnNose(y) \Rightarrow LethalToolCausedDeath(y); \\ r_3 : AccidentallyDrownedInFamilyPool(y) \\ \Rightarrow AccidentalDeath(y); \\ r_4 : BeMurdered(y) \Rightarrow \neg AccidentalDeath(y); \\ r_5 : AccidentalDeath(y) \Rightarrow \neg BeMurdered(y); \\ r_6 : TapeToWrap(y), ShaddyAutopsy(z) \Rightarrow \neg TapeOnNose(y); \\ r_7 : Guardian(x), LiedToPolice(x), BeMurdered(y) \\ \Rightarrow CircumstantiallyProvedGuilty(x); \\ r_8 : Guardian(x), DelayedReportingAbout(x, y), Missing(y) \\ \Rightarrow CircumstantiallyProvedGuilty(x); \\ r_9 : CircumstantiallyProvedGuilty(x) \sqsubseteq Guilty(x); \\ r_{10} : \neg DirectlyProvedGuilty(x) \Rightarrow \neg Guilty(x) \end{array} \right\}$$

$$\mathcal{R}_s^{T\,10} = \left\{ \begin{array}{l} r_{11} : DirectEvidencesForGuilty(x) \rightarrow DirectlyProvedGuilty(x); \\ r'_{11} : \neg DirectlyProvedGuilty(x) \\ \rightarrow \neg DirectEvidencesForGuilty(x); \\ r_{12} : DirectlyProvedGuilty(x) \rightarrow DirectEvidencesForGuilty(x); \\ r'_{12} : \neg DirectEvidencesForGuilty(x) \\ \rightarrow \neg DirectlyProvedGuilty(x) \end{array} \right\}$$

$$\mathcal{K}_n^A = \left\{ \begin{array}{l} FoundRemain(Caylee); \\ Guardian(Casey); \\ LiedToPolice(Casey); \\ DelayedReportingAbout(Casey, Caylee); \\ Missing(Caylee); \\ ShaddyAutopsy(ForensicExpert); \\ \neg DirectEvidencesForGuilt(Casey) \end{array} \right\}$$

$$\mathcal{K}_p^A = \left\{ \begin{array}{l} TapeOnNose(Caylee); \\ TapeToWrap(Caylee); \\ AccidentallyDrownedInFamilyPool(Caylee) \end{array} \right\}$$

[10] rule r'_{11} and r'_{12} are the transposed rules of rule r_{11} and r_{12} respectively.

3.2 Inferences Performance

Given a DL ontology, the reasoning tasks based on it include the reasoning for concepts, for the TBox and the ABox respectively, and for the TBox and the ABox together [1].

To perform the reasoning tasks, one of the key points is to make sure all the arguments that can be established in the argumentation system are contained in the set of arguments $\mathcal{A}$. In related works, how to achieve this task is often neglected without proper explanation. The current paper presents an approach to fulfill this goal and the pseudocode is shown in Fig. 1.[11] The main idea of this approach is to traverse the set of premises and set of rules respectively, and hide the premises/antecedents that appear in the body of rules. When all the antecedents of a rule are hidden, an argument will be constructed through the rule and its conclusion will be added to a set of premises/antecedents. The cyclic process continues until no new antecedent shows up. In this way, we can obtain a finite set of arguments containing all the available arguments according to an argumentation theory.

According to Example 2, we can get the following 22 arguments:

$A_1 : FoundRemain(Caylee)$ $A_2 : Guardian(Casey)$
$A_3 : LiedToPolice(Casey)$ $A_4 : DelayedReportingAbout(Casey, Caylee)$
$A_5 : Missing(Caylee)$ $A_6 : \neg DirectEvidencesForGuilt(Casey)$
$A_7 : TapeOnNose(Caylee)$ $A_8 : TapeToWrap(Caylee)$
$A_9 : AccidentallyDrownedInFamilyPool(Caylee)$
$A_{10} : ShaddyAutopsy(ForensicExpert)$
$A_{11} : A_7 \Rightarrow LethalToolCausedDeath(Caylee)$
$A_{12} : A_1, A_{11} \Rightarrow BeMurdered(Caylee)$
$A_{13} : A_9 \Rightarrow AccidentalDeath(Caylee)$
$A_{14} : A_{12} \Rightarrow \neg AccidentalDeath(Caylee)$
$A_{15} : A_9 \Rightarrow \neg BeMurdered(Caylee)$
$A_{16} : A_8, A_{10} \Rightarrow \neg TapeOnNose(Caylee)$
$A_{17} : A_2, A_3, A_{12} \Rightarrow CircumstantiallyProvedGuilty(Casey)$
$A_{18} : A_{17} \Rightarrow Guilty(Casey)$
$A_{19} : A_2, A_4, A_5 \Rightarrow CircumstantiallyProvedGuilty(Casey)$
$A_{20} : A_{19} \Rightarrow Guilty(Casey)$
$A_{21} : A_6 \rightarrow \neg DirectlyProvedGuilty(Casey)$
$A_{22} : A_{21} \Rightarrow \neg Guilty(Casey)$

Based on Definition 3, there are following attack relations between arguments: (A_7, A_{16}), (A_{12}, A_{15}), (A_{13}, A_{14}), (A_{14}, A_{13}), (A_{15}, A_{12}), (A_{15}, A_{14}), (A_{15}, A_{17}), (A_{15}, A_{18}), (A_{16}, A_7), (A_{16}, A_{11}), (A_{16}, A_{12}), (A_{16}, A_{14}), (A_{16}, A_{17}), (A_{16}, A_{18}), (A_{18}, A_{22}), (A_{20}, A_{22}), (A_{22}, A_{18}), (A_{22}, A_{20}).

[11] The rules and arguments have the same format $(a_1, a_2, a_3, \ldots, a_n, +\backslash-, con)$, where $a_1, a_2, \ldots a_n$ are premises, $+$ and $-$ denote the strict rules and the defeasible rules respectively, and con denotes the conclusion. Function $Premise(x)$ is used to obtain the premises of an argument x, $Conclusion(x)$ is used to obtain the conclusion of an argument x, and $Argumentget(r)$ is used to obtain an argument through rule r.

Algorithm 1

Input: **Ruleset** $= \{rule(n)(0)|rule = (a_1, a_2, a_3, \ldots, a_n, +\backslash-, con)\}$;**Premiseset** $= \{a_1, a_2, a_3, \ldots, a_n\}$.

Output: **Argumentset** $= \{A_n : argument|argument = a_n or(a_1, a_2, a_3, \ldots, a_n, +\backslash-, con)\}$.

1: **Argumentset** $= \{A_1 : a_1, A_2 : a_2, \ldots, A_n : a_n\}$; **Premiseset** $= \{a_1, a_2, \ldots, a_n\}$; $max_i = 0, i \in [1, n]$; $number = n$;
2: **for** $premise \in$ **Premiseset do**
3: **for** $rule(i)(max_i) \in$ **Ruleset do**
4: **if** $premise \subset$ **Premise**$(rule(i)(max_i))$ **then**
5: $max_i = max_i + 1, rule(i)(max_i) = rule(i)(max_i - 1).remove(premise)$
6: **if** **Premise**$(rule(i)(max_i)) = \emptyset$ **then**
7: **Premiseset** $=$ **Premiseset** $\cup$ (**Conclusion**$(rule(i)(max_i))$
8: $argument =$ **Argumentget**$(rule(i)(0))$
9: $number = number + 1$
10: **Argumentset** $=$ **Argumentset** $\cup (A_{number} : argument)$
11: **end if**
12: **end if**
13: **end for**
14: **end for**

Fig. 1. The procedure to get all the arguments of AT

Getting Acceptable Knowledge Sets In some situations, facing an inconsistent ontology, we may want to get a set of knowledge that can be collectively accepted rather than an answer for a certain query. This task can be achieved through the argument evaluation procedure based on $DL\text{-}AT$. Extensions of arguments can be obtained according to different standards, i.e. argumentation semantics. A very prudent agent may choose to apply the grounded semantics, while a less prudent agent may prefer to get more consistent results and choose to apply the preferred semantics.

Assuming that all the arguments constructed based on Example 2 have equal strength, then all the attacks succeed as defeats. Accordingly, an AF can be established. We apply the grounded semantics and the preferred semantics as instances. From the AF, we can get one grounded extension $E_g = \{A_1, A_2, A_3, A_4, A_5, A_6, A_8, A_9, A_{10}, A_{19}, A_{21}\}$. Therefore, these arguments are all skeptically justified. While based on preferred semantics, we can get several extensions, and arguments A_{22}, or A_{18} and A_{20} are credulously justified with respect to these extensions.

Acceptance of an Assertion In a *DL-AT*, assertions are the conclusions of arguments.[12] So that based on the extension of arguments, we can decide whether an assertion is accepted. Therefore, the acceptance of assertions can be defined as follows.

Definition 7. *Given a DL-AT* $_{\Delta} = (\mathcal{L}, \mathcal{R}^T, \mathcal{K}^A, n)$*, let* $\mathcal{A}$ *be the set of all the arguments constructed based on DL-AT* $_{\Delta}$*. An assertion* X *is skeptically/credulously accepted under certain argumentation semantics* $\mathcal{S}$*, iff* $\exists A \in \mathcal{A}$*, s.t.* A *is skeptically/credulously justified w.r.t.* $\mathcal{E}_{\mathcal{S}}$ *and* $Conc(A) = X$*.*

Property 1. Let $E_{\mathcal{S}}$ be an extension obtained based on a *DL-AT* $_{\Delta}$ under certain argumentation semantics $\mathcal{S}$ introduced in Definition 5, $\mathcal{O}_{\mathcal{S}} = \{Conc(A) | A \in E_{\mathcal{S}}\}$ is the set of conclusions according to $E_{\mathcal{S}}$, if X, Y are assertions, then $\nexists X, Y \in \mathcal{O}_{\mathcal{S}}$, s.t. $X = -Y$.

In the Casey Anthony trial, although many believed that Casey was unquestionably guilty, the jury, however, felt differently because there is no direct evidence. Casey was acquitted of the murder charge at last. According to this fact, the jurors' preferences are obvious: they believed that direct evidences were more convincing than circumstantial evidences. Therefore, $r_9 \preceq r_{10}$ and not vice versa. Based on the *the last link principle* for obtaining preferences over arguments provided by $ASPIC^+$ [14], argument A_{22} would be skeptically justified. Consequently, we can see why the conclusion "$\neg Guilty(Casey)$" was adopted by the jurors.

Consistency Checking Consistency checking of the ontology is translated into the consistency checking of the set of arguments, which can be defined as follows.

Definition 8. *Given a DL ontology* $\Delta = (T, A)$ *and a DL-AT* $_{\Delta} = (\mathcal{L}, \mathcal{R}^T, \mathcal{K}^A, n)$*. Let* $\mathcal{A}$ *contain all the arguments that can be established, the ABox of* Δ *is consistent w.r.t. the TBox of* Δ *iff* $\mathcal{A}$ *is conflict-free based on attack relations, i.e.,* $\nexists A, B \in \mathcal{A}$ *s.t.* A *attacks* B*.*

According to the arguments constructed based on Example 2, we can draw the conclusion that the ontology Δ_1 shown in Example 1 is inconsistent, because the set $\mathcal{A} = \{A_1, A_2, \ldots, A_{22}\}$ is not conflict-free based on attack relations.

Instances Checking To determine whether a particular individual is an instance of a concept, we translate this problem into whether an assertion about this individual can be accepted as a conclusion of an accepted (justified) argument.

The following definition defines instance checking based on a *DL-AT* $_{\Delta}$ for all the possible forms of classes.

[12] Assertions can also be the premises in $\mathcal{K}$, whereas according to Definition 2, if an argument has the form $A = \varphi$ and $\varphi \in \mathcal{K}$, then $Conc(A) = \varphi$.

Definition 9. *Let φ be an individual, skeptically or credulously, it holds that φ is an instance of class:*

- *C ($/\neg C$), iff $\exists A \in \mathcal{A}$, s.t. A is skeptically/credulously justified w.r.t. $\mathcal{E}_{\mathcal{S}}$ and $Conc(A) = C(\varphi)(/\neg C(\varphi)$;*
- *$C \sqcap D$, iff $\exists A, B \in \mathcal{A}$ s.t. A and B are both skeptically/credulously justified w.r.t. $\mathcal{E}_{\mathcal{S}}$ and $Conc(A) = C(\varphi)$, $Conc(B) = D(\varphi)$;*
- *$C \sqcup D$, iff $\exists A, B \in \mathcal{A}$ s.t. at least one of A and B are skeptically/credulously justified w.r.t. $\mathcal{E}_{\mathcal{S}}$ and $Conc(A) = C(\varphi)$, $Conc(B) = D(\varphi)$;*
- *$\exists P.D$, iff $\exists A, B \in \mathcal{A}$ s.t. A and B are both skeptically/credulously justified w.r.t. $\mathcal{E}_{\mathcal{S}}$ and $Conc(A) = P(\varphi, x)$, $Conc(B) = D(x)$ (x is an individual);*
- *$\forall P.D$, iff 1) $\exists A \in \mathcal{A}$ s.t. $Conc(A) = P(\varphi, x)$; and 2) $\forall A \in \{A | Conc(A) = P(\varphi, x)\}$, $\exists B \in \mathcal{A}$, s.t. $Conc(B) = D(x)$.*

Explanation To explain the acceptance of an assertion, we give an explanation of why an argument A that concludes this assertion is accepted, which is defined as follows.

Definition 10. *Assuming that X is a skeptically/credulously accepted assertion under certain argumentation semantics $\mathcal{S}$, then $\exists A \in \mathcal{A}$ s.t. $Conc(A) = X$, the explanation of*

1. *how this assertion is reached is $Prem(A) \cup Rule(A)$;*
2. *why this assertion is accepted is $Prem(B) \cup Rule(B)$ along with $\preceq$, s.t. B defends A based on defeat relations.*

Definition 10 provides a formal explanation of why an assertion X concluded by argument A is accepted, which consists of two parts. The first part explains how X is reached by presenting all the premises contained in $\mathcal{K}^A$ and all the rules contained in $\mathcal{R}^T$ that applied to construct argument A. In other words, it indicates all the declarations contained in the ABox and the TBox of a DL ontology that used to conclude X. The second part explains why this assertion is accepted by presenting all the premises and rules applied to construct the arguments that defend A. Similarly, this explanation indicates all the relevant declarations of the DL ontology.

Consider Example 1, if we propose a query about whether Casey is not guilty, we can check if there exists a skeptically justified argument whose conclusion is $\neg Guilty(Casey)$. Apparently, according to the jurors' preference, argument A_{22} is skeptically justified w.r.t. the grounded/preferred extensions and $Conc(A_{22}) = \neg Guilty(Casey)$. So we can reach a conclusion that $\neg Guilty(Casey)$ is skeptically accepted. $Exp_1 = \{\neg DirectEvidencesForGuilt(Casey)\} \cup \{r'_{12}, r_{10}\}$ offers an explanation of how this assertion is reached, while $Exp_2 = \{TapeToWrap(Caylee), ShaddyAutopsy(ForensicExpert)\} \cup \{r_6\}$ and $\preceq_{jury}$ can offer an explanation of why this assertion is accepted.

Retrieval Consider that in some cases, people want to obtain all individuals that are instances of a concept C, or want to know all the concepts that an individual x is an instance of, the following two methods can be applied to answer the above two questions respectively.

Given a $DL\text{-}AT_{\Delta}$, we can get a set $\mathcal{E}_{\mathcal{S}}$ of all the extensions under an argumentation semantic $\mathcal{S}$. For an extension $E_s \in \mathcal{E}_{\mathcal{S}}$, let $\mathcal{O}_{\mathcal{S}} = \{Conc(A)|A \in E_s\}$ be the set of conclusions according to E_s, we have

Method 1: to obtain all the individuals that are instances of concept C, we find all the individual x s.t. $C(x) \in \mathcal{O}_{\mathcal{S}}$ based on Definition 7.

Method 2: to obtain all the concepts that x is an instance of, we find all the concepts C s.t. $C(x) \in \mathcal{O}_{\mathcal{S}}$ based on Definition 9.

4 Conclusions and Future work

In this paper we propose a combined theory called *DL-AT* based on the DL ontology and $ASPIC^{+}$ framework to handle reasoning with inconsistent legal ontologies. Firstly, we translate the DL ontology into formulas in the knowledge base and rules of argumentation theory. According to the translation, we define the *DL-AT* and show how to perform reasoning with inconsistency based on it. Particularly, we give the pseudocode of an algorithm for ensuring that all the constructible arguments are obtained. In addition, we define how to fulfill the traditional reasoning tasks of DL with *DL-AT* and provide a formal definition of explanation.

In paper [14, 16], properties for a well-defined argumentation theory are specified. A well defined *DL-AT* should adopt all relevant requirements presented in [14]. In addition, some desired properties (called *rationality postulates*) for rule-based argumentation systems are proposed in [5]. These postulates should be satisfied by a well defined *DL-AT*. Property 1 given in this paper corresponds to the postulate of *direct consistency*. However, the proofs are omitted due to space limitation. We would like to present these contents in future work. What is more, in future work, we plan to investigate how to apply our approach to more expressive description logic languages such as SHI. We will investigate how to perform other reasoning tasks, such as integrating different legal ontologies of the same domain. In addition, we will explore the possibility of combining this approach with data driven approaches such representation learning.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: Theory, implementation, and applications. Cambridge University Press (2003)
2. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. The Knowledge Engineering Review **26**(4), 365–410 (2011)
3. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence **128**(1), 203 – 235 (2001)

4. Breuker, J., Valente, A., Winkels, R.: Use and Reuse of Legal Ontologies in Knowledge Engineering and Information Management, pp. 36–64. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
5. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artificial Intelligence (2007)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321 – 357 (1995)
7. Gangemi, A., Sagri, M.T., Tiscornia, D.: A Constructive Framework for Legal Ontologies, pp. 97–124. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
8. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming **4**(2), 95–138 (jan 2004)
9. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Reasoning with inconsistent ontologies through argumentation. Applied Artificial Intelligence **24**(1&2), 102–148 (2010)
10. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: Proceedings of the 12th International Conference on World Wide Web. p. 48–57. WWW 03, Association for Computing Machinery, New York, NY, USA (2003)
11. Hoekstra, R., Breuker, J., Bello, M.D., Boer, A.: The LKIF core ontology of basic legal concepts. In: Casanovas, P., Biasiotti, M.A., Francesconi, E., Sagri, M. (eds.) Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques June 4th, 2007, Stanford University, Stanford, CA, USA. CEUR Workshop Proceedings, vol. 321, pp. 43–63. CEUR-WS.org (2007)
12. Hoekstra, R., Breuker, J., Bello, M.D., Boer, A.: LKIF core: Principled ontology development for the legal domain. In: Breuker, J., Casanovas, P., Klein, M.C.A., Francesconi, E. (eds.) Law, Ontologies and the Semantic Web - Channelling the Legal Information Flood. Frontiers in Artificial Intelligence and Applications, vol. 188, pp. 21–52. IOS Press (2009)
13. Lu, Y., Yu, Z.: Argumentation theory for reasoning with inconsistent ontologies (extended abstract). In: Borgwardt, S., Meyer, T. (eds.) Proceedings of the 33rd International Workshop on Description Logics (DL 2020), Online, September 12th to 14th, 2020. CEUR Workshop Proceedings, vol. 2663. CEUR-WS.org (2020)
14. Modgil, S., Prakken, H.: A general account of argumentation with preferences. Artificial Intelligence **195**, 361–397 (2013)
15. Modgil, S., Prakken, H.: The aspic+ framework for structured argumentation: a tutorial. Argument & Computation **5**(1), 31–62 (2014)
16. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument & Computation **1**(2), 93–124 (2010)
17. Valente, A.: Legal Knowledge Engineering: A Modelling Approach. IOS Press (1995)
18. Valente, A., Breuker, J., Brouwer, B.: Legal modeling and automated reasoning with on-line. International Journal of Human-Computer Studies **51**, 1079–1125 (December 1999)
19. Van Engers, T., Boer, A., Breuker, J., Valente, A., Winkels, R.: Ontologies in the Legal Domain, pp. 233–261. Springer US, Boston, MA (2008)
20. Zhang, X., Lin, Z.: An argumentation framework for description logic ontology reasoning and management. Journal of Intelligent Information Systems **40**(3), 375–403 (2013)

COLIEE 2020: Methods for Legal Document Retrieval and Entailment

Juliano Rabelo[1,2], Mi-Young Kim[1,3], Randy Goebel[1,2], Masaharu Yoshioka[4,5], Yoshinobu Kano[6], and Ken Satoh[7]

[1] Alberta Machine Intelligence Institute, Edmonton AB, Canada
[2] University of Alberta, Edmonton AB, Canada
{rabelo,miyoung2,rgoebel}@ualberta.ca
[3] Department of Science, Augustana Faculty, Camrose AB, Canada
[4] Graduate School of Information Science and Technology, Hokkaido University, Kita-ku, Sapporo-shi, Hokkaido, Japan
yoshioka@ist.hokudai.ac.jp
[5] Global Station for Big Date and Cybersecurity, Global Institution for Collaborative Research and Education, Kita-ku, Sapporo-shi, Hokkaido, Japan
[6] Faculty of Informatics, Shizuoka University, Naka-ku, Hamamatsu-shi, Shizuoka, Japan
kano@inf.shizuoka.ac.jp
[7] National Institute of Informatics, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
ksatoh@nii.ac.jp

Abstract. This paper presents a summary of the 7th Competition on Legal Information Extraction and Entailment. The competition consists of four tasks on case law and statute law. The case law component includes an information retrieval task (Task 1), and the confirmation of an entailment relation between an existing case and an unseen case (Task 2). The statute law component includes an information retrieval task (Task 3) and an entailment/question answering task (Task 4). Participation was open to any group based on any approach. Ten different teams participated in the case law competition tasks, most of them in more than one task. We received results from 9 teams for Task 1 (22 runs) and 8 teams for Task 2 (22 runs). On the statute law task, there were 14 different teams participating, most in more than one task. Eleven teams submitted a total of 28 runs for Task 3, and 13 teams submitted a total of 30 runs for Task 4. We describe in this paper the approaches, our official evaluation, and analysis on our data and submission results.

Keywords: Legal Documents Processing · Textual Entailment · Information Retrieval · Classification · Question Answering.

1 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) intends to develop the state of the art for information retrieval and entailment using legal texts. It is usually co-located with JURISIN, the Juris-Informatics

workshop series, which was created to promote community discussion on both fundamental and practical issues on legal information processing, with the intention to embrace various disciplines, including law, social sciences, information processing, logic and philosophy, including the existing conventional "AI and law" area. In alternate years, COLIEE is organized as a workshop the International Conference on AI and Law (ICAIL), which was the case in 2017 and 2019. Until 2017, COLIEE consisted of two tasks: information retrieval (IR) and entailment using Japanese Statute Law (civil law). Since COLIEE 2018, IR and entailment tasks using Canadian case law were introduced.

Task 1 is a legal case retrieval task, and it involves reading a new case Q, and extracting supporting cases S_1, S_2, ..., S_n from the provided case law corpus, hypothesized to support the decision for Q. Task 2 is the legal case entailment task, which involves the identification of a paragraph or paragraphs from existing cases, which entail a given fragment of a new case. For the information retrieval task (Task 3), based on the discussion about the analysis of previous COLIEE IR tasks, we modify the evaluation measure of the final results and ask participants to submit ranked relevant articles results to discuss the detailed difficulty of the questions. For the entailment task (Task 4), we performed categorized analyses to show different issues of the problems and characteristics of the submissions, in addition to the evaluation accuracy as in previous COLIEE tasks.

The rest of the paper is organized as follows: Sections 2, 3, 4, 5 describe each task, presenting their definitions, datasets, list of approaches submitted by the participants, and results attained. Section 6 presents final some final remarks.

2 Task 1 - Case Law Information Retrieval

2.1 Task Definition

This task consists in finding which cases, in the set of candidate cases, should be "noticed" with respect to a given query case. "Notice" is a legal technical term that denotes a legal case description that is considered to be relevant to a query case. More formally, given a query case q and a set of candidate cases $C = \{c_1, c_2, ..., c_n\}$, the task is to find the supporting cases $S = \{s_1, s_2, ..., s_n \mid s_i \in C \wedge noticed(s_i, q)\}$ where $noticed(s_i, q)$ denotes a relationship which is true when $s_i \in S$ is a noticed case with respect to q.

2.2 Dataset

The training dataset consists of 520 base cases, each with 200 candidate cases from which the participants must identify those that should be noticed with respect to the base case. The training dataset contains a total of 104,000 candidates cases with 2,680 (2.57%) being true noticed cases. The official COLIEE test dataset has 130 cases. For those cases, the golden labels are only disclosed after the competition results were published. The test dataset has a total of 26,000 candidates cases with 636 (2.44%) being true noticed cases.

2.3 Approaches

Eight teams submitted a total of 22 runs for this task. IR techniques and machine learning based classifiers were commonly used. More details are shown below:

- **cyber (three runs)** [18] created a method based on a selection of the top 30 candidate cases using a paragraph similarity score based on a universal sentence encoder, and then applied an SVM model based on the vector representation between base case and candidate case in TF-IDF space. The base method is augmented by applying additional auto-weighting of classes in SVM training and by using a TF-IDF vectorizer trained on all available texts, including test samples.
- **UB (two runs)** [6] uses a Learning to Rank approach with features generated from Terrier weighting models such as BM25 and TF-IDF. All documents from the training and test datasets were used to build the ranking model. A Learning to Rank approach with a combination of text similarity and distance metrics' generated features was also used.
- **iiest (three runs)** [9] applied filtered-bag-of-ngrams (FiBONG), BM25 and other techniques in three runs. FiBONG is an extended version of BOW and uses several pre-processing filters (stopword removal, POS filtering, lemmatization, etc.) over unigrams, bigrams and trigrams. The first run used BM25 upon a FiBONG representation of the case documents. In the second run, the FiBONG representation was used with a different scoring function. A modified version of BM25 where the new IDF term is multiplied with a standardized and normalized value of the collection frequency. The third run used the FiBONG representation with a different scoring function called "PSlegal" [8].
- **TLIR (three runs)** [14] applied the word-entity duet (weduet) framework which uses 11 interaction features to generate the ranking scores. The authors also submitted a run based on the usage of the BERT-PLI framework, with one-layer forward GRU as the RNN component. The uncased-base BERT model is used and fine-tuned on the data of COLIEE 2019 Task 2. LMIR (Language model for Information Retrieval) is used to select top-30 candidate cases in the first stage. Last, they use the 11-dimensional features in "weduet" and extract the output vector (2-dimensional) of the softmax function in "bertgru". The authors also apply the seed-driven Document Ranking algorithm and obtain 2-dimensional features (similarity scores calculated based on words and entities, respectively). Then the first paragraph of the query and a candidate case are used as input of BERT to fine-tune a sentence pair classification task and extract the vector (2-dimensional) given by the softmax function as additional features. In total, 17-dimensional features are obtained and applied to a RankSVM model.
- **TR (three runs)** [5] used a ranking approach followed by a classification task. First the the candidate cases for a given case are ranked based on their similarity. Next the dataset is split into subdatasets based on their ranks to classify if a candidate case is a supporting case. The ranking task is

straightforward and does not require specific parameters. XGBoost is used for the classification task.
- **AUT99 (three runs)**, which applied a model based on CEDR[7] with different parameters. The authors haven't submitted a paper with a detailed description of their method.
- **DACCO (one run)** hasn't submitted a paper describing the method used.
- **TAXI (one run)** [1] uses Catboost with the following features as input: 400 word limit summarized documents input to Count Vectorizer with n-gram ranged 1-2 and 60,000 maximum features and TF-IDF with IDF smoothing.
- **JNLP (three runs)** [10] applies a system which is based on the BERT base model, fine-tuned for a text-pair classification task. The text-pairs are extracted from candidate cases using designed heuristics. The text-pair supporting scores and lexical matching scores (BM25) are computed from comparing paragraph-paragraph to measure query case-candidate case relevance. Machine learning model and setting: BERT [3] with 768 hidden nodes, 12 layers, 12 attention heads, 110M parameters, 512 max input length.

2.4 Results

The F1-measure is used to assess performance in this task. We use a simple baseline model that uses the Universal Sentence Encoder to encode each candidate case and base case into a fixed size vector, and then applies the cosine distance between both vectors. The baseline result was 0.3560. The actual results of the submitted runs by all participants are shown on table 1, with the cyber team attaining the best F1 score. TLIR and cyber also achieved good results.

Table 1. Results attained by all teams on the test dataset of task 1.

Team	File	F1	Team	File	F1
cyber	task1 cyber02.txt	0.6774	TLIR	t1_run1_thuir.txt	0.5148
cyber	task1 cyber03.txt	0.6768	iiest	iiest_ps_t1_1.txt	0.4821
TLIR	t1_run3_thuir.txt	0.6682	TR	submission2	0.3800
cyber	task1 cyber01.txt	0.6503	TR	submission3	0.3792
JNLP	JNLP.task1.BMW25.txt	0.6397	TR	submission1	0.3388
TLIR	t1_run2_thuir.txt	0.6379	AUT99	AUTIRT1R1.txt	0.2658
JNLP	JNLP.task1.W25.txt	0.6358	DACCO	T1 DACCO.txt	0.2077
JNLP	JNLP.task1.W30.txt	0.6278	AUT99	AUTIRT1R2.txt	0.1617
UB	UB_RUN1.res	0.5866	AUT99	AUTIRT1R3.txt	0.0898
iiest	iiest_bm26_t1_3.txt	0.5288	UB	UB_RUN2.res	0.0592
iiest	iiest_bm25_t1_2.txt	0.5272	taxi	task1_TAXICATTFCV.txt	0.0457

3 Task 2 - Case Law Entailment

3.1 Task Definition

Given a base case and a specific fragment from it, and a second case relevant to the base case, this task consists in determining which paragraphs of the second

case entail that fragment of the base case. More formally, given a base case b and its entailed fragment f, and another case r represented by its paragraphs $P = \{p_1, p_2, ..., p_n\}$ such that $noticed(b, r)$ as defined in section 2 is true, the task consists in finding the set $E = \{p_1, p2, ..., p_m \mid p_i \in P\}$ where $entails(p_i, f)$ denotes a relationship which is true when $p_i \in P$ entails the fragment f.

3.2 Dataset

The training dataset has 325 base cases, each with its respective entailed fragment in a separate file. For each base case, a related case represented by a list of paragraphs is given, from which the paragraph(s) that entail the base-case-entailed fragment must be identified. The training dataset contains 11,494 paragraphs in the related cases, 374 (3.25%) of which are true entailing paragraphs. The test dataset has 100 cases and was initially released without the golden labels, which were only disclosed after the competition results were published. It contains 3,672 paragraphs, with 125 (3.40%) being true entailing paragraphs.

3.3 Approaches

Eight teams submitted a total of 22 runs to this task. The most used techniques were those based on transformer methods, such as BERT [3] or ELMo [11]. More details on the approaches are show below.

- **cyber (three runs)** [18], whose method is based on the selection of top 10 candidate paragraphs, using a sentence similarity score based on a universal sentence encoder, and then applying an SVM model based on the vector formed between the base case and candidate case representations in TF-IDF. The authors also submitted runs augmenting the base approach and training a TF-IDF vectorizer on all available texts, including test samples and excluding certain anomalous samples excluded from training.
- **DACCO (one run)** hasn't submitted a paper describing the method used.
- **iiest (three runs)** [9] based their submissions in techniques such as filtered-bag-of-ngrams (FiBONG) and BM25 as in task 1. The first run used BM25 upon a FiBONG representation of the case documents. In the second run, the FiBONG representation was used with a different scoring function. A modified version of BM25 where the new IDF term is multiplied with a standardized and normalized value of the collection frequency. The third run used centroids of word embeddings to represent the candidate paragraphs and the base judgements. Cosine distance was used to measure similarity. The word embeddings are taken from Law2Vec[8].
- **JNLP (three runs)** [10] applied an approach similar to the one used in Task 1. The system has a model capturing the supporting relation of a text pair, based on the BERT base model, then fine-tuned for a supporting text-pair classification task. The set of supporting text-pairs includes the text-pairs

[8] https://archive.org/details/Law2Vec

from Task 1 candidate cases using designed heuristics, and the gold data of Task 2 (decision-paragraph). The system also has a BERT model fine-tuned on SigmaLaw (a law dataset) for the masked language modeling task. Together with scoring by the BERT models, lexical matching (BM25) is also considered for predicting decision-paragraph entailment.

- **tax-i (three runs)** [1] applied an Xgboost classifier with the following features as input: NLI probability (bert-nli), similarity between entailed fragment and paragraphs based on fine-tuned BERT (bert-base-uncased), and BM25 similarity between entailed fragment and paragraphs. The authors also submitted runs using other features as input: n-grams, BM25, NLI, and EUR-LEX (81,000 sentences from EU legal documents) fine-tuned ROBERTA and BERT (bert-base-uncased) derived similarity features.

- **TLIR (three runs)** [14] fine-tune BERT (uncased-base) in a sentence pair classification task. If the total input tokens exceed the length limitation (512), the texts are truncated symmetrically. The model is trained for no more than 5 epochs with $lr = 1e{-}5$ and selected according to the F1 measure on the validation set. The difference in the second run is the truncation of text asymmetrically. They limit the tokens of decision fragment to 128 and only truncate the tokens in the candidate paragraph if the total length of the text pair exceeds 512 tokens. In their last run, the authors extract the output vector of the fully-connected layer of the two previous models (4-dimensional in total) as features. Besides, they calculate the BM25 scores (1-dimensional). The position ID and the length of the paragraph are used as 2 additional features. In total, 7-dimensional features are generated and then a RankSVM model is applied.

- **TR (three runs)** [5], whose approach consists of two stages: (1) similarity features-based ranking and (2) Random Forest binary classification. Paragraphs are ranked according to a criterion that combines the individual ranks given by the cosine similarity coefficients obtained using different sentence vectorizers (n-grams, universal sentence encoder, averaged glove embeddings, topic modelling probability scores). The likelihood of the relevant paragraph falling into the top K paragraphs is estimated for different values of K using the training data. Then for a specific value of likelihood, similarity features are computed on the top K paragraphs and fed to a random forest classifier.

- **UA (three runs)** [12], which applied transformer-based techniques to generate features which were then fed to a Random Forest classifier. The features were generated by fine-tuning a pre-trained BERT model text entailment on the provided training dataset and using the score produced in this task, two transformer-based models fine-tuned on a generic entailment data set, and another one applying zero-shot techniques by using BERT fine tuned for paraphrase detection. They also used data augmentation techniques based on back translation to increase the size of the training data.

3.4 Results

The F1-measure is used to assess performance in this task. The score attained by a simple baseline model which uses the Universal Sentence Encoder to encode each candidate paragraph and the entailed fragment into a fixed size vector and applies the cosine distance between both vectors was 0.1760. The actual results of the submitted runs by all participants are shown on table 2, from which it can be seen that the JNLP team attained the best results. The TAXI and TLIR teams also achieved good results for the F1-score.

Table 2. Results attained by all teams on the test dataset of task 2.

Team	Submission File	F1-score	Team	Submission File	F1-score
JNLP	JNLP.task2.BMWT.txt	0.6753	cyber	task2 cyber01.txt	0.5600
JNLP	JNLP.task2.BMW.txt	0.6222	TLIR	t2_run3_thuir.txt	0.5495
taxi	t2-taxiXGBaft.txt	0.6180	TLIR	t2_run1_thuir.txt	0.5428
TLIR	t2_run2_thuir.txt	0.6154	UA	UA1.txt	0.5425
JNLP	JNLP.task2.WT+L.txt	0.6094	UA	UA2.txt	0.5179
taxi	t2-taxiXGBaf.txt	0.5992	iiest	iiest_l2v_t2_3.txt	0.5067
taxi	t2-taxiXGB3f.txt	0.5917	UA	UA_translate.txt	0.4647
cyber	task2 cyber03.txt	0.5897	TR	submission1.txt	0.4107
iiest	iiest_bm25_t2_1.txt	0.5867	TR	submission3.txt	0.4107
iiest	iiest_bm26_t2_2.txt	0.5867	TR	submission2.txt	0.4018
cyber	task2 cyber02.txt	0.5837	DACCO	T2 DACCOr.txt	0.0622

4 Task 3 - Statute Law Retrieval

4.1 Task Definition

This task is involves reading a legal bar exam question Q, and retrieve a subset of Japanese Civil Code Articles S_1, S_2,..., S_n to judge whether the question is entailed or not ($Entails(S_1, S_2, ..., S_n, Q)$ or $Entails(S_1, S_2, ..., S_n, notQ)$).

4.2 Dataset

For task 3, questions related to Japanese civil law were selected from the Japanese bar exam. Since there was update of Japanese Civil Code at April 2020, we revised text for reflecting this revision for Civil Code and its translation into English. However, since English translated version is not provided for a part of this code, we exclude these parts from the civil code text and questions related to these parts. As a results number of the articles used in the dataset is 768. Training data (the questions and relevant article pairs) was constructed by using previous COLIEE data (696 questions). In this data, questions related to revised articles are reexamined and ones for excluded articles are removed from the training data. For the test data, new questions selected from the 2019 bar exam are used (112 questions). The number of questions classified by the number of relevant articles is as follows (1 answer: 87, 2 answers 22, 3 answers 3).

4.3 Approaches

The following 11 teams submitted their results (28 runs in total). Three teams (HUKB, JNLP, and UA) had participated in previous COLIEE editions, and eight teams (CU, Cyber, GK_NLP, HONto, LLNTU, OvGU, TAXI, TRC3) were new competitors. Compared to previous years, many teams use BERT[3] for analyzing text. From the results, BERT-based approach is good for improve the retrieval quality. In addition, this approach also allows the team to select two or more articles for one question. Other common techniques used were well known IR engines such as elasticsearch Indri [15], Hierachical Optimal Topic Tranport (HOTT) [20] based on topic model, gensim, scikit-learn with various scoring function such as TF-IDF, BM25. For the indexing of ordinal IR system, the most common method was ordinal word base indexing with stemming. Several teams use N-gram, word sequence, word embedding using legal texts.

- **CU (three runs)** [2] uses TF-IDF and BERT model with different settings.
- **cyber (three runs)** [18] calculate similarity between the sentence in the articles using TF-IDF and BM25 and aggregate the results.
- **GK_NLP (one run)** GK_NLP uses elastic search using TF-IDF model.
- **HONto (three runs)** [17] uses HOTT for calculating the similarity between question and article using different word embedding methods.
- **HUKB (three runs)** [19] uses BERT-based IR system and Indri for the IR module and compare the result of each system output to make final results.
- **JNLP (three runs)** [10] uses BERT model with different settings to classify the articles are relevant or not.
- **LLNTU (one run)** [13] uses BERT to classify articles as relevant or not.
- **OvGU (three runs)** [17] uses TF-IDF and BM25 with different indexing methods.
- **TAXI (three runs)** [1] uses TF-IDF model and IR model that uses word embeddings based on the legal texts.
- **TRC3 (three runs)** [5] uses TF-IDF for the basic IR system and Wikibooks on Japanese civil law to calculate similarity between the query and articles.
- **UA (two runs)** uses TF-IDF and language model as an IR module.

4.4 Results

Table 3 shows the evaluation results of submitted runs (due to page limit constraints, only the best run in terms of F2 is selected from each team). The official evaluation measures used in this task were macro average of precision, recall and F2 measure. We also calculate the mean average precision (MAP), recall at k (R_k: recall calculated by using the top k ranked documents as returned documents) by using the long ranking list (100 articles).

This year, LLNTU is the best among all runs. JNLP achieves good performance when they submit an answer. However, since there are several questions that returns no relevant article, overall performance of JNLP is lower than LLNTU. We confirmed recent development of deep learning technology based on BERT is also effective to retrieve relevant articles for the questions.

Table 3. Evaluation results (Task 3)

sid	lang	return	retrieved	F2	Precision	Recall	MAP
LLNTU	J	122	**84**	**0.659**	**0.688**	**0.662**	**0.760**
JNLP.tfidf-bert-ensemble	E	104	76	0.553	0.577	0.567	0.662
cyber1	E	204	70	0.529	0.506	0.554	0.554
HUKB-1	J	250	75	0.516	0.420	0.591	0.569
CUBERT1	E	126	68	0.514	0.540	0.519	0.585
TRC3_1	J	159	65	0.501	0.456	0.536	0.598
OvGU_bm25	E	248	69	0.477	0.400	0.534	0.510
TAXI_R3	E	230	64	0.455	0.439	0.509	0.506
GK_NLP	E	224	64	0.427	0.286	0.499	0.498
UA.tfidf	E	112	48	0.391	0.429	0.387	0.478
HONto_hybrid	E	162	36	0.282	0.254	0.299	0.014

Figure 1.2 shows average of evaluation measure for all submission runs. As we can see from left part of Figure 1, there are many easy questions that almost all system can retrieve the relevant article. Easiest question is R01-12-U whose relevant article is almost same as a question. However, there are also many queries for which none of the systems can retrieve the relevant articles (Figure 1 right). R1-14-U[9] is an example of this question. The relevant article is Article 87 [10]. It is necessary to interpret the relationship between the "building and leased land" in the question as "first thing attaches a second thing" in the article. Even though BERT is good at ranking articles that take into account the context, it is difficult to estimate such interpretation that requires legal knowledge to interpret the context.

One characteristic difference from the previous COLIEE is improvement of the retrieval quality for questions with multiple answers. In the previous COLIEE most of the team returned only one article for each question to keep a good precision. This year, many teams returned two or more answers to such questions. As a result, there are 4 questions whose recall is higher than 0.5. For COLIEE 2019, there were no questions with multiple answers with recall higher than 0.5.

5 Task 4 - Statute Law Entailment

5.1 Task Definition

Task 4 is a task to determine entailment relationships between a given problem sentence and article sentences. Competitor systems should answer "yes" or "no" regarding the given problem sentences and given article sentences. Until COLIEE 2016, the competition had pure entailment tasks, where t1 (relevant article sentences) and t2 (problem sentence) were given. Due to the limited number of available problems, COLIEE 2017, 2018 did not retain this style of task. In the Task 4 of COLIEE 2019 and 2020, we returned to the pure textual entailment task to attract more participants, allowing more focused analyses.

[9] "In cases where a mortgage is created with respect to a building on leased land, the mortgage may not be exercised against the right of lease."

[10] "(1) If the owner of a first thing attaches a second thing that the owner owns to the first thing to serve the ordinary use of the first thing, the thing that the owner attaches is an appurtenance. (2) An appurtenance is disposed of together with the principal thing if the principal thing is disposed of."

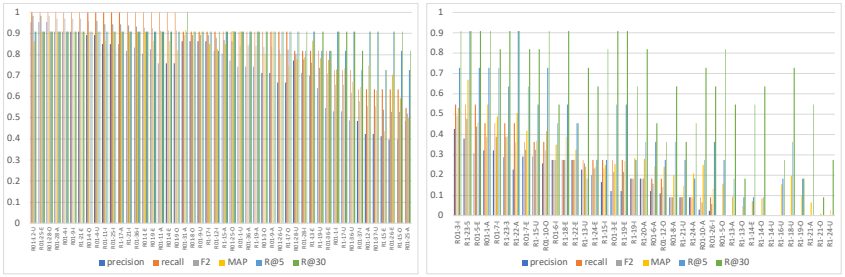


Fig. 1. Avg. of prec., rec., F2, MAP, R_5 and R_30 (questions with 1 relevant article)

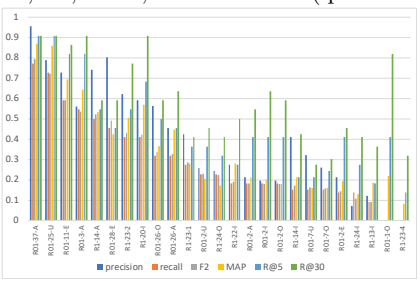


Fig. 2. Avg. of prec., rec., F2, MAP, R_5 and R_30 (questions with 1+ relevant article)

5.2 Dataset

Our training dataset and test dataset are the same as Task 3. Questions related to Japanese civil law were selected from the Japanese bar exam. The organizers provided a data set used for previous campaigns as training data (768 questions) and new questions selected from the 2019 bar exam as test data (112 questions).

5.3 Approaches

The following 12 teams submitted their results (30 runs in total). 3 teams (JNLP, KIS, and UA) had experience in submitting results in the previous campaign. We describe each system's overview below.

- **CU (two runs)** [2] uses multilingual cased BERT model for sequence classification trained and evaluated only with given relevant articles (CUGIVEN), plus additional articles returned using TFIDF (CUPLUS).
- **cyber (one runs)** [18] uses RoBERTa based method which is fine-tuned on sentence pair classification with the SNLI corpus. The resulting model fine-tuned on text pair classification with COLIEE training data.
- **GK_NLP (one run)** uses similarity measure on BERT embeddings and GloVe word embeddings with lightgbm classification model.
- **HONto (three runs)** [17] uses linear kernel SVM with TF-IDF and n-grams.
- **JNLP (three runs)** [10] uses BERT; JNLP.BERT and JNLP.TfidfBERT with the Google's original BERT_Base, JNLP.BERTLaw pretrained by American cases of 8.2M sentences/182M words in English. JNLP.BERTLaw and JNLP.BERT were fine-tuned by lawfulness classification on Augmented JAPAN

Civil Code + COLIEE training data; JNLP.TfidfBERT was fine-tuned by COLIEE training data, no cross-fold validation.
- **KIS (three runs)** [4] built a range of Japanese legal dictionaries for predicate argument structures and paraphrases, which can integrate PROLEG, an legal logic language. KIS is their rule-based ensemble NLP system; KIS_2 uses SVM instead of rules in KIS; KIS_3 uses PROLEG to answer some of the questions in KIS_2.
- **LLNTU (one run)** [13] combines each query from COLIEE training dataset with all civil law articles, trains BERT-based ensemble models.
- **OvGU (three runs)** [17] uses Bidirectional LSTM and a modified Bahdanau's attention with inputing Law2Vec embeddings (baseline_attention. task4.OvGU), with the similarity measure and negations (sim_neg.task4. OvGU), adding POS of each token (POS_simneg.task4.OvGU).
- **tax-i (two runs)** [1] uses legal embeddings (FastText trained on US Caselaw) as input to a Bi-directional GRU with 128 Hidden Layers and 1 GRU layer (LEBIGRU), last hidden state of BERT base-cased was used as input to an XGBoost classifier (BERTXGB).
- **TRC3 (three runs)** [5] uses GloVe word embedding. Multee (TRC3mt) was trained phase one against single sentence NLI datasets (SNLI, MutliNLI) and then trained phase two on multiple sentence NLI datasets (OpenbookQA, COLIEE). The Text-To-Text Transfer Transformer (T5) run (TRC3t5) was fine-tuned on three denoising tasks (Civil Code Article, Civil Code Titles, Translated Wikibook Articles) and one entailment task (COLIEE).
- **UA (three runs)** [12] uses a decomposable attention model, which is a simple neural architecture for natural language inference, decomposing a problem into sub-problems (UA_attention_final), a RoBERTa trained model (UA_roberta_final), their previous model that showed the best performance in COLIEE 2019 (UA_structure).
- **UEC (three runs)** [16] translates t1 texts into an easier one (t1p) with a paraphrase dictionary, extracts subject/predicate/object tuples from the main and conditional clauses, then constructs tuple-based similarity features for ¡t1p, t2¿ pair (UEC1 and UEC2), for both the ¡t1, t2¿ and ¡t1p, t2¿ pairs (UECplus). LightGBM is used for binary classification.

5.4 Results

Table 4 shows evaluation results of Task 4 (accuracy was the metric used). Because an entailment task is a complex composition of different subtasks, we manually categorized our test data into categories, depending on what sort of technical issues are required to be resolved. Table 5 shows our categorization results. As this is a composition task, overlap is allowed between categories. Our categorization is based on the original Japanese version of the legal bar exam.

6 Final Remarks

In this paper we summarized the results of COLIEE 2020. Task 1 deals with the retrieval of noticed cases, and Task 2 poses the problem of identifying which

Table 4. Evaluation results of submitted runs (Task 4). L: Dataset Language (J: Japanese, E: English), #: number of correct answers (112 problems in total)

Team	L	#	Accuracy	Team	L	#	Accuracy
JNLP.BERTLaw	E	81	0.7232	KIS_3	J	61	0.5446
TRC3mt	E	70	0.6250	sim_neg.OvGU	E	61	0.5446
TRC3t5	E	70	0.6250	UEC1	J	61	0.5446
UA_attention_final	?	70	0.6250	taxi_BERTXGB	E	60	0.5357
UA_roberta_final	?	70	0.6250	UECplus	J	60	0.5357
KIS_2	J	69	0.6161	CUGIVEN	E	58	0.5179
llntu	J	69	0.6161	CUPLUS	E	58	0.5179
cyber	E	69	0.6161	linearsvm_no_ngram.HONto	E	57	0.5089
UA_structure	?	68	0.6071	POS_simneg.OvGU	E	57	0.5089
GK_NLP	?	63	0.5625	taxi_le_bigru	E	57	0.5089
linearsvm.HONto	E	63	0.5625	TRC3A	E	56	0.5000
JNLP.BERT	E	63	0.5625	UEC2	J	55	0.4911
KIS	J	63	0.5625	baseline_attention.OvGU	E	54	0.4821
linearsvm_no_ngram.HONto	E	62	0.5536	AUT99-BERT-MatchPyramid	E	52	0.4643
JNLP.TfidfBERT	E	62	0.5536	AUT99-LSTM-CNN-Attention	E	50	0.4464

paragraphs of a relevant case entail a given fragment of a new case. Task 3 is about retrieving articles to decide the appropriateness of the legal question, and Task 4 is a task to entail whether the legal question is correct or not. Ten (10) different teams participated in the case law competition (most of them in both tasks). We received results from 9 teams for Task 1 (a total of 22 runs), and 8 teams for Task 2 (a total of 22 runs). Regarding the statute law tasks, there were 14 different teams participating, most in both tasks. Eleven (11) teams submitted 28 runs for Task 3, and 13 teams submitted 30 runs for Task 4.

A variety of methods were used for Task 1: exploitation of the case structure information, deep learning based techniques (such as transformer methods and tools such as the Universal Sentence Encoder), lexical and latent features, different text embedding techniques, information retrieval techniques and different classifiers (such as tree based and SVM) were the main ones. For Task 2, transformer-based tools were used (among which BERT was prevalent), but IR techniques and textual similarity features have also been applied. Some teams leveraged techniques similar to the ones they developed for task 1, which shows the tasks are somewhat connected. The results attained were satisfactory, but there is much room for improvement, especially if one considers the related issue of explaining the predictions made; deep learning methods, which attained the best results this year, would not be so appropriate in a scenario where explainability is necessary. For future editions of COLIEE, we plan on continuing expanding the data sets in order to improve the robustness of results, as well as evaluating ways of introducing explainability-aware tasks into the competition.

For Task 3, we confirmed that BERT-based approach improves overall retrieval performance. However, there are still numbers of questions that are difficult to retrieve by any systems. It is better to discuss the type of information necessary to find out the relationship between question and articles for the next step. For Task 4, overall performance of the submissions is still not sufficient to use their systems in real applications, mainly due to lack of coverage for some

Table 5. Technical category statistics of questions, correct answer ratios of submitted runs for each category in percentages sorted in the order of ranks for each run.

Team Rank	Conditions	Predicate argument	Negation	Legal fact	Person role	Person Relationship	Verb Paraphrase	Morpheme	Dependency	Anaphora	Entailment	Normal terms	Case role	Article search	Itemized	Normal terms	Ambiguity	Calculation
Total #	74	73	69	55	48	48	41	33	24	23	20	16	14	12	11	3	1	1
1	**.42**	**.44**	.43	**.42**	**.40**	**.42**	.44	.58	.46	.52	.55	.38	.50	.50	.64	.00	1.00	1.00
2	.49	.48	.46	.47	.46	.46	.44	.45	.58	**.35**	.50	.44	.50	.42	**.18**	.33	1.00	.00
3	.53	.51	.59	.58	.56	.56	.59	.36	.42	.48	.40	.50	.57	.58	.27	.67	.00	.00
4	.53	.51	.59	.58	.56	.56	.59	.36	.42	.48	.40	.50	.57	.58	.27	.67	.00	.00
5	.62	.64	**.70**	.60	.60	.60	.61	.64	.67	.52	.45	.69	.64	.50	.45	.67	1.00	1.00
6	.61	.52	.51	.53	.46	.48	.49	.70	.63	.57	.60	.56	.43	.67	.36	.00	1.00	.00
7	.57	.53	.58	.55	.52	.52	.49	.64	.46	.48	.60	.44	**.36**	**.25**	.45	.67	1.00	.00
8	.54	.53	.54	.53	.50	.50	.46	.67	.54	.52	.60	.50	**.36**	**.25**	.45	.67	1.00	.00
9	.50	.48	.55	**.42**	.44	.44	.49	.64	.54	.39	.55	.38	**.36**	**.25**	.36	.67	1.00	1.00
10	.53	.56	.49	.53	.52	.54	.59	.79	.63	.61	.35	.69	.57	.75	.64	.33	.00	.00
11	**.64**	**.78**	.68	**.67**	**.73**	**.73**	**.78**	**.91**	**.75**	**.74**	**.80**	**.75**	.64	.58	.55	**1.00**	1.00	.00
12	.59	.51	.54	.58	.54	.58	.51	.58	.50	.43	.55	.56	.64	**.83**	.64	.67	.00	.00
13	.59	.55	.61	.51	.50	.48	.51	.67	.58	.57	.60	.56	.57	**.25**	.45	.67	1.00	1.00
14	.59	.62	.58	.62	.60	.63	.63	.82	.71	.65	.55	.56	.64	.75	.64	.33	.00	.00
15	.57	.52	.58	.47	.50	.48	.49	.67	.50	.61	.60	.56	.50	.33	.45	.67	1.00	.00
16	.61	.63	.59	.60	.60	.60	.66	.64	.67	.61	.60	.56	.71	.75	.64	.67	1.00	.00
17	.54	.48	.57	.55	.48	.48	.46	**.33**	**.33**	.57	.55	**.31**	.50	.42	.55	.67	1.00	.00
18	.57	.53	.52	.49	.48	.50	.56	.52	.58	.52	.55	.63	.64	.58	.55	.67	1.00	.00
19	.53	.59	.59	.58	.58	.58	.61	.52	.58	.57	.55	.56	.71	.67	.55	.33	1.00	1.00
20	.55	.55	.51	.45	.44	.44	.49	.58	.54	.57	.65	.50	.57	.42	.64	.67	.00	.00
21	.57	.52	.58	.53	.48	.48	.54	.39	.46	.57	.50	.44	.71	.58	**.73**	.67	1.00	.00
22	.49	.49	.49	.55	.52	.54	.46	.58	.58	.43	**.30**	.63	.43	.67	.55	.67	.00	.00
23	.62	.62	.67	.55	.60	.60	.61	.61	.71	.65	.55	.44	.64	.50	.55	**1.00**	.00	1.00
24	.58	.64	.58	.62	.52	.54	.63	.67	.67	.65	.55	.56	**.79**	.75	.64	.67	.00	.00
25	.58	.62	.65	.62	.63	.63	.56	.67	.58	.61	.60	.63	.64	.58	.45	.67	.00	1.00
26	.53	.66	.59	.62	.63	.60	.49	.70	.63	.70	.55	.63	.64	.50	.36	.33	1.00	1.00
27	.58	.60	.61	.58	.54	.52	.54	.73	.67	.57	.55	**.75**	.57	.42	.64	**1.00**	1.00	1.00
28	.57	.56	.51	.55	.60	.63	.49	.48	.67	.61	.55	.56	.50	.58	.55	.33	.00	1.00
29	.46	.49	**.42**	.49	.52	.56	**.41**	.45	.54	.52	.50	.50	.43	.67	.45	.33	.00	.00
30	.54	.49	.54	.62	.58	.60	.46	.52	.54	.48	.35	.63	.50	.75	.64	.67	.00	.00

classes of problems, such as anaphora resolution. We found this task is still a challenging one to discuss and develop deep semantic analysis issues in the real application and natural language processing in general.

Acknowledgements

This research was supported by the National Institute of Informatics, Shizuoka University, Hokkaido University, and the University of Alberta's Alberta Machine Intelligence Institute (Amii). Special thanks to Colin Lachance from vLex for his unwavering support in the development of the case law data set, and to continued support from Ross Intelligence and Intellicon.

References

1. Alberts, H., Ipek, A., Lucas, R., Wozny, P.: COLIEE 2020: Legal information retrieval & entailment with legal embeddings and boosting. In: COLIEE (2020)
2. Aydemir, A., de Castro Souza, P., Gelfman, A.: Using BERT and TF-IDF to predict entailment in law-based queries. In: COLIEE (2020)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
4. Hayashi, R., Kiyota, N., Fujita, M., Kano, Y.: Legal bar exam solver integrating legal logic language proleg and argument structure analysis with legal linguistic dictionary. In: COLIEE (2020)
5. Hudzina, J., Madan, K., Chinnappa, D., Harmouche, J., Bretz, H., Vold, A., Schilder, F.: Information extraction & entailment of common law & civil code. In: COLIEE (2020)
6. Leburu-Dingalo, T., Thuma, E., Motlogelwa, N., Mudongo, M.: Ub_Botswana at COLIEE 2020 case law retrieval. In: COLIEE (2020)
7. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: Cedr: Contextualized embeddings for document ranking. In: SIGIR (2019)
8. Mandal, A., Ghosh, K., Pal, A., Ghosh, S.: Automatic catchphrase identification from legal court case documents. In: Proceedings of the Conference on Information and Knowledge Management. p. 2187–2190. ACM, New York, NY, USA (2017)
9. Mandal, A., Ghosh, S., Ghosh, K., Mandal, S.: Significance of textual representation in legal case retrieval and entailment. In: COLIEE (2020)
10. Nguyen, H.T., Vuong, H.Y.T., Nguyen, P.M., Dang, B.T., Bui, Q.M., Vu, S.T., Nguyen, C.M., Tran, V., Satoh, K., Nguyen, M.L.: JNLP team: Deep learning for legal processing in COLIEE 2020. In: COLIEE (2020)
11. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
12. Rabelo, J., Kim, M.Y., Goebel, R.: Application of text entailment techniques in COLIEE 2020. In: COLIEE (2020)
13. Shao, H.L., chia Chen, Y., Huang, S.: BERT-based ensemble model for the statute law retrieval and legal information entailment. In: COLIEE (2020)
14. Shao, Y., Liu, B., Mao, J., Liu, Y., Zhang, M., Ma, S.: THUIR@COLIEE-2020: Leveraging semantic understanding and exact matching for legal case retrieval and entailment. In: COLIEE (2020)
15. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. pp. 2–6 (2005)
16. Suematsu, Y., Matsuyoshi, S., Utsumi, A.: Recognizing textual entailment for japanese legal text using lexical simplication and tuple-based matching and similarity features. In: COLIEE (2020)
17. Wehnert, S., Murugadas, V., Nandakumar, S., Saha, A., Khan, T.R., Urban, M., Luca, E.W.D.: Legal information retrieval and entailment detection: Hybrid approaches of traditional machine learning and deep learning. In: COLIEE (2020)
18. Westermann, H., Šavelka, J., Benyekhlef, K.: Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment. In: COLIEE (2020)
19. Yoshioka, M., Suzuki, Y.: HUKB at COLIEE 2020 information retrieval task. In: COLIEE (2020)
20. Yurochkin, M., Claici, S., Chien, E., Mirzazadeh, F., Solomon, J.: Hierarchical optimal transport for document representation (2019)

COLIEE 2020: Legal Information Retrieval & Entailment with Legal Embeddings and Boosting*

Houda Alberts[0000−0001−6924−5995], Akin Ipek[0000−0002−1363−5254], Roderick Lucas[0000−0002−5689−2197], and Phillip Wozny[0000−0001−9417−7170]

Deloitte NL, tax-i team, Amsterdam, Netherlands
{hdevries-alberts,aipek,rlucas,pwozny}@deloitte.nl

Abstract. In this paper we investigate three different methods for several legal document retrieval and entailment tasks; namely, new pre-trained embeddings, specifically trained on documents in the legal domain, transformer models and boosting algorithms. Task 1, a case law retrieval task, utilized a pairwise CatBoost resulting in an F1 score of .04. Task 2, a case law entailment task, utilized a combination of BM25+, embeddings and natural language inference (NLI) features winning third place with an F1 of 0.6180. Task 3, a statutory information retrieval task, utilized the aforementioned pre-trained embeddings in combination with TF-IDF features resulting in an F2 score of 0.4546. Lastly, task 4, a statutory entailment task, utilized BERT embeddings with XGBoost and achieved an accuracy of 0.5357. Notably, our Task 2 submission was the third best in the competition. Our findings illustrate that using legal embeddings and auxiliary linguistic features, such as NLI, show the most promise for future improvements.

Keywords: Legal Information Retrieval · Textual Entailment · Classification · Natural Language Inference · Ranking · Legal Embeddings · BERT · Boosting

1 Introduction

Search engines have become the gateway to the internet for both the layman and the scholar alike [25]. Google, the largest search engine by market share [6], has become an indispensable tool for academic researchers [33]. However, legal researchers have unique needs that require unique search tools [1]. As the methods that search engines use to rank results shape how the user interacts with web content [12], it is critical that legal researchers have access to tools designed with them in mind. Given the size of corpora that legal researchers must search through, any methods to increase the efficiency of legal research

* Supported by the tax-i team within Deloitte. Each author contributed equally and is responsible for one specific task; In order of the authors, responsible for task 3, 1, 2 and 4.

have disruptive potential for the industry [7]. The use of machine learning in the legal domain has the potential to reduce the amount of time required for legal research and reasoning [7]. Within tax-i [1], we develop machine learning tools built with legal researchers in mind, offering specialized search functionalities such as cross-lingual search.

Two main machine learning applications in the legal domain are entailment and information retrieval [[7], [32]]. Entailment aims to address the question of whether a given given proposition is true or false based on a piece of evidence [26]. Machine learning systems can then reason over entailed claims [22]. Information retrieval involves searching through a corpus of documents and ranking them according to their relevance to a query [19]. These are only two examples of how machine learning can disrupt the legal industry through the automation of costly, yet repetitive tasks [13].

As a means of cultivating research in these domains University of Alberta, along with a host of co-sponsors, hosts the Competition on Legal Information Extraction/Entailment (COLIEE). In its current iteration, COLIEE comprises four tasks. Task one requires identifying the set of cases from a case law corpus which support the decision of a query case. Given the decision fragment of a case that is supported by another case, task two aims to discern which paragraph of the supporting case entail the fragment. Tasks three and four use statutory data and bar exam questions. Task three involves retrieving statutory articles relevant to a bar exam question. Task four aims to determine the entailment of a bar exam question by a set of relevant articles.

1.1 Contributions

Our aims in COLIEE are the following:

- We intend to employ the state of the art natural language processing (NLP) methods to address the four tasks.
- Second, we address the fact that many state of the art language models do not transfer well to the legal domain. We do this by training our own embeddings on legal text.

The rest of the paper is as follows: Section 2 continues with the related work and Section 3 explains our novel approach with our legal embeddings. Next, in Section 4, 5, 6, and 7 we will focus on the methodology, experimental set-up and results for task 1, 2, 3 and 4 respectively. Finally, Section 8 concludes our paper with possible extensions for our research.

2 Related Work

In this section, we will focus on related research on both legal information retrieval and legal entailment. Although both serve different purposes, they can be addressed using common methods.

[1] https://tax-i.deloitte.nl/

Classic information retrieval techniques include BM-25 [21] and TF-IDF [18], which obtain normalized bag of word representations of a corpus of documents and a query. The aforementioned are then compared to rank the queried documents by relevancy. Such practices are common in the legal domain [[14], [10], [29]].

Richer document representation methods introduced in legal information retrieval include, Doc2Vec [16] and more advanced transformer methods [28], such as BERT [8]. One shortcoming of BERT is that it takes a fixed sequence length of 512 tokens, which is problematic given the length of legal documents [2]. Previous attempts to address this shortcoming have involved using summarization tools such as Gensim [23] or trimming sequences to the maximum length [10]. Another downside of out-of-the-box BERT is its inability to generalize to specific domains due to the fact that it was trained on Wikipedia-like data [11].

BERT can also be used only as means of generating features from text without invoking the entire transformer architecture [8]. Common methods of using BERT derived features include taking the mean of the last four hidden layers, taking a weighted sum of the last four hidden layers, or just using the last hidden layer [8]. BERT derived features can then be used as input to a separate model [10].

Both information retrieval and entailment tasks can be configured as classification problems through pairwise relevance prediction and binary classification, respectively. Legal classification problems can be addressed through deep learning methods. For instance, Chalkidis, Androutsopoulos & Aletras (2019) employed a Bi-Directional Gated Recurrent Unit with Attention (Bi-GRU-ATT) model [4] to predict the outcome of European Court of Human Rights (ECHR) cases. Bi-GRU-ATT models are useful in the legal domain because they are sensitive to context [3]. Previous COLIEE submissions have illustrated the effectiveness of encoding entailment inputs into separate Long Term Short Term Memory (LSTM) models whose combined output is used for binary classification [29].

Non deep learning methods can also be used for classification, such as K-Nearest Neighbor, Random Forest, and boosting algorithms [23]. The downside of this latter problem framing is that class imbalanced become more common due to more non-relevant documents; however, tree-based approaches such as XGBoost [5] or CatBoost [9] tend to handle such imbalances better. Furthermore, tree based methods can make use of meta information, such as the date or header, alongside textual features [31].

3 Legal Embeddings

As explained in Section 2, most competitive embedding based models are trained on a general corpus, such as Wikipedia or (short) stories. However, when applied to legal data, the results fall short. Legal English is different than regular English

[2] https://eur-lex.europa.eu/legal-content/NL/ALL/?uri=CELEX%3A61996CJ0349

with respect to syntax, semantics, vocabulary and morphology, which explains this shortage in performance [30]. Based on these findings, we trained a legal FastText model [3] at the start of this year for our own applications within our intelligent information retrieval system, tax-i. The need for legal embeddings is verified by recent research that has found that training a legal BERT does aid in legal-based entailment and question answering [11].

To train our FastText [2] model, we make use of a partition of legal data that we have available in our tax-i platform. We only use US-related content and take roughly 1/3rd of this which translates to 1M US cases. We apply the pre-processing on this as needed by our models (i.e. lowercase, punctuation removal etc.) and use this during training.

We then use the unsupervised FastText [4] to train embeddings with the legal data via a skipgram model, training for 4 epochs, 6 threads, no wordngrams, a 300-dimensionality for the word embeddings and all remaining parameters remain default. This yields a final loss of around ~ 0.05. The legal embeddings resulted in sub-par performance on tasks one and two during initial experimentation; therefore, alternative methods were employed.

4 Task 1: Case Law Information Retrieval

The first task focuses on case law information retrieval where given a case law document, Q, we want to retrieve relevant case law to this document from a finite set of candidates $S_1, S_2, ...S_n$.

4.1 Methodology

We formulate this retrieval task into a pairwise classification problem meaning, where each candidate is labeled as relevant or not. We use English case law and have a fixed amount of candidates per case, namely 200, where the candidates differ for each base case.

To classify each candidate, we conjoin the query document and a summary of the candidate case, extracted using either Gensim or regex, to generate TF-IDF with IDF weight smoothing and absolute word count features using uni- and bi-grams. These features are then put into a boosting algorithm, either XGBoost [5] or CatBoost [20], where the latter has shown promising results in text-related tasks [[24], [27]]. Afterwards, we use precision, recall and the F1 measure to assess how this model performs by only taking the ones classified as relevant into account for these measures. That is, we do not take the non-relevant cases into account to get a proper estimation of the classification in this retrieval task.

4.2 Experimental Setup

We use a 90-10% split on the training data to obtain a training and validation set. Using absolute counts and TF-IDF features with XGBoost and CatBoost

[3] We opted for FastText rather than BERT due to limited computational resources.
[4] https://github.com/facebookresearch/fastText

resulted in F1 scores of 0.37 and 0.54, respectively. As such, we determined the superior method to be CatBoost with 1500 iterations, a learning rate of 0.1, l2 leaf regularization of 3.5, and depth of 4.

4.3 Results

Among the COLIEE 2020 submissions this year the used method, absolute counts and TF-IDF into CatBoost, we obtain a test F1 score of 0.0457, where the top-3 submissions obtain 0.6774 (team cyber), 0.6768 (team cyber) and 0.6682 (team TLIR) in descending order. The difference between the final test results and the training data test results was due to human error in the method of document summarization employed. That is, the model was trained on Gensim summarized documents and the model was evaluated on regex extracted summary documents. Applying Gensim summarization to the 2020 test data resulted in an actual F1 score of .18. The drop in performance is indicative of overfitting. This can be caused by the use of a pairwise TF-IDF approach which depends on a shared vocabulary between train and test data. This can be overcome by using the cosine distance between TF-IDF vectors as a feature. Furthermore, repeated evaluation on the training data showed high variance model performance with values between 0.43 and 0.54.

5 Task 2: Case Law Entailment

The case law entailment task requires finding an entailing paragraph from a case, given a base case with a specified fragment. Or put formally: Given a base case, Q, its entailed fragment, f, and another related case, r, composed of the paragraph set $P = \{p_1, p_2, \ldots, p_n\}$, find the paragraph set $E = \{p_1, p_2, \ldots, p_m \mid p_i \in P\}$, such that p_i entails f.

5.1 Methodology

The three feature groups are ensembled in a XGBoost classifier in order to predict the probability of p_i entailing f. The feature groups are classical features, embedding similarity and Natural Language Inference (NLI).

Feature Group 1 - Classical Features These consist of classical word features, such as length of the paragraph p_i, place of paragraph within the article, count of overlapping words between f and p_i and a BM-25 ranking.

Feature Group 2 - Embedding Similarity This consists of the similarity of the means of three different word-embeddings: RoBERTa [17], BERT [8] and a fine-tuned version of the latter all uncased on the COLIEE 2020 training data. As a similarity measure between the embedded p_i and the embedded f, we use the cosine similarity.

Feature Group 3 - Natural Language Inference NLI is the task of determining whether two statements entail or contradict each other. Ideally this would be done utilizing a fine-tuned model for the legal domain. However, due to the training and validation time constraints, we use the pre-trained BERT NLI model.

Ensemble model The previous mentioned feature groups are all min-max normalized per base case Q. Predictions are the probabilistic output of a XGBoost classifier. For every base case Q, we check how many paragraphs of p_i are predicted to be entailing fragment f. Within the test set this is either one or two. If no paragraphs are found for a base case, the algorithm picks the one with the highest probabilistic outcome (albeit it with a low probability). If more than two are predicted, only the two with the highest scores are allowed in the submitted results.

5.2 Experimental Setup

We will focus on three different approaches based on our previous mentioned feature groups, namely choosing based on feature importance the following three features; fine-tuned BERT similarity, BM-25 ranking between entailing fragment f and p_i and the NLI probability. The second one being all feature groups followed by XGBoost with simple hyperparameter tuning and lastly all feature groups followed by XGboost but with Bayesian optimization. We cross validate the training set with five folds, and our results can be found in Table 1. It is evident that we try to optimize precision over recall for more precise results.

Approach	Precision	Recall	F1
Feature Importance	0.5990	0.5800	0.5855
XGBoost	0.6168	0.5855	0.5984
XGBoost Bayesian	0.7054	0.4785	0.5674

Table 1. Validation results of the models for task 2 using 5-fold cross validation.

5.3 Results

Applying the previous mentioned models on the official COLIEE 2020 test set, we get the results as described in Table 2. All of the models achieve F1 scores high enough for the top seven submissions this year. Using Bayesian tuning, we obtain the best results from our own submissions and obtain third place, showing the importance of proper tuning.

Team	F1
JNLP BMWT	0.6753
JNLP BMW	0.6222
TAXI XGBoost Bayesian	0.6180
TLIR	0.6154
JNLP WT	0.6094
TAXI XGBoost	0.5992
TAXI Feature Importance	0.5917

Table 2. The top seven F1 scores of COLIEE 2020 task 2.

Analysis Upon further analysis of the features, we can see that using the most important features is necessary, but not sufficient for state-of-the-art results. Furthermore, grid search hyperparameter tuning improves performance, but not as much as Bayesian optimization methods.

6 Task 3: Statute Law Information Retrieval

The statue law information retrieval task focuses on finding relevant articles $S_1, S_2, ..S_N$ from the complete Japanese Civil Code to a legal bar exam question Q, such that the use of these articles in combination with the exam question would yield entailment or not. We use the English version of this data, which is the translated version of the original Japanese dataset.

6.1 Methodology

We will use three different approaches to generate a feature embedding for each bar exam question and all articles. These are then compared and ranked using cosine similarity. We then always return the most similar article, and any additional ones based on the difference δ between the previously retrieved article and the next most similar article. If that difference is below a certain threshold H, we continue to return more articles until either the threshold is exceeded or a maximum amount of retrieved articles R is found. To evaluate our models, we use the macro-average of precision, recall and the F-2 measure, where the latter is chosen due to the higher importance of recall.

Approach 1 - TF-IDF Vectors This feature approach, which is based on last year's winner [15], is TF-IDF with IDF weight smoothing. We generate TF-IDF vectors for each article and bar exam question, containing the relative frequency of each word in a given vocabulary.

Approach 2 - Legal Embeddings The other approach is based on our legal embeddings, discussed in Section 3. Embeddings have been proven to capture

context, whereas a simple word approach as TF-IDF does not, hence the choice of using embeddings as well. For all the articles and bar exam questions, we generate a legal question/article embedding by finding the legal embeddings for its words and take the average for the final representation.

Approach 3 - Combination Given that the earlier mentioned approaches have their own shortcomings as well as strengths, we propose another approach to combine the best of both. We use both approaches separately and combine their top 100 retrieved articles and re-evaluate the order with the same ranking mechanism as explained before.

6.2 Experimental Setup

Since each of these approaches has parameters to tune, we use a grid search method to find the best parameters via the hold out validation set. This yields for the TF-IDF implementation a maximum amount of 3 retrieved articles and a $\delta = 0.007$ which results in a larger recall over precision. For the legal embedding approach we set a maximum amount of 4 retrieved articles and $\delta = 0.007$. Lastly, for the combined model, the maximum amount of retrieved articles is 4 and $\delta = 0.015$. Moreover, the embeddings use lowercasing, punctuation removal, stopword removal and number to text conversion. For the TF-IDF, we keep casing, do not remove punctuation, use number to text conversion, keep stopwords and use both uni- and bigrams.

We split our training data into a train and validation set to obtain intermediate results as well as having interpretable numbers during hyper-tuning and use 600 examples for training and 96 for validation. These results are shown in Table 3, where we can observe that recall is indeed larger. Critically, the combination of the two features show a more balanced precision and a larger recall.

Approach	Precision	Recall	F2
TF-IDF	0.5130	0.5217	0.5117
LE	0.3823	0.5382	0.4490
Combination	0.4790	0.5660	0.5161

Table 3. Validation results of the models for task 3, where LE stands for the legal embedding approach.

6.3 Results

When evaluating our models against other COLIEE 2020 participants on the test set, we get the results mentioned in Table 4. Our legal embeddings on their own do not cover enough semantics and context to obtain sufficient performance; however, in combination with TF-IDF they do boost the recall significantly, showing that they have promising prospects.

Team	Precision	Recall	F2	MAP	R@5	R@10	R@30
LLNTU	0.6875	0.6622	0.6587	0.7604	0.8071	0.8571	0.9214
JNLP.tfidf-bert-ensemble	0.5766	0.5670	0.5532	0.6618	0.6857	0.7143	0.7786
cyber1	0.5058	0.5536	0.5290	0.5540	0.5500	0.6929	0.8000
TAXI_R3	0.4393	0.5089	0.4546	0.5057	0.5714	0.6143	0.6786
TAXI_R1	0.4435	0.4152	0.4112	0.4883	0.5857	0.6214	0.7214
TAXI_R2	0.2872	0.4182	0.3400	0.3741	0.3786	0.4214	0.5643

Table 4. Quantitative results on task 3 by the top-3 participants on COLIEE 2020 and our three submissions. R1, R2 and R3 stand for the TF-IDF, Legal Embedding and combination approach respectively.

Analysis

Training Statistics When we evaluate the cosine similarities values between the bar exam questions and all possible articles, we can see a clear difference in their similarity values, which is shown in Table 5. The TF-IDF similarity values indicate a large difference between relevant and non-relevant articles compared to a bar exam question. However, there is a large standard deviation among relevant articles, indicating that some relevant articles to bar exam questions are not found by TF-IDF which are found by our legal embeddings.

Approach	Relevant	Non-relevant
TF-IDF	0.1651 ± 0.1629	0.0092 ± 0.0184
LE	0.9264 ± 0.0436	0.8754 ± 0.0437

Table 5. Mean cosine similarity value $\pm$ their standard deviation for relevant and non-relevant articles to the bar exam questions on the training data with both feature methods.

Strengths & Shortcomings When we manually evaluate the performance of both the TF-IDF and legal embeddings on a few test examples, we can see a clear difference in their strengths and shortcomings. TF-IDF tends to work quite well on finding relevant articles to bar exam questions when the vocabulary used is the same. However, our legal embeddings also find the correct relevant articles to a bar exam question even without shared vocabulary between them. For example, it has learned that a *higher statutory agent* can also be a *legal representative*, which gives us a good indication that these embeddings are essential for improved text comparison.

7 Task 4: Statute Law Entailment

The statue law entailment task requires determining whether a legal bar exam question, Q, is entailed in the text of a set of articles, $S_1, S_2, ...S_N$ relevant to Q. Entailment is taken to mean whether Q is true or false given the content of $S_1, S_2, ...S_N$. This was done in two ways: using a BERT-XGBoost combination and using the legal embeddings with a Bi-GRU. The former is inspired by the previous 2019 COLIEE submission of Gain and colleagues (2019) for the purposes of bench-marking the latter.

7.1 Methodology

BERT-XGBoost Combination For each example, the set of articles S were concatenated to each other, then to the question, Q, with a separator token, and summarized with Gensim if necessary. The last hidden layers of the BERT base cased model were then input to XGBoost.

Legal Embeddings Bi-GRU For each example, the set of articles S were concatenated to each other and then to the question, Q, with a separator token. Stop words were not removed as they contain important information regarding negation and affirmation. The tokenized and padded data was then fed to the Bi-GRU.

7.2 Experimental Setup

XGBoost, was then hyperparameter tuned using five-fold cross validation and Bayesian optimisation methods. The hyperparameters tuned were the following: subsample; the amount of training data to be used per sample, max depth; the maximum tree depth, eta; the learning rate, colsample by level; the subsample ratio per tree used when making a level, and colsample by tree; the subsample ratio per column used when making a tree. Resulting values are .60, 4, .50, .34, .98, respectively. The tuned model was evaluated on a hold out validation set of 20% of the data yielding an average precision, recall and F1 of .63.

Initial experiments found that the Bi-GRU-ATT used in previous legal prediction tasks [3] has too many parameters for such a small dataset. Therefore, we used only one GRU layer and removed the attention. Due to time constraints the model was not fully hyperparameter tuned. The resulting precision, recall, accuracy and F1 on the test set were .59, .58, .58, .56, respectively.

7.3 Results

The results of our two submissions, taxi_BERTXGB and taxi_le_brigru, can be found in Table 6 alongside the top three wining submissions. The former answered 60 questions correct with an accuracy of 0.5357 and the latter answered 57 questions correct with an accuracy of 0.5089.

Model	Number Correct	Accuracy
JNLP.BERTLaw	81	0.7232
TRC3mt	70	0.6250
TRC3t5	70	0.6250
taxi_BERTXGB	60	0.5357
taxi_le_bigru	57	0.5089

Table 6. The top three performing models of COLIEE 2020 task 4 along with our two submissions.

Analysis By using TF-IDF cosine similarity as a measure of shared vocabulary we can elucidate both models' strengths and shortcomings.

The necessity of shared vocabulary for the BERTXGB implementation is evident in the similarity difference between correctly and incorrectly predicted labels, 0.4252 and 0.3429, respectively. However, BERTXGB was not competitive with the top performing submissions. This is likely due to the fact that BERT cased has not been trained on legal text.

The shortcomings of the Bi-GRU model are less explicit, as the TF-IDF cosine similarity is actually higher in the incorrectly predicted labels than in the correctly predicted labels. Indeed, the Bi-GRU implementation was not much better than random chance.

8 Conclusion

In this paper, we propose several approaches to both legal information retrieval and entailment. For task 1, the case law retrieval task, we use CatBoost with both absolute word counts and TF-IDF features obtaining a F1 score of 0.04. The second task, the case law entailment task, makes use of Information Retrieval features such as BM-25, NLI probabilities and fine-tuned embeddings. This model achieves a F1 score of 0.6180, while also achieving third place in the competition. The third task, the statutory information retrieval task, utilizes pre-trained legal embeddings in combination with TF-IDF obtaining a F2 of 0.4546. Lastly, task 4, the statutory entailment task, achieves an accuracy of 0.5357 with BERT embeddings into XGBoost.

The use of extra linguistic features, such as NLI, have been shown to be important and useful to obtain better and state-of-the-art performance. Moreover, we showed that even though the current legal embeddings are not state-of-the-art, they do indicate a better understanding of legal terms that is necessary for obtaining better performance. Future research will involve re-training the legal embeddings using a contextually sensitive model such as BERT, for a deeper understanding of legal nuance.

References

1. Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A.: Law and the semantic web: legal ontologies, methodologies, legal information retrieval, and applications, vol. 3369. Springer (2005)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
3. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. arXiv preprint arXiv:1906.02059 (2019)
4. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Extreme multi-label legal text classification: a case study in eu legislation. arXiv preprint arXiv:1905.10892 (2019)
5. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y.: Xgboost: extreme gradient boosting. R package version 0.4-2 pp. 1–4 (2015)
6. Clement, J.: Worldwide desktop market share of leading search engines from january 2010 to april 2019. Retrieved March **12**, 2019 (2019)
7. Dale, R.: Law and word order: Nlp in legal tech. Natural Language Engineering **25**(1), 211–217 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Dorogush, A.V., Ershov, V., Gulin, A.: Catboost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363 (2018)
10. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: Iitp in coliee@ icail 2019: Legal information retrieval using bm25 and bert (2019)
11. Holzenberger, N., Blair-Stanek, A., Van Durme, B.: A dataset for statutory reasoning in tax law entailment and question answering. arXiv preprint arXiv:2005.05257 (2020)
12. Introna, L.D., Nissenbaum, H.: Shaping the web: Why the politics of search engines matters. The information society **16**(3), 169–185 (2000)
13. Kerikmäe, T., Hoffmann, T., Chochia, A.: Legal technology for law firms: determining roadmaps for innovation. Croatian International Relations Review **24**(81), 91–112 (2018)
14. Kim, M.Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 283–289 (2019)
15. Kim, M.Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. p. 283–289. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3322640.3326742, https://doi.org/10.1145/3322640.3326742
16. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
18. Manning, C.D., Schütze, H., Raghavan, P.: Introduction to information retrieval. Cambridge university press (2008)

19. Mitra, B., Craswell, N.: Neural models for information retrieval. arXiv preprint arXiv:1705.01509 (2017)
20. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features (2017)
21. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Special Publication Sp **109**, 109 (1995)
22. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiskỳ, T., Blunsom, P.: Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664 (2015)
23. Rossi, J., Kanoulas, E.: Legal information retrieval with generalized language models. Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019)
24. Saha, P., Mathew, B., Goyal, P., Mukherjee, A.: Hateminers: detecting hate speech against women. arXiv preprint arXiv:1812.06700 (2018)
25. Salehi, S., Du, J.T., Ashman, H.: Use of web search engines and personalisation in information searching for educational purposes. Information Research: An International Electronic Journal **23**(2), n2 (2018)
26. Sammons, M., Vydiswaran, V.V., Roth, D.: Ask not what textual entailment can do for you... In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1199–1208 (2010)
27. Shelmanov, A., Pisarevskaya, D., Chistova, E., Toldova, S., Kobozeva, M., Smirnov, I.: Towards the data-driven system for rhetorical parsing of russian texts. In: Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019. pp. 82–87 (2019)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Wehnert, S., Hoque, S.A., Fenske, W., Saake, G.: Threshold-based retrieval and textual entailment detection on legal bar exam questions. arXiv preprint arXiv:1905.13350 (2019)
30. Wydick, R.: Plain english for lawyers: Teacher's manual (2005)
31. Yoshioka, M., Song, Z.: Hukb at coliee 2019 information retrieval task - utilization of metadata for relevant case retrieval. Proceedings of the 6th Competition on Legal Information Extraction/Entailment. (2019)
32. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., Sun, M.: How does nlp benefit legal system: A summary of legal artificial intelligence. arXiv preprint arXiv:2004.12158 (2020)
33. Zientek, L.R., Werner, J.M., Campuzano, M.V., Nimon, K.: The use of google scholar for research and research dissemination. New Horizons in Adult Education and Human Resource Development **30**(1), 39–46 (2018)

Using BERT and TF-IDF to Predict Entailment in Law-Based Queries

Arman Aydemir[1][0000-0002-7072-6216], Pedro de Castro Souza[2][0000-0003-1281-7125], and Andrew Gelfman[1][0000-0001-5450-6988]

[1] University of Colorado, Boulder, CO 80309, USA
arman.aydemir@colorado.edu
[2] Chemin Eugène-Rigot 2A, 1202 Genève, Switzerland

Abstract. The Competition on Legal Information Extraction/Entailment, COLIEE for short, is an annual 4-task contest to create a machine learning model that can answer law-based questions. Previous applicants have used a variety of methods to solve the issues presented. Among them are term frequency - inverse document frequency, recurrent neural networks, and structural analysis. Our paper will discuss our attempts to solve two of these tasks. Specifically, we address our approach of Task 3 with BERT and Task 4 with a combination of TF-IDF and BERT. We believe that, while not currently the best result, we created multiple worthy contenders in both tasks with a clear path to improvement for next year.

Keywords: Legal Documents Processing · Textual Entailment · Information Retrieval · Classification · Question Answering · BERT

1 Background

The challenges we address in this paper come from COLIEE 2020. The Competition on Legal Information Extraction/Entailment (COLIEE) is a series of evaluation campaigns to discuss the state-of-the-art information retrieval and entailment of legal texts. It is an annual event run in association with the International Conference on Artificial Intelligence and Law (ICAIL). We believe the legal domain is an excellent opportunity to apply Natural Language Processing and Information Extraction and to advance the state-of-the-art. We also think the solutions to these challenges could have a significant application both in helping professional lawyers and in assisting everyday people in understanding the laws they live by. There are two datasets and four tasks that are part of the competition. The first dataset and the first two tasks are related to the Federal Court of Canada case laws. The second dataset and third and fourth tasks are related to a dataset of the Japanese legal bar exam questions, along with the Japanese Civil Code. We addressed the second of these datasets and created solutions to Task 3 and Task 4.

Task 3 involves reading a legal bar exam question Q and extracting a subset of Japanese Civil Code Articles S1, S2,..., Sn from the entire Civil Code which are those appropriate for answering the question such that Entails(S1, S2, ..., Sn, Q) or Entails(S1,

S2, ..., Sn, not Q). Given a question Q and the entire Civil Code Articles, the solution must retrieve the set of "S1, S2, ..., Sn" as the answer to this task.

We used two approaches to solve Task 3. Our first and most basic model for Task 3 is based on Term Frequency and Inverse Document Frequency (TF-IDF) [1], using it to rank articles then applying a heuristic to decide how many articles to return. TF-IDF is a score of how important a term is. Its value increases proportionally to the number of times the word appears in the document and then it is offset by the number of documents the term is used in. Our second model used a BERT [2] sequence classifier to classify query and article pairs as either relevant or not relevant categories.

Task 4, on the other hand, consists of the identification of an entailment relationship such that Entails(S1, S2, ..., Sn, Q) or Entails(S1, S2, ..., Sn, not Q). Given a question Q and relevant articles S1, S2, ..., Sn, determine if the related articles entail "Q" or "not Q."

We used only one approach for Task 4, however, trained and evaluated on slightly different data. The singular approach we used for Task 4 is similar to the BERT approach for Task 3, in that it used a sequence classification model to classify a question and relevant article text pair as either Yes or No. For our first approach, our model only considered the given relevant articles, while in the second approach we also added articles deemed relevant by our TF-IDF model.

2 Methods and Implementation

The first step in our implementation was finding a way to split the raw civil code text into the appropriate list of articles. As a first attempt to do this, we used the same code used by the IITP paper [3], which we found on GitHub. This code was our starting point, but quickly realized significant bugs were causing it not to return the articles correctly. We fixed the code and then went through each article by hand to make sure that articles were delineated correctly. We did our best to make sure that the 'titles' of the articles in parenthesis were also included as part of the text. Further, we made our updated code available on GitHub [4].

We formed a six-fold cross-validation set for hyperparameter selection from the training data we had before the competition. We mainly tested for learning rates, which pre-trained BERT model to fine-tune (BERT-large, BERT-cased, multilingual BERT), early stopping criteria, sequence limit, etc. These sets were also the main factor in deciding which models we would submit.

2.1 Task 3: Statute Law Information Retrieval

We used two approaches to solve Task 3. Our first and most basic model for Task 3 is CUTFIDF, based on papers from IITP and UA [5]. We received a cosine similarity ranking for each of the articles in our set using TF-IDF. The formula for finding this score for some term t and document d is found here:

$$tf - idf(t, d) = tf(t, d) * idf(t) \qquad (1)$$

Using our hyperparameter validation sets we settled on a heuristic on how many articles to return. This model returned every article with a similarity score within 95% of the top score with a hard maximum of a total of five articles returned.

The second approach had not been attempted in previous editions of this task. The approach was to use BERT sequence classifiers to classify query and article pairs as either relevant or not relevant. This means for each query we performed over 700 classification evaluations since one was needed for each article in our set. We, again, used our hyperparameter validation datasets to settle on our exact specifications.

We evaluated three different BERT models:

- the multilingual BERT model, referred to as 'multi,'
- the BERT-large cased model, referred to as 'cased,' and
- the BERT-large uncased model, referred to as 'uncased.'

The second submission, CUBERT1, used the multilingual BERT model. The third submission, CUBERT2, used the BERT-large cased model. These two were chosen because they yielded the best two average F2 scores from the datasets. The competition defines precision, recall, and F2 measure as macro averages, with exact equations here:

$$precision = \text{average of}\left(\frac{Number\ of\ Correctly\ \ Retrieved\ Articles\ for\ Each\ Query}{Number\ of\ Retrieved\ Articles\ for\ Each\ Query)}\right) \quad (2)$$

$$recall = \text{average of}\left(\frac{Number\ of\ Correctly\ Retrieved\ Articles\ for\ Each\ Query}{Number\ of\ Relevant\ Articles\ for\ Each\ Query}\right) \quad (3)$$

$$F2\text{measure} = \frac{(5 \times precision \times recall)}{(4 \times precision + recall)} \quad (4)$$

For each of our three models, Figure 1 shows the average precision, recall, and F2 scores, respectively. We decided on our two final models because they got the highest two average F2 scores.

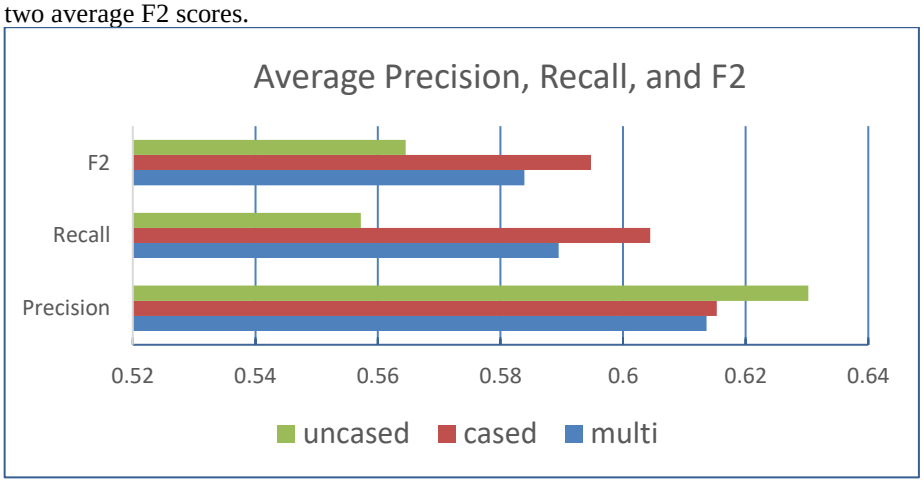


Figure 1. Averages of precision, recall, and F2 scores for each of our three models over our hyperparameter validation sets.

2.2 Task 4: Statute Law Entailment

For Task 4, we only had one approach. This approach was based on BERT and included two submissions. We used our hyperparameter validation datasets to select which BERT models to use again, but this time we also used it to select what data we would be feeding into the models. We found that for all datasets that we attempted the multilingual BERT model performed best. For the first attempt, we simply fed given related articles, this model is referred to as 'given.' In another, we fed it the whole civil code text, referred to as 'full'. In the third, we fed it the given related articles plus extra that were returned using our earlier CUTFIDF model for Task 3 and referred to it as 'plus.' We can see from Figure 2 that our 'given' model yielded the best accuracy on one half of the hyperparameter validation datasets while the 'plus' model achieved the best accuracy on the other half. Because of these scores, these were the two models selected for submission.

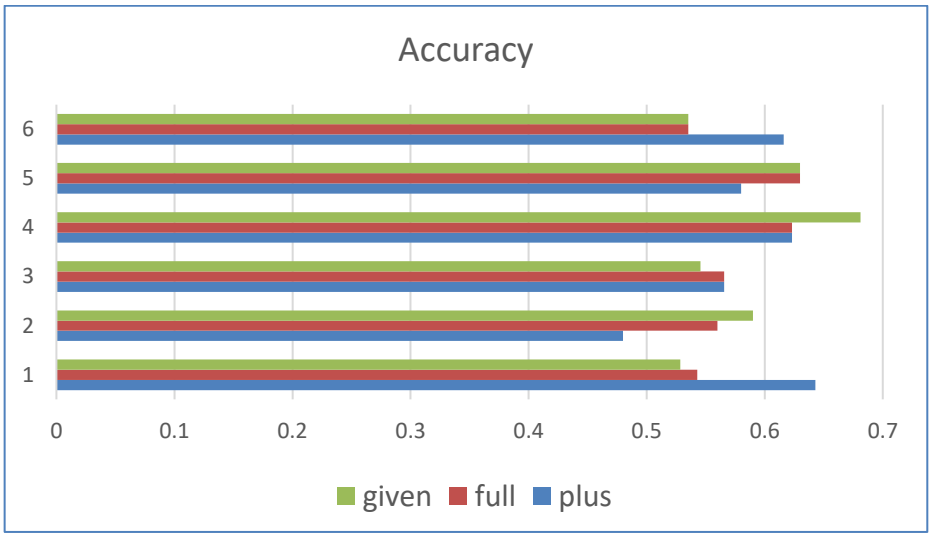


Figure 2. Accuracy score for each of our six hyperparameter validation datasets for our three models.

3 Results

3.1 Task 3: Statute Law Information Retrieval

Our three submissions for Task 3 were not prize-winning but we still learned a good amount from these results. The submissions were ranked and evaluated based on three main criteria – precision, recall, and F2 – which were all defined earlier. In addition, we submitted our top 100 results to create the MAP, R_5, R_10, and R_30 measures. MAP stands for Mean Average Precision, which is defined as the mean of the average

precision for each classification. R_k is defined as recall using the top k rankings of documents. It is noticeable that our TF-IDF method did slightly better than the UA.tfidf method which it was based on. This aligns with our previous work and knowledge. This being true, it still did remarkably worse than both of our BERT models and was second to last in terms of F2 score. This is a good sign that our BERT model has a higher level of understanding than the simple TF-IDF method. Between our two BERT models, CUBERT1 (based on multilingual BERT), did better in every single category compared to CUBERT2 (based on BERT-large cased), except for in the very last category of R_30. While our best model (CUBERT1) was not very close to prize-winning in terms of F2 (9th place), it did fairly well in terms of precision (4th place). We presume our multilingual model performed better because of the many foreign language influenced terms common in legal text. It seems that some other BERT approaches also did very well this year. It will be interesting to compare their approach with ours and see how they achieved such good results.

Table 1. Task 3 competition results (our methods highlighted in yellow)

Submission Id	F2	Precision	Recall	MAP	R_5	R_10	R_30
LLNTU	0.659	0.688	0.662	0.760	0.807	0.857	0.921
JNLP.tfidf-bert-ensemble	0.553	0.577	0.567	0.662	0.686	0.714	0.779
cyber1	0.529	0.506	0.554	0.554	0.550	0.693	0.800
cyber2	0.525	0.435	0.597	0.554	0.550	0.693	0.800
JNLP.tfidf-lawbert	0.518	0.539	0.527	0.685	0.714	0.771	0.793
cyber3	0.518	0.413	0.624	0.554	0.550	0.693	0.800
HUKB-1	0.516	0.420	0.591	0.569	0.671	0.721	0.814
JNLP.tfidf-bert	0.515	0.541	0.527	0.685	0.714	0.771	0.793
CUBERT1	0.514	0.540	0.519	0.585	0.643	0.686	0.707
HUKB-2	0.514	0.404	0.591	0.574	0.643	0.693	0.793
TRC3_1	0.501	0.456	0.536	0.598	0.693	0.771	0.843
CUBERT2	0.500	0.516	0.510	0.556	0.579	0.650	0.757
TRC3_A	0.491	0.415	0.534	0.554	0.607	0.700	0.843
TRC3_H	0.479	0.431	0.522	0.585	0.707	0.786	0.850
OVGU_bm25	0.477	0.400	0.534	0.510	0.593	0.614	0.721
OvGU_combined_bm25	0.470	0.393	0.515	0.548	0.586	0.657	0.700
TAXI_R3	0.455	0.439	0.509	0.506	0.571	0.614	0.679
OVGU_tfidf	0.439	0.344	0.491	0.515	0.571	0.621	0.750
GK_NLP	0.427	0.286	0.499	0.498	0.557	0.636	0.721

Table 2. Task 3 competition result (Continued) (our methods highlighted in yellow)

Submission Id	F2	Precision	Recall	MAP	R_5	R_10	R_30
HUKB-3	0.414	0.438	0.411	0.544	0.629	0.721	0.829
TAXI_R1	0.411	0.444	0.415	0.488	0.586	0.621	0.721
CUTFIDF	0.407	0.445	0.405	0.487	0.550	0.593	0.693
UA.tfidf	0.391	0.429	0.387	0.478	0.543	0.607	0.671

3.2 Task 4: Statute Law Entailment

Our competition results from Task 4 were not nearly as impressive as they were for Task 3 and did not give us much insight into our approaches. Both our CUGIVEN and CUPLUS models got the same accuracy which was worse than the baseline. Here there is opportunity to make some improvements. One approach could be using our BERT results from Task 3 which were clearly superior instead of our TF-IDF results. We could also attempt some type of ensemble with different data fed into each model. We will again be looking forward to learning more about how the other groups decided to approach this task.

Table 3. Task 4 competition results (Our methods listed in yellow)

Submission Id	Correct	Accuracy
Base Line	Yes 59/All 112	0.5268
JNLP.BERTLaw	81	0.7232
TRC3mt	70	0.625
TRC3t5	70	0.625
UA_attention_final	70	0.625
UA_roberta_final	70	0.625
KIS_2	69	0.6161
llntu	69	0.6161
cyber	69	0.6161
UA_structure	68	0.6071
GK_NLP	63	0.5625
linearsvm.HONto	63	0.5625
JNLP.BERT	63	0.5625
KIS	63	0.5625
linearsvm_no_ngram.HONto	62	0.5536
JNLP.TfidfBERT	62	0.5536
KIS_3	61	0.5446

Table 4. Task 4 competition results (Continued) (Our methods listed in yellow)

Submission Id	Correct	Accuracy
sim_neg.OvGU	61	0.5446
UEC1	61	0.5446
taxi_BERTXGB	60	0.5357
UECplus	60	0.5357
CUGIVEN	58	0.5179
CUPLUS	58	0.5179
linearsvm_no_ngram_nofuzz.HONto	57	0.5089
POS_simneg.OvGU	57	0.5089
taxi_le_bigru	57	0.5089
TRC3A	56	0.5
UEC2	55	0.4911
baseline_attention.OvGU	54	0.4821
AUT99-BERT-MatchPyramid	52	0.4643
AUT99-LSTM-CNN-Attention	50	0.4464

References

1. K. S. Jones, A statistical interpretation of term specificity and its application in retrieval. In: Willett, P. (ed.) Document Retrieval Systems, pp. 132-142. Taylor Graham Publishing, London, UK, UK, 1988.
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2nd version, arXiv.org Abs:1810.04805, http://arxiv.org/abs/1810.04805, (2019).
3. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: IITP in COLIEE@ICAIL 2019: Legal Information Retrieval using BM25 and BERT. (2019)
4. CU_COLIEE_2020, https://github.com/armanaydemir/CU_COLIEE_2020, 2020/10/12
5. Mi-Young, K., Rabelo, J., Goebel, R.: Statute Law Information Retrieval and Entailment. In: Proceedings of the 17th International Conference on Artificial Intelligence and Law. ICAIL'2019 (2019)
6. Kano, Y., Kim, M., Lu, Y., Rabelo, J., Kiyota, N., Goebel, R., Satoh, K.: COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment. JSAI International Symposium on Artificial Intelligence. New Frontiers in Artificial Intelligence. JSAI-isAI 2018. Springer, Cham (2018).

Legal Bar Exam Solver integrating Legal Logic Language PROLEG and Argument Structure Analysis with Legal Linguistic Dictionary

Ryuji Hayashi, Naoki Kiyota, Masaki Fujita, and Yoshinobu Kano

Shizuoka University, Johoku, 3-5-1, Naka-ku, Hamamatsu-shi, Shizuoka, 432-8011, Japan
rhayashi@kanolab.net, nkiyota@kanolab.net, mfujita@kanolab.net, kano@inf.shizuoka.ac.jp

Abstract. We aim to build a legal document processing system which integrates logics and NLP in a white box manner, rather than to use end-to-end machine learning methods. We used PROLEG, a logic-type language for legal reasoning, and predicate argument structure analysis for our system. We built a range of Japanese legal dictionaries for predicate argument structures and paraphrases, which can integrate PROLEG with our NLP system. We applied our system to the Japanese legal bar examination in the Task 4 of the COLIEE 2020 shared task as team KIS, in which formal run results for our best run submission was ranked as the sixth-best (accuracy 0.6161) among 32 submitted runs. The experimental results showed that our PROLEG module was able to make partially correct factual judgments, while not able to fully solve the problems. Because this is our first and preliminary study with our small set of dictionaries, we plan to extend our dictionary and our system to cover broader lexicon

Keywords: COLIEE, Question Answering, Legal Bar Exam, Legal Information Extraction, Predicate Argument Structure Analysis, PROLEG..

1 Introduction

Objective reasoning is required to derive conclusions in the legal domain. For such objective reasoning, legal document processing by natural language processing (NLP) system is critically important. However, the NLP system is required to adapt to legal situations and legal concepts, in addition to complex linguistic structures such as entailments, syntax, semantics, semantic roles, paraphrasing, personal relationships, logical structures. The system is also required to show the process of legal reasoning for human users. Therefore, black-boxed end-to-end machine learning is not suitable for the legal domain.

Mi-Young, et al. [1] achieved high accuracy by defining hand crafted rules for conditions, conclusions and exceptions to obtain entailments, creating their negative expression dictionary, comparing with article texts. Gain, et al. [2] and Raiyani et al. [3] employed information retrieval and deep learning methods to answer yes/no questions of the Japanese legal bar exam. However, the legal bar examination requires logics

regarding personal relationships and contract information. We need logics and NLP integrated together for objective judgements.

PROLEG expresses legal reasoning structures as formulas. There have been few previous studies focusing on relationships of legal reasoning, attempted to convert into PROLEG [4]. Navas-Loro, et al. [5] created a framework that expresses relationships of legal reasoning of sentences regarding sales contracts. We propose an approach to extract factual relationships from predicate argument structures rather than inferential relationships in a given problem, which aimed to complete an inference by PROLEG.

The COLIEE (Competition on Legal Information Extraction and Entailment) shared task series have been held to provide a challenge to evaluate legal document processing. Previous COLIEE challenges include COLIEE 2014 [6], COLIEE 2015 [7], COLIEE 2016 [8], COLIEE 2017 [9], COLIEE 2018 [10], and COLIEE 2019 [11]. These COLIEE shared tasks were held in association with the JURISIN (Juris-informatics) workshop and ICAIL (International Conference on Artificial Intelligence and Law). In this paper, we describe our challenge for the latest COLIEE 2020. The COLIEE 2020 shared task consists of four tasks. Task 1 and Task 2 use the legal cases of Canadian Federal Court. Given a Japanese legal bar exam question, Task 3 asks participants to extract a relevant subset of Japanese Civil Law articles that are required to answer the given question. In Task 4, participants are given a problem sentence and its related article(s), asked to answer whether the sentence is correct or not. We participated COLIEE 2020 Task 4, applying our proposed system.

We describe our system in Chapter 2, the experimental results in Chapter 3, discussion in Chapter 4, concluding the paper in Chapter 5.

2 System

2.1 System Overview

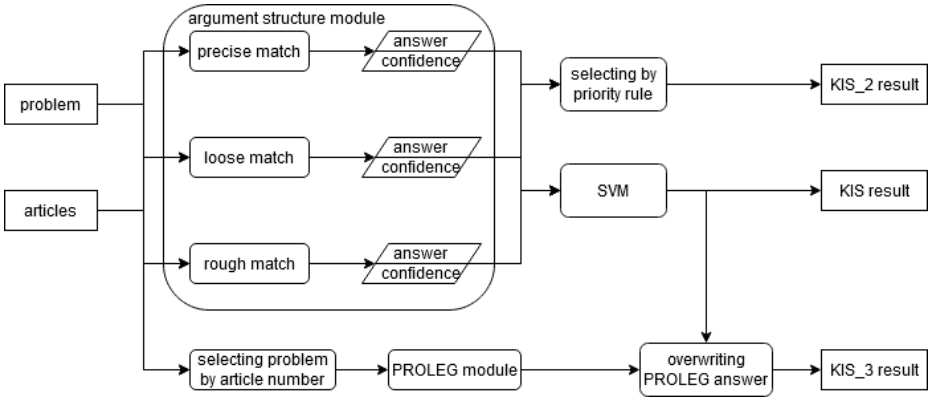


Fig 1. The overall configuration of our solver system

Fig 1 shows a conceptual module design of our system. Roughly speaking, our system consists of two modules: a predicate argument matching module and a PROLEG

module. The predicate argument matching module is same as our previous work in the previous COLIEE shared task. The PROLEG module selects questions of relevant clause numbers, which could be answered by this PROLEG module as we describe later. While there is potentially a large number of combinations of our internal sub-modules, we prepared three combinations as follows, because COLIEE allows to submit three submissions at most. Our team name is KIS. We submitted three results, KIS, KIS_2, andKIS_3. KIS is an output of SVM [13], which is trained to select one of three preceding modules. KIS_2 selects one of the three modules by its priority rules. KIS_3 is based on KIS but overwrites some of the KIS's answers by the PROLEG module.

2.2 PROLEG module

PROLEG

PROLEG extends the PROLOG language [14] for legal reasoning, which uses the presupposed ultimate fact theory ("要件事実論"). The presupposed ultimate fact theory draws a conclusion by its requirements, its exceptions, and their respective factual relationships. A requirement is a condition for a legal conclusion to be valid. If a requirement is valid, then an ancestor conclusion is valid as well. There may be multiple requirements, in which case, if all the requirements are satisfied, the conclusion is satisfied.

PROLEG describes the requirements and rules as follows.

Predicate (Argument 1, Argument 2, ... Argument n)

Fig 2 shows an example. The original predicate names and argument names are in Japanese. Hereafter, we show the original Japanese descriptions with English translations.

管理費用の支払い(_共有者1,_対象物,_金額,_支払時)
'payment_of_management_fees'(_co-owner1,_objects,_amount,_time_of_payment)

Fig 2. An example of a PROLEG predicate.

PROLEG rules consist of *Rule Bases* and *Fact Bases*. A Rule Base is a development of requirements including exceptions. A Fact Base expresses a factual relationship. PROLEG assumes that a problem has a corresponding single final conclusion, thus PROLEG develops requirement rules from that final rule. A Rule Base develops requirements using "<=" symbols. If there are additional requirements for that requirement, such additional requirements are described in the same way using the "<=" symbols. If an exceptional reason exists, PROLEG uses a special predicate "例外事由 (Exceptional Reason)". A Fact Base states facts of expanded terminal requirements. An end PROLEG reasoning can be excuted by applying a special predicate "請求権存在 (existence of a claim)" to the top-level rule of the Rule Bases. Fig 3 shows an example of PROLEG rules, where strings after "%" are comments which describe English translation of the Japanese predicate names.

```
s:-請求権存在( 'A'('V','W') ). %"請求権存在" means "existence of a claim"
b:-block( 'A'('V','W') ).

%Rule Base
'A'(V,W) <=
   'B'(V,W),
   'C'(V,W).
'C'(V,W) <=
   'D'(X,Y,Z).
  例外事由( 'C'(V,W), 'E'(V,W) ).  %"例外事由" means "Exceptional reasons"
'E'(V,W) <=
   'F'(V,W).
例外事由( 'E'(V,W), 'G'(V,W) ).
'G'(V,W) <=
   'H'(V,W).

%Fact Base
主証( 'B'('V','W') ). %"主証" means "primary evidence"
主証( 'D'('X','Y','X') ).
主証( 'F'('V','W') ).
要件事実不存在( 'H'('V','W') ). %"要件事実不存在" means "ultimate facts do not exist"
```

Fig 3. An example of PROLEG rules

In the example in Fig 3, requirements B, D, F, and H need to be described in terms of factual relationships. Fact Bases describe them using special predicates of "主証(primary evidence)" and "要件事実不存在 (utlimate facts do not exist)". The "主証(primary evidence)" predicate expresses a "fact" which corresponding requirement is "satisfied (true)" in the Rule Bases. The "要件事実不存在 (ultimate facts do not exist)" predicate expresses "not a fact" which corresponding requirement is "not satisfied (false)" in the Rule Bases.

Fig 4 illustrates PROLEG's reasoning process.

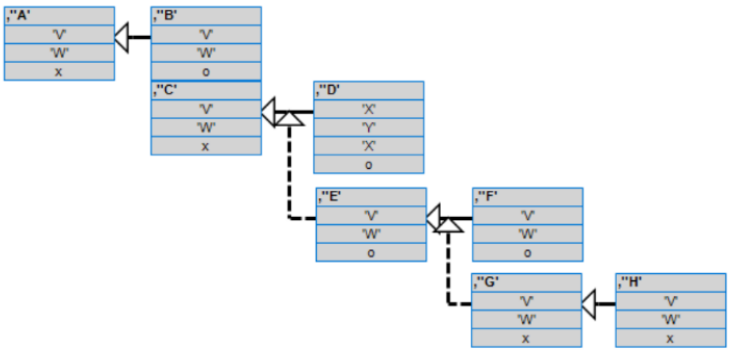


Fig 4. PROLEG's reasoning process

The reasoning results of the requirements propagate as indicated by the arrow symbols. The dashed arrows indicate an exceptional reasoning. In this example, the exception E being true, the result of C becomes false, even though D is true.

PROLEG dataset.
We used 233 PROLEG rules compiled by Satoh, et al, which correspond to some of the past questions of the COLIEE shared task (from H18 to H27) and the entire Japanese civil law articles. The PROLEG rules of the past questions are linked to relevant article numbers of the Japanese Civil Law. We preprocessed these rules to obtain the corresponding PROLEG rules by giving article numbers of the Japanese Civil Law.

PROLEG Module.
Our system performs legal reasonings using PROLEG. Our system develops all possible requirements given relevant articles, then extracts factual relationships of Fact Base rules by comparing with predicate argument structures of a given problem sentence. In contrast to previous studies that select relevant requirements during developments of the requirements, our system greatly relies on the NLP side.

Fig 5 shows our PROLEG module design. Suppose that the PROLEG rules A, B, and C that are tied to it are obtained from the relevant articles to the problem. The nodes at the end of the arrows represent the requirements of the ancestor nodes.

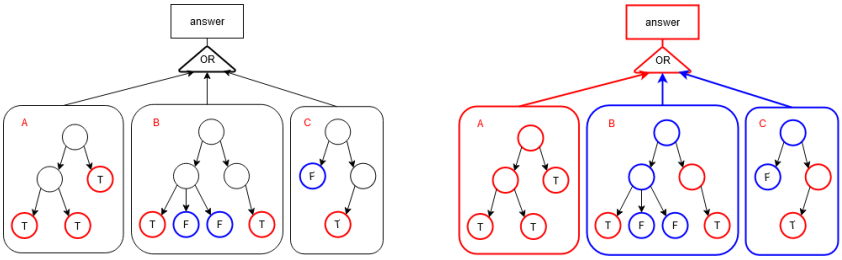


Fig 5. A conceptual diagram of our PROLEG module's design.
(left) leaf nodes are filled by T (True) or F (False)
(right) factual relationships of the terminal nodes are determined

Fig 5 (right) is an example filling the leaf nodes with T (True, red) or F (False, blue). Fig 5 (left) shows the same example with Fig 5 (left), but factual relationships of the terminal nodes are determined, which allows subsequent reasonings. In this example, reasonings of B and C become false, and a reasoning of A becomes true. This means that the rules and requirements of B and C were not suitable for solving this problem. We make a conclusion by taking the logical sum of all of these reasonings. Since A is true in this example, the final conclusion becomes true.

Our system determines whether requirements are relevant or not by focusing on terminal factual relationships of developed requirements, but not during the development of requirements. In other words, our system indirectly takes rules as reasoning evidences as a result.

2.3 Argument Structure Analysis

Fact Base Predicate's Argument Structure.
This section describes PROLEG's arguments and the linguistic predicate argument analysis of the problem texts in the legal bar exam. Our system recognizes a Fact Base's requirement as a fact when it is stated in the given problem text. We explain details based on the examples in Table 1 and Table 2.

Table 1. An example of predicate and arguments (H18-27-I)

売主が，売買目的物の引渡しの提供をした上，相当期間を定めて代金の支払を催告した場合，催告期間の経過後，解除権行使前に，買主から弁済の提供を受けたとしても，売主は，これを拒絶して解除権を行使することができる。
Where the seller has offered delivery of the subject matter of the sale and has demanded payment of the price within a reasonable period of time, the seller may exercise the right of release by refusing to accept the offer of payment from the buyer after the expiration of the period of demand and before the exercise of the right of release, even if the buyer offers to pay the price.

Table 2. An example of Fact Base's predicate

主証(履行の催告(売主,買主,_契約,_催告時)).
主証('Demand_for_Performance' ('Seller', 'Buyer',_Contract,_At_the_time_of_demand)). %"主証" means "primary evidence"

A fact "支払を催告した (demanded payment)" is stated in the problem text of the examples. This fact corresponds to "履行の催告 (demand for performance)" in the Fact Base, so the fact make this requirement "履行の催告 (demand for performance)" to be regarded as a fact. The following two issues need to be resolved to process this relationship: 1) synonyms and 2) structures between the Fact Base predicates in PROLEG and the predicate argument structures in the problem text. We created our hand-crafted thesaurus dictionary for the legal domain to resolve these issues, as described in the next section.

Thesaurus Dictionary.
The word "履行 (performance)" in the previous example's Fact Base refers to the act of executing a claim. On the other hand, the word "支払 (payment)" in the problem refers to the act of executing the claim in the sales contract. In order to obtain the correct answer, it is necessary to understand that these words refer to the same act while they are different lexical words. Likewise, we need to handle synonymous expressions. Therefore, we compiled a legal thesaurus dictionary of synonymous expressions in the Japanese Civil Law. Table 3 shows some of the entries of our dictionary. We plan to make our dictionary available when it covers most of expected terminologies.

Table 3. Some entries of our thesaurus dictionary

hypernym	hyponyms

契約(contract)	双務契約 (bilateral contract), 片務契約 (one-sided contract)
双務契約 (bilateral contract)	売買 (sale), 賃貸借 (lease), 交換 (exchange), 雇用 (employment), 請負 (contracts for work), 組合(partnerships), 和解(settlements)
制限行為能力者 (a person with legal capacity)	未成年者 (minor), 成年被後見人 (adult ward), 被補助人 (a person under assistance), 被保佐人 (person under curatorship)
物的担保 (real security)	約定担保 (stipulated security), 法定担保 (legal security)
約定担保 (stipulated security)	抵当権 (lien), 質権 (pledge)
過失 (negligence)	軽過失 (slight negligence)・ 重過失/重大な過失 (gross negligence)

Because there is no Japanese thesaurus dictionary for legal terms, it was difficult to match legal terms, e.g. the bilateral contract and the sales contract. We extracted relationships between legal terms which are frequently used in the training data, referring to a civil law textbook [14]. Our dictionary includes hypernym-hyponyms relationships based on descriptions in the legal textbooks, forming hierarchical structures. This dictionary allowed us to absorb synonymous differences between the problem text and the PROLEG names.

Factual Determination by PROLEG's Predicate Argument Analysis.
This section describes our method which matches Fact Base predicates with predicate argument structures of problem texts. Then we describe our dictionaries which are used to convert predicate argument structures.

Predicate argument structure analysis for verbal PROLEG predicates.
Since predicates of Fact Base are sometimes verbal, in such cases, we used KNP [15] to obtain a predicate argument structure, and then match with a predicate argument structure in the problem text.

Predicate argument structure analysis for nominative PROLEG predicates using dictionaries
In the previous examples in Table 1 and Table 2, "支払を催告した (demanded payment)" is analyzed as

{predicate : subject : object} = {催告(demand) : null : 支払(payment)}.

In order to match "履行の催告 (demand for performance)" with the predicate argument in the problem sentence, we need to analyze "催告 (demand)" as the predicate, and "履行 (performance)" as the object. Because the original PROLEG predicate names are arbitrary defined by humans, we renamed these predicate names to be a predicate, a

subject, and an object. We created our Japanese predicate argument dictionary which includes 475 entries for the legal domain for this purpose, examples shown in Table 4.

Table 4. Example entries of "no (の)" in our Japanese predicate argument dictionary

#	Name of predicate	predicate	subject	object	denial
1	法定代理人の同意 (consent of legal representative)	同意 (consent)	法定代理人 (legal representative)		
2	追認の催告 (a demand for a supplementary charge)	催告 (notice)		追認 (ratification)	
3	後見監督人の同意不存在 (absence of the supervisor of guardian consent)	同意 (consent)	後見監督人 (supervisor of guardian)		T
4	後見監督人の同意を要する行為 (acts requiring the consent of the supervisor of the guardian)	同意 (consent)	後見監督人 (supervisor of guardian)		

The Japanese case particle "の(no)" can express predicate argument structures implicitly. For example, "法定代理人の同意 (consent of the legal representative)" in #1 of Table 4, "同意 (to consent)" denotes a predicate and "法定代理人 (the legal representative)" denotes a subject, which can be converted into an explicit form like "法定代理人が同意する (the legal representative consents)". On the other hand, in the case of "追認の催告 (demand for recognition)" in #2 of Table 4, "追認(recognition)" is an object of a predicate "催告 (demand)", which can be converted into an explicit form "追認を催告する (to demand for recognition)". In other words, in the form "A's B (AのB)", the noun A and the noun B can be either in a subject-predicate relationship or in a predicate-object relationship. Our dictionary includes such manually determined relationships, correctly converts such different patterns.

In general, if a sentence problem includes a negative expression, we should assume that no requirement fact exists. However, there are some Fact Base predicates which contain a negative meaning. For example, we need to precisely extract a negative meaning from a Fact Base predicate "後見監督人の同意不存在 (absence of guardian's consent)" as the requirement is based on "不存在 (absence)" of consent. We manually assigned a negative expression tag to some of the Fact Base predicates which contain negative expressions.

We regard a requirement is satisfied when either a pair of subjects or a pair of objects is agreed between a PROLEG predicate and a problem sentence, in addition to the negation agreements.

Paraphrase dictionary

As part of our predicate argument dictionary, we added paraphrases which are superficially quite different, using the COLIEE training data, as shown in the Table 5.

Table 5. Examples of paraphrases in our Japanese Predicate Argument Dictionary

#	Predicate name of Fact Base	predicate	sub-ject	object	de-nial
5	欺罔行為 (deception)	行う (commit)		詐欺 (fraud)	
6	本人が行為能力者となった (The person in question has become a person with a capacity to act)	達する (reach)		成年 (the age of majority)	
7	債権者不確知 (unidentified creditor)	確知 (ascertain-ment)		債権者 (creditor)	T

For example, a predicate "欺罔行為 (deception)" of the example #5 of Table 5, we registered an object of a predicate "行う (do)" as "詐欺 (fraud)", because the original training data describe as "欺罔行為 (defrauding)". Similarly, a Fact Base predicate "本人が行為能力者となった (the person himself/herself has become competent to act)" in #6 is synonymous with "成年に達した (has reached the age of majority)" in the problem text. Therefore, we registered "成年(majority)" as the object of the predicate "達した (has reached) ".

When a Fact Base predicate contains a negative meaning, we assigned a negative tag as we described before. Regarding a Fact Base predicate "債権者不確知 (unidentified creditor)" in #7, the corresponding problem text describes as "債権者を確知できない (the creditor cannot be ascertainment)". Therefore, we registered "債権者(the creditor)" as an object of the predicate "確知(ascertainment)".

Omitted Predicates

Some Fact Base predicates have no corresponding predicate in the problem text, which requires our system to somehow determine as "fact". Fig 6 and Fig 7 show examples.

代金の一部だけを支払った段階で目的物についての隠れた瑕疵が明らかになり，損害賠償請求が認められる場合には，買主は，残代金の支払について，損害賠償との同時履行の抗弁を主張することができる。
If a hidden defect in the subject matter becomes apparent after only a portion of the price has been paid, and a claim for damages is recognized, the buyer may assert a defense of concurrent performance with the payment of damages for the remaining price.

Fig 6. An example of predicates omission in problem text (H18-1-5)

主証(合意(売買,売主,買主,目的物,_契約時)). 主証(権利抗弁(同時履行(買主,売主,契約(売買,売主,買主,目的物,_契約時)))). 主証(隠れた瑕疵(契約(売買,売主,買主,目的物,_契約時))).
主証 ('Agreement' ('Sale', 'Seller', 'Buyer', 'Object', _Contract)). 主証('Defense_of_Right'('Simultaneous_Performance'('Buyer', 'Seller', 'Contract' ('Purchase', 'Sale', 'Seller', 'Buyer', 'Object', _Contract))).

```
主証('hidden_defect'('contract'('sale', 'seller', 'buyer', 'object', _contract)).
%"主証" means "primary evidence"
```

Fig 7. An example of predicates omission in Fact Base

Although the above problem in Fig 6 does not describe "合意 (agreement)", we need to regard it as agreed when the sale and purchase are performed, regard this as "fact".

2.4 PROLEG applicable problems

We use our PROLEG module limited to the typical contract problems, which correspond to the Civil Law Articles 521~696. This is because the problem text of typical contracts is easier to obtain the subject-predicate relationships, thus suitable for the factual situation determination by PROLEG.

3 Results and Experiments

3.1 Formal Run Results

Table 6 shows the formal run results of COLIEE 2020 participant teams.

Table 6. the results of each project

Team	Correct	Accuracy
JNLP.BERTLaw	81	0.7232
TRC3mt	70	0.6250
TRC3t5	70	0.6250
UA_attention_final	70	0.6250
UA_roberta_final	70	0.6250
KIS_2	**69**	**0.6161**
(omitted 6 teams)		
KIS	**63**	**0.5625**
(omitted 2 teams)		
KIS_3	**61**	**0.5446**
(omitted 4 teams)		
BaseLine	**59**	**0.5268**
(omitted 10 teams)		

3.2 Experiments on Training Dataset

We applied our system to the training data. Table 7 shows evaluation results on training data for each of our module.

Table 7. The number of correct answers and accuracy for each of our systems

Year of Training Dataset	# of questions	KIS (accuracy)	KIS_2 (accuracy)	KIS_3 (accuracy)
H18	36	**21 (.58)**	**20 (.56)**	**21 (.58)**
H19	37	**21 (.57)**	**18 (.49)**	**20 (.54)**
H20	41	**24 (.58)**	**26 (.63)**	**26 (.63)**
H21	54	**29 (.54)**	**36 (.67)**	**29 (.54)**
H22	47	**29 (.62)**	**27 (.57)**	**28 (.60)**
H23	41	**21 (.51)**	**27 (.66)**	**24 (.59)**
H24	79	**51 (.65)**	**49 (.57)**	**48 (.61)**
H25	60	**44 (.73)**	**33 (.55)**	**40 (.67)**
H26	74	**40 (.54)**	**42 (.57)**	**39 (.53)**
H27	50	**27 (.54)**	**26 (.52)**	**30 (.60)**
H28	49	**35 (.71)**	**30 (.61)**	**31 (.63)**
H29	58	**35 (.60)**	**31 (.53)**	**35 (.60)**
H30	70	**40 (.57)**	**42 (.60)**	**40 (.57)**
Total	696	**417 (.60)**	**403 (.58)**	**411 (.59)**

4 Discussion

We discuss the contribution of our PROLEG module in this section. The following tables show three examples with our analysis, which Fact Base predicates are determined as a fact (true).

Table 8 shows an example where our system applied KNP to the Fact Base predicate, and the pair of predicates is matched between the Fact Base and the problem. Table 9 shows details of our predicate argument structure analysis of this example.

Table 8. An example of a verbal Fact Base predicate that could be analyzed by KNP (H22-27-5)

請負契約に基づき報酬の支払を請求する訴訟においては、請負契約が締結されたこと及び仕事の目的物の引渡しを要するときはこれを引き渡したことを請求原因として主張立証しなければならない。
In a lawsuit demanding the payment of remuneration based on a contract for work, proof must be demonstrated with objects of the claim that the contract for work has been concluded and, if the delivery of the subject matter of work is required, that this has been delivered.

Table 9. Details of predicate argument analysis with verbal Fact Base predicate

	Original Form	predi-cate	sub-ject	object	de-nial
Problem	仕事の目的物の引渡しを要するときは (when the object of the work requires delivery)	要する (requir-ing)		引渡し (deliv-ery)	F
Fact Base	物の引渡しを要する (Requiring delivery of an object)	要する (requir-ing)		引渡し (deliv-ery)	F

Table 10 and Table 11 show an example, which used our dictionary entries of the Japanese particle "no (の)" in Table 4.

Table 10 An example which uses our dictionary entries of the Japanese particle "no (の)" (H26-36-I)

抵当不動産の第三取得者は、抵当不動産について通常の必要費を支出した場合には、果実を取得したときであっても、抵当不動産の代価から、他の債権者より先にその償還を受けることができる。
In cases where a third party acquirer of Mortgaged Immovable Properties has incurred ordinary necessary expenses with respect to the Mortgaged Immovable Properties, even if he/she has acquired fruits he/she shall be entitled to obtain reimbursement of the same out of the proceeds of the Mortgaged Immovable Properties prior to other obligees.

Table 11. Detailed analysis of an example which uses our dictionary entries of the Japanese particle "no (の)"

	Original Form	predicate	subject	object	denial
Problem	果実を取得したときであっても (Even when the fruit is acquired)	取得 (acquired)		果実 (fruit)	F
Fact Base	果実の取得 (Acquiring the fruit)	取得 (acquired)		果実 (fruit)	F

Table 12 and Table 13 show an example which uses the paraphrase dictionary (Table 5).

Table 12. An example which uses our paraphrase dictionary (H20-30-I)

心裡留保の場合、相手方が表意者の真意を知らなかったとしても、知らないことについて重大な過失がなければ、その意思表示は有効である。
In the case of concealment of true intention, assuming the other party was unaware of the true intention of the party making the manifestation, and there was no gross negligence regarding that lack of knowledge, that manifestation of intention shall be valid.

Table 13. Detailed analysis of an example which uses our paraphrase

	Original Form	predi-cate	subject	object	de-nial
Prob-lem	相手方が表意者の 真意を知らなかったとしても (Even if the other side didn't know the true intentions of the person they were dealing with)	知る (know)	相手方 (oppo-nent)	真意 (true meaning)	T
Fact Base	真意ではない (It's not true.)	知る (know)		真意 (true meaning)	T

These analyses show that our PROLEG module could correctly analyzed partial determination of facts. Unfortunately, we could not find any problem in which our PROLEG module could correctly determine all of the Fact Bases developed from the problems and reasonings.

As a future work, we aim to expand our predicate argument structure dictionary and omitted predicate supports to cover a wider range of clauses and problem examples. As we discussed before, there are entailments such as a contract entails an agreement. We would like to support omitted predicates by employing such entailment relationships specific to the legal domain.

5 Conclusion

We aim to build a legal document processing system which integrates logics and NLP in a white box manner, rather than to use end-to-end machine learning methods. We used PROLEG, a logic-type language for legal reasoning, and predicate argument structure analysis for our system. We built a range of Japanese legal dictionaries for predicate argument structures and paraphrases, which can integrate PROLEG with our NLP system. We applied our system to the Task 4 of the COLIEE 2020 shared task, in which formal run results for our system was sixth-best (accuracy 0.6161) among 32 submitted runs. The experimental results showed that our PROLEG module was able to make factual judgments partially as discussed, while not able to fully solve the problems. Because this is our first and preliminary study with our small set of dictionaries, we plan to extend our dictionary and our system to cover broader lexicon.

ACKNOWLEDGMENTS

This research was partially supported by JST CREST and MEXT Kakenhi.
We thank Moeto Yamaguchi for his expert advice on legal issues.

References

1. Kim, M. Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL), pp. 283-289 (2019).
2. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: IITP in COLIEE@ ICAIL 2019: Legal Information Retrieval using BM25 and BERT, In Proceedings of the Competition on Legal Information Retrieval and Entailment Work-shop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law (2019).
3. Raiyani, K., Quaresma, P.: Keyword & machine learning based Japanese statute law retrieval and entailment task at COLIEE-2019. In Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law (2019).
4. Satoh, K., Asai, K., and Hurukawa, T.: PROLEG : PROLEG : Implementing the Presupposed ultimate fact theory of Japanese Civil Code by Logic Programming, In Proceedings of the Knowledge-Based Systems / The Japanese Society for Artificial Intelligence (知識ベースシステム研究会 / 人工知能学会), vol. 92, pp. 1–8 (2011).
5. Navas-Loro, M., Satoh, K., Rodríguez-Doncel, V.: Contractframes: Bridging the gap between natural language and logics in contract law, In Proceedings of the 32nd Japanese Society for Artificial Intelligence (JSAI 2014) International Symposium on Artificial Intelligence, pp. 101-114. Springer, Cham (2018).
6. Competition on Legal Information Extraction/Entailment (COLIEE-14), In Proceedings of the Eighth International Workshop on Juris-informatics (JURISIN 2014), [Online]. Available: http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html.
7. M.-Y. Kim, Goebel, R., Satoh, K.: COLIEE-2015: Evaluation of Legal Question Answering, In Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN 2015).
8. Kim, M., Goebel, R., Kano, Y., Satoh, K.: COLIEE2016: Evaluation of the Competition on Legal Information Extraction and Entailment 2. In Proceedings of the Tenth International Workshop on Juris-informatics (JURISIN 2016) (2016).
9. Kano, Y., Kim, M., Goebel, R., Satoh, K.: Overview of COLIEE 2017, In Proceedings of the Competition on Legal Information Retrieval and Entailment Work-shop (COLIEE 2017) in association with the 16th International Conference on Artificial Intelligence and Law, pp. 1–8 (2017)
10. Yoshioka, M., Kano, Y., Kiyota, N., Satoh, K.: Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018, In Proceedings of the Twelfth International Workshop on Juris-informatics (JURISIN 2018), pp. 1–12 (2018).
11. Rabelo, J., Kim, M. Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: In Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law (2019).
12. Hoshino, R., Kiyota, N., Kano, Y.: Question Answering System for Legal Bar Examination using Predicate Argument Structures focusing on Exceptions, In Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law (2019).
13. Thorsten, J., Text categorization with Support Vector Machines: Learning with many relevant features. In Machine Learning: ECML-98, Nédellec Claire and Rouveirol Céline (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, pp.137–142 (1998).
14. Ito, M.: Introduction to civil law of Makoto Ito. 7th edn. (伊藤真の民法入門[第 7 版]) Nippon Hyoron sha co., Ltd, Tokyo (2020)
15. "日本語構文・格・照応解析システム KNP."(Japanese syntax / case / reference resolution analysis system KNP) [Online]. Available: http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

Information Extraction & Entailment of Common Law & Civil Code

John Hudzina[1], Kanika Madan[1], Dhivya Chinnappa[1], Jinane Harmouche[1], Hiroko Bretz[1], Andrew Vold[1], and Frank Schilder[1]

Center for AI and Cognitive Computing, Thomson Reuters
https://www.thomsonreuters.com/en/artificial-intelligence.html

Abstract. With the recent advancements in machine learning models, we have seen improvements in Natural Language Inference (NLI) tasks, but legal entailment has been challenging, particularly for supervised approaches. In this paper, we evaluate different approaches on handling entailment tasks for small domain-specific data sets provided in the Competition on Legal Information Extraction/Entailment (COLIEE). This year COLIEE had four tasks, which focused on legal information processing and finding textual entailment on legal data. We participated in all the four tasks this year, and evaluated different kinds of approaches, including classification, ranking and transfer learning approaches against the entailment tasks. In some of the tasks, we achieved competitive results when compared to simpler rule-based approaches, which so far have dominated the competition for the last six years.

Keywords: legal AI · information retrieval · textual entailment · natural language processing · multi-task learning · transfer learning

1 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) has run a challenge for extraction and entailment since 2014 that examines the decision process for both case-based and statute-based legal systems. We explored several approaches for information extraction and text entailment related to both common law and civil code. In general, both legal systems provide a rationale for a decision. For common law, judges base legal decisions on past precedents via a process known as *stare decisis*. For civil code, judges base legal decisions on applying one or more statutes to a given situation without altering the central legal issue, i.e., *mutatis mutandis*.

For each legal system, COLIEE lays out a two-step process. The first step *retrieves* the relevant cases or statutes to apply to a decision. Once the system discovers the relevant text, the second steps determine if the relevant text supports or *entails* the decision. Our approach for each step is further described in the following sections:

- **Section 2 - Task 1:** Common Law Retrieval

- **Section 3 - Task 2:** Common Law Entailment
- **Section 4 - Task 3:** Civil Code Retrieval
- **Section 5 - Task 4:** Civil Code Entailment

2 Task 1: The Legal Case Retrieval Task

The goal of Task 1 is to explore and evaluate legal document retrieval technologies that are both effective and reliable. The task investigates the performance of systems that search a set of case laws that support an unseen case law. In response to a query case, the task aims to return *supporting* cases in the given collection. A case is *supporting* to a query case if it supports the decision of the query case. In this task, the query case does not include the decision, because the goal is to determine how accurately a machine can capture decision-supporting cases for a new case (with no decision yet).

The corpus used for training is composed of case laws from the Federal Court of Canada. The training data contains query cases alongside a pool of case laws, and the gold *supporting cases*. Thus for a given a case, the goal is to identify *supporting cases* from a pool of candidate cases. To first find out cases which are similar to the query case and to train our machine learning models, we start with a ranking approach, followed by a classification task. To identify *supporting cases* of a given query case, we first rank all the candidate cases based on their similarity to the base case. Then, depending on the rank a candidate case receives, we build classification models to identify the *supporting cases* from this ranked pool of candidate cases.

The training dataset consists of 520 query cases with 200 candidate cases per query case. The frequency of *supporting cases* against the number of base cases is depicted in Figure 1. As we can see from the figure, there are 97 base cases with 1 *supporting case* and there is 1 base case with 31 *supporting cases*. We conduct experiments to choose the *supporting cases* by pairing each base case with the 200 associated candidate cases. Thus we work with 104,000 instances containing *supporting* and *unsupporting* cases. We observe that there are 2,680 (2.57%) *supporting* pairs and 101,320 *unsupporting* pairs in the dataset.

2.1 Experiments

To identify *supporting* pairs, we begin by ranking the candidate cases based on similarity scores and then conduct classification experiments based on the ranking. Thus our experiments include two tasks.

1. The ranking task
2. The classification task

We pre-processed each case document by removing stop words and punctuation. Each word in the case document is lowercased before undergoing any similarity calculation.

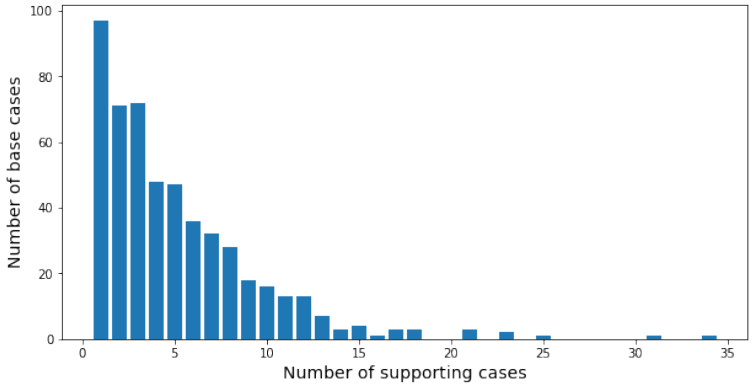


Fig. 1. Frequency of supporting cases vs. number of base cases

Table 1. Similarity scores and their description used in ranking candidate cases for Task 1

	Input	Features
Jaccard score	Entire text	Jaccard similarity
BOW similarity	Entire text	Cosine similarity between BOW vectors
GloVe embeddings similarity	2000 most frequent words	Cosine similarity between average of GloVe embeddings
Word2vec embeddings similarity	2000 most frequent words	Cosine similarity between average of word2vec embeddings

The ranking task: We calculated four similarity scores to rank the candidate cases. We used one or more of these ranking methods to identify the *supporting* cases.

1. **Jaccard similarity**: Jaccard similarity between the base case and each candidate case.
2. **BOW similarity**: Cosine similarity between the bag-of-words vector between the base case and each candidate case.
3. **GloVe embedding similarity**: Cosine similarity between the *average GloVe [7] embedding* between the base text and each candidate case. The *average GloVe embedding* is obtained by averaging the GloVe pretrained embedding of the most frequent 2,000 words from a case text.
4. **Word2Vec embedding similarity** is the cosine similarity between the *average word2vec [3] embedding* between the base text and each candidate case. The *average word2vec embedding* is obtained by averaging the google news pretrained word2vec embedding of the most frequent 2,000 words from a case text.

A summary of the similarity scores and description can be found in Table 1.

We ranked the candidate cases based on each of these scores and also by combining them. Table 2 shows the range of ranks of *supporting* pairs based on one or more of the similarity scores. Our goal is to identify the methods that rank the *supporting* cases in the top. Clearly, despite being a simple method, Jaccard similarity ranks *supporting* cases on the top. Additionally, a combination of Jaccard similarity and Word2Vec embedding similarity, and Jaccard similarity and GloVe embedding similarity also ranks the *supporting* pairs at the top.

The classification task: We use this ranking approach to divide our dataset into subdatasets. We do so to group similar instances together. We believe that dividing the dataset based on the ranks would help the classifiers learn better in identifying the *supporting* cases.

We use the best performing similarity methods—methods with more *supporting* cases in the top 50—to split the dataset. Thus we conduct and present results by splitting the dataset into subdatsets using (i) Jaccard similarity, (ii) Jaccard similarity + Word2Vec embedding similarity, and (iii) Jaccard similarity + Glove embedding similarity.

2.2 Results

First, we divide the dataset into three subdatasets based on the ranks (<= 50, >50 & <150, and >= 150) using each of the aforementioned chosen methods. Then we build classification models for each of the subdataset. We used all individual similarity scores, their combinational scores, and all ranks as features for the classifiers. We experimented with support vector machines(SVM), logistic regression and xgboost classifiers. We found that the Logistic regression models always performed worse than SVMs and xgboost classifiers, and hence based on our pilot experiments, we decided to work with xgboost classifiers as they were comparable to SVMs in performance but have a faster execution time. We created a 80/20 stratified split for training and development over each of the subdataset. The stratification ensures that the training set and the development set have the same distribution of labels (*supporting/unsupporting* cases). The results on the development set for each subdataset is presented in Table 3. The

Table 2. Counts of supporting cases distributed across ranks based on different methods for Task 1. The total number of supporting cases are 2,680.

Rank range	**Method**										
	GloVe	w2v	BOW	Jacc.	GloVe+ w2v	GloVe+ BOW	GloVe+ Jacc.	w2v+ BOW	w2v+ Jacc.	BOW+ Jacc.	all
<= 50	1,757	1,925	285	2,252	1866	174	2,059	284	2,111	514	1,716
>50 & <150	559	494	445	401	497	845	368	1,149	411	1,700	708
>= 150	364	261	1,950	27	317	1661	253	1,247	158	466	256

Table 3. Results obtained in the development set over the subdatasets for Task 1. The first row is for *supporting* cases and the second row is for *unsupporting* cases

Jaccard									**Jaccard+w2v**									**Jaccard+glove**								
<=50			>50& <150			>=150			<=50			>50& <150			>=150			<=50			>50& <150			>=150		
P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
.67	.43	.52	.37	.23	.28	.33	.06	.10	.68	.41	.51	.72	.18	.29	.50	.09	.19	.68	.42	.52	.72	.12	.20	.63	.27	.38
.94	.98	.96	.99	.99	.99	.99	1.0	.99	.94	.98	.96	.99	.99	.99	.99	.99	.99	.95	.98	.96	.99	.99	.99	.99	.99	.99

results are an average of models build using 6 different random seeds. The final results on the test set, with F1 score of 0.38, was lower than our cross validation numbers reported here, which we believe happened due to overfitting of our models to the small dataset. In this task, imbalance in the data distribution played a big factor, and we intend to focus on improving it in the next year.

3 Task 2: Legal Case Entailment

Task 2 consists of recognizing entailment between a new case and a relevant case. Input is a decision fragment from an unseen case and a relevant case. Output is one or more specific paragraphs from the relevant case, which entail the given fragment of the unseen case. The approach we took consists of two stages: (1) similarity features-based ranking and (2) binary classification.

The first stage tries to rank the paragraphs of the relevant case based on their similarity with the given decision fragment. We evaluate this ranking process on the training dataset with respect to the total number of paragraphs of each relevant case. This evaluation allows us to set a variable threshold for the top K paragraphs selected in the subsequent classification stage. The approach shows promising results on the training set, yet less on the test set, due to the multiple parameters that the approach involves and which require fine-tuning to get the best performance for a specific dataset.

3.1 Experiments and Results

In the ranking phase, the paragraphs of the relevant case are ranked according to their similarity with the decision paragraphs. The similarity score is decided by combining different ranking given by multiple sentence vectorizers.

The vectorizers used include n-gram vectors, topic distribution vectors obtained through latent semantic analysis, universal sentence encoder vectors and averaged GloVe embeddings. The vectorizers are used with different levels of preprocessing (removing numbers, with and without stopwords, with and without lemmatization, etc.). With every sentence vectorizer, pairwise cosine coefficients are computed between the sentences of a paragraph from the relevant case and

the decision paragraph. Then, paragraphs of the relevant case are ranked according to the maximum cosine coefficient. In order to get the final ranking, two parameters, K_1 and K_2 are set. The first parameter, K_1, is function of the total number of paragraphs within the relevant case. Let us denote the number of paragraphs N. Then, $K_1 = \alpha N$, where $\alpha < 1$. Each paragraph p is given a rank R_i^p by the vectorizer v_i. Let F_p be the rate of the paragraph p falling into the top K_1, in other words, the proportion of times $R_i^p < K_1$. The paragraphs are then ranked according to the factor F_p.

We evaluate this ranking process on the training dataset by locating the golden paragraph in the obtained order. Table 4 summarizes the ranking results.

Table 4. Ranking results on the dataset for Task 2

Number of Paragraphs	Frequency	Rank at 80 percentile
N <=30	57%	3
30< N <=50	26%	5
N >50	17%	8

In the classification stage, a variable number of paragraphs, denoted K_2, are selected according to the table above (third column). 100 percentiles of the pairwise cosine coefficients of the top K_2 paragraphs are considered as features and fed to a classifier. Three random forest classifiers are trained, one for each interval of the number of paragraphs. Results of cross validation are presented in the the Table 5.

Table 5. Cross Validation results from Random Forest classifier for Task 2

Number of Paragraphs	Frequency	Precision	Recall	F1-score
N <=30	57%	0.71	0.66	0.68
30< N <=50	26%	0.66	0.68	0.67
N >50	17%	0.37	0.39	0.38

Although results on the training dataset are promising, the experiments have shown sensitivity of the results to the parameters K_1 and K_2,potentially explaining the relatively low global accuracy obtained on the unseen dataset. Our final test unseen F1 score was 0.41, indicating the models did not generalize very well on the test set. Given the size of the datasets, adding more regularization and trying out simpler methods might result in better performance on the test set, which we intend to focus more on the next year's tasks.

4 Task 3: Civil Code Article Retrieval

Task 3 involves selecting the most relevant Japanese Civil Code articles needed to answer given legal questions. The questions are from Japanese legal bar ex-

ams and given in the form of true/false questions. More than 70 percent of the questions can be answered by a single article, but some questions need multiple articles to infer the answer, while some questions have multiple associated articles each of which can answer them independently. Selecting the right number of articles for each question is important.

The text for Japanese Civil Code is provided by the competition organizers both in Japanese and English. Questions from past exams were used for training, and the test set for this year's COLIEE competition were also provided in Japanese and English. For task 3, we chose to use Japanese text to leverage Japanese domain knowledge from the team. We attempted multiple approaches but the ones that yielded most promising results were based on TF-IDF, which was motivated by the approach of last year's winning team [5]. After retrieving candidates, we ranked them based on the cosine similarity and applied a threshold to select only the relevant articles. For some types of questions, TF-IDF performed poorly. To accommodate those questions we used additional training data which we will explain in the next section.

4.1 Experiments

TF-IDF on Civil Code For task 3, we used TF-IDF and for each question, we computed cosine similarity between the question and articles from the civil code, and choose the closest article. Dealing with Japanese text has additional challenge because there are no spaces to indicates word segmentation. We decided to develop our own tokenizer instead of relying on third-party libraries to have control over how we segment the text. Japanese writing system has 3 components, kanji, hiranaga, and katakana. Kanjis are characters imported from Chinese and each character carries meanings. Hiragana and katanaka characters both originated in Japan, and each character do not carry meanings much like English alphabet. Hiragana is mainly used for conjunctions, post positional particles, and kanji suffixes. Katakana is typically used to phonetically represent words imported from foreign languages (e.g. computer, Internet, etc.). Since the legal terminologies and idioms are almost exclusively written in kanji, we extracted consecutive kanji characters and used hiragana and other punctuation characters as delimiters which we did not include as tokens. We did not explicitly exclude katakana words, but they are rare in the data set. We excluded some conjunctions that contain kanji characters such as 及び(and), 又は (or), and 若しくは (or), since they can appear adjacent to kanji terminologies, causing improper segmentation. This approach has an advantage of preserving idioms and terminologies. With traditional tokenizer such as Macab, idioms and terminologies are divided into smaller segments. For example, 家庭裁判所 (family court) is further divided into 家庭 (family) and 裁判所 (court). However, since a legal term carries specific meanings only when the words in it are put together, it is much more meaningful to consider it as a single token.

TF-IDF on Wikibook TFIDF approach is effective for definition questions where the definition of the article is directly asked. In such questions, there are

many overlapping words between them and their corresponding articles. However, many questions are more indirect and complex. For example, consider the following question-article pair in Table 6.

Table 6. Question Article Pair Examples: Task 3

Question:	An unborn child may not be given a gift on the donor's death.
Article:	(1) The enjoyment of private rights commences at birth.
	(2) Unless otherwise prohibited by applicable laws, regulations, or treaties, foreign nationals enjoy private rights.

In such cases, since there is no overlapping words, the TF-IDF approach did poorly. In the attempt to alleviate this problem, we searched for additional data to close the gap. We chose ja.wikibooks.org which has an entry for each civil code article. For each entry, there are additional explanation of the article as well as the original definition. It also has a section for cases, each of which is linked to the public court documents. We extracted the court documents as well. We appended the text from each article to the corresponding article from the provided COLIEE civil code article data, and used that as training data. However, this had mixed results. Even though it helped some specific cases, overall the training data without the additional data performed slightly better. Our highest scoring run (TRC3_1) was trained without those additional data.

4.2 Results

Table 7. Evaluation results compared to other runs (Task 3).

Run	lang	F2	Prec.	Recall	MAP	R_5	R_{10}	R_{30}
LLNTU	E/J	**0.6587**	0.6875	0.6622	**0.7604**	**0.8071**	**0.8571**	**0.9214**
JNLP.tbe	E/J	0.5532	0.5766	0.5670	**0.6618**	0.6857	0.7143	0.7786
cyber1	E/J	0.5290	0.5058	0.5536	0.5540	0.5500	0.6929	0.8000
HUKB-1	J	0.5160	0.4196	0.5908	0.5687	0.6714	0.7214	0.8143
CUBERT1	E/J	0.5139	0.5402	0.5193	0.5848	0.6429	0.6857	0.7071
TRC3_1	J	0.5011	0.4561	0.5357	**0.5978**	**0.6929**	**0.7714**	**0.8429**
OVGU_bm25	E/J	0.4768	0.4003	0.5342	0.5095	0.5929	0.6143	0.7214
TAXI_R3	E/J	0.4546	0.4393	0.5089	0.5057	0.5714	0.6143	0.6786
GK_NLP	E/J	0.4273	0.2857	0.4985	0.4982	0.5571	0.6357	0.7214
UA.tfidf	E/J	0.3913	0.4286	0.3869	0.4777	0.5429	0.6071	0.6714
HONto_hybrid	E/J	0.2822	0.2545	0.2991	0.0142	0.0071	0.0071	0.0500

Table 7 displays the evaluation results from the competition. Of the 3 runs submitted, TRC3_1 was the best run. In the competition, the results were evaluated based on the F2 Score. Our team ranked the 6th with the F2 score of

0.5011. As you see in the R_5, R_{10}, and R_{30} recall metrics, we ranked 2nd in those criteria. This means our approach effectively selected the right articles in the pool, but we were not successful in rank them, which resulted in lower f2 score.

5 Task 4: Civil Code Entailment

COLIEE has run an entailment challenge over Japanese bar exam questions since 2014. Before this year, the entailment task has resisted deep learning approaches because of the large premise sizes and the relatively small training set size. We implemented recent work on transfer-learning [8] and multi-sentence entailment [11] to set a foundation for moving beyond rule-based heuristics.

Although teams have attempted machine learning entailment models in past years [10], civil code entailment remains challenging because the system must evaluate several inter-dependent articles $[P_1, P_2, \ldots P_n]$ against a statement H to determine if H is true or false. Each civil code article represents a set of conditions, exceptions, and conclusions. H represents a set of facts and a conclusion. The system must apply the facts to the articles' conditions and determine if H came to the correct conclusion [5]. Table 8 shows a single sentence example matching the conditions with facts. In contrast, standard entailment datasets, like MultiNLI, remain comparatively simple because they focus on single sentence entailment [13].

Table 8. An example with conclusions , conditions , and facts : Task 4

Sentence	Sentence Text
P_1	A mandate shall terminate when the mandator or mandatary dies.
H	The mandate terminated upon the mandator's death .

In addition to the multiple sentence entailment, deep learning approaches have performed especially poorly on the Japanese Civil Code entailment tasks because the training set size is comparatively small to similar entailment tasks. Table 9 compares COLIEE 2020 civil code entailment data set to other single and multiple sentence data sets. The COLIEE data is 2 to 5 times smaller than the multiple sentence data set and 1000x smaller than the single sentence data sets. The COLIEE data set alone is too small for supervised machine learning[15].

We investigated one supervised approach and two transfer learning approaches to address the premise length and data set size issues. The naive supervised approach demonstrates the issue with supervised learning on small datasets. We then implemented a multi-sentence Natural Language Inference (NLI) model, Multee [11], which applies transfer learning from a single-sentence NLI dataset. Finally, we evaluated a multi-task model, Text-To-Text Transfer Transformer (T5) [8], which applies transfer learning across related NLP tasks.

Table 9. Entailment data set sizes.

Dataset	MultiNLI	SNLI	MultiRC	OpenBookQA	**COLIEE**
Train	392,702	550,152	5,131	4,957	**626**
Dev	20,000	10,000	4,848	500	**70**
Test	20,000	10,000	4,583	500	**112**

5.1 Experiments

Naive Feature-based Approach In order to determine the performance gain of our transformer classifiers on our validation data set, it was decided that a simple TF-IDF based classifier would be used as a benchmark. This TF-IDF approach utilized our team's customized tokenizer over all of the unigrams and bigrams in the data. The TF-IDF vectorizer was fit on all of the articles, wikibook data, and the public court documents.

With this TF-IDF vectorizer, features for both the questions and articles were created and then concatenated to form a unique feature vector for each question/article pair. Due to the high dimensionality of TF-IDF vectors, training and inference of machine learning models is expensive in both computation and memory. In order to mediate this, the dimensionality of the feature space was reduced to maintain 99% of its cummulative explained variance ratio. This resulted in a much smaller, compact feature space.

After reducing the feature space dimensionality, the next task was to identify and train a machine learning model with said features. A linear SVM classifier, a random forest classifier, and a gradient boosting classifier were all studied with various hyperparameter configurations. By studying the accuracy of the models' performance on the validation data set, it was determined that the gradient boosting classifier with 2 estimators and a .5 sampling fraction was the best model, achieving a validation accuracy of 64%.

Multiple Sentence NLI Based on the observations from past decomposable attention approaches [10][4], we experimented with Multee because it both increases the training set size and deals with multiple sentence entailment. Multee trains the model in two phases: single sentence pre-training and multiple sentences entailment. The first phase remained utterly unchanged compared to Multee's original experiments [11]. This phase pre-trained against the relatively massive SNLI and MultiNLI data sets and produced weight uses in the second training phase.

The second phase required some modifications to the Multee original experiments because the COLIEE data set is not multiple choice. The COLIEE model implemented a binary cross-entropy loss function for a single hypothesis. The second phase's training set includes both COLIEE and OpenBookQA examples because the OpenBookAQ provided a useful generalized multi-sentence entailment data set with complete sentences in the hypothesis. To make both data sets consistent with each other, we made two changes. First, we labeled the COLIEE

example as entailment and neutral for the Y and N, respectively. Second, we converted the OpenbookQA pairs in a single hypothesis format.

Unlike our next approach, both phases leveraged GloVe word embedding (680 Billion Words, 300 dimensions) instead of contextual embedding (i.e., BERT-style embeddings). A contextual embedding was not possible with this specific model because the premise sequence length create an attention matrix too large to fit on a 16GB GPU. Although the Multee weights each sentence, the model still performs cross-attention over the entire premise passage. The GloVe embeddings were chosen to avoid a truncated premise.

Once the second phase completed, we evaluated the COLIEE 2020 task 4 test set.

T5 Multi-Task For the second transfer-learning approach, we leverage T5 [8], which is a sequence to sequence model implemented with PyTorch transformers [14]. Like most BERT-style transformers, T5 includes both pre-training and fine-tuning stages. We decided to leverage the "t5-base" embedding for the pre-training because the domain-specific training sets tend to over-fit [8]. The t5-base embedding was pre-trained on an unsupervised denoising task. The pre-training task denoised text from the Colossal Clean Crawled Corpus (C4) [8].

While we leveraged generalized embedding, the fine-tuning stage updated the model with tasks against legal data. Each task followed the same format with an identifying prefix, input text and target text. The input and target text varied based on the task type and the format is noted in Table 10.

Table 10. Fine Tuning Tasks: Task 4

Task Type	Input Text	Target Text
Entailment:	hypothesis: An unborn ... premise: Article 3 (1) ...	Y or N
Denoise:	The enjoyment of private rights commences <id_1>. <id_1> at birth	

For the entailment task, the input text includes both the hypothesis (i.e., the exam question) and premise (i.e., the articles) with a prefix denoting the start of each respective sequence. The target text contains either "Y" for entailment or "N" representing does not entail.

For the denoising task, we followed the replacement span procedure documented in the T5 study [8]. The denoise task uses an unsupervised object where the data prep corrupts random text spans. The corrupted spanned is then replace with a single identifier token, for example "<id_1>" in Table 10. The target text contains each span identifier, followed by the missing text span.

The submitted T5 run (TRC3t5) used the following hyper-parameters, development set, and training sets. The model was fine-tuned with the transformer library T5-base's default settings except a learning rate of 1e-4, max target sequence of 100, and batch size of 16. The model trained on four separate tasks:

1. **Entailment:** COLIEE task 4 pairs (years from H18 to H29)

2. **Denoise:** Japanese Civil Code Article text
3. **Denoise:** Japanese Civil Code Article titles
4. **Denoise:** Wikibook articles on the Japanese Civil Code[1][2].

The TRC3t5 run configured early stopping. The early stopping process saves the model's parameters every training epoch and selects the best epoch based on an evaluation metric. Instead of using the run-time loss, the training run selected the epoch with the highest entailment accuracy on a validation set (year H30).

5.2 Results

Our Civil Code Entailment results are noted in Table 11. Given a bar exam question as a hypothesis H and a set of relevant articles as the premise $P = [P_1, P_2, \cdots P_n]$, determine if P entails H or not (i.e. Y or N). The baseline in the chart below assumes all 122 questions are entailed by the relevant statues. Of the 122 questions only 59 questions are support by the statute, thus the baseline has an accuracy of 0.5268. Our worst run (TRC3A) score below the baseline. Our two best runs (in bold) tie for second with UA at 0.625 accuracy.

Table 11. Evaluation results for Task 4 runs.

Run	Correct	Accuracy	Run	Correct	Accuracy
Baseline	59/122	0.5268	Baseline	59/122	0.5268
JNLP.BERTLaw	81	0.7232	**TRC3mt**	**70**	**0.6250**
UA_attention_final	70	0.6250	**TRC3t5**	**70**	**0.6250**
UA_roberta_final	70	0.6250	TRC3A	56	0.5000

Run TRC3mt The Multee approach preformed surprisingly well compared to this year's approaches with 70 out of 122 questions correct and tied for the second place. It's ranking is surprising for two reasons: lack of a legal training set and the passé non-contextual embeddings. Although Multee leverages a slightly more complicated 2-stage attention architecture, training data was mostly comprised of general NLI datasets. The result may indicate that at least some questions are answerable with generalize reasoning instead of domain-specific legal reasoning.

Additionally, Multee's ranking is surprising because the model used a simplistic non-contextual embedding, GloVe. If tuned properly, NLI models leveraging contextual should have outperformed Multee on multiple sentence entailment. In the SuperGlue leaderboard [12], both RoBERTa and T5 outperformed Multee's

[1] Extracted pages 民法第1条 to 民法第726条 from https://ja.wikibooks.org/wiki/コンメンタール民法

[2] The articles where translated from Japanese to English using the Google translation API

MultiRC F1a score of 69.9, with 84.4 and 88.1, respectively [11][6][8]. While contextual embedding have outperformed in a standard multi-sentence NLI benchmarks, both our T5 and UA's RoBERTa submission tied with the Multee on COLIEE specific task. The reason for this bears further investigation.

Run TRC3t5 Despite our best efforts to diversify the training set with multiple domain specific tasks, the T5 run appears to have over-fit on the training set. The T5 run's accuracy on the validation set (H30) was 0.6714 compare to 0.6250 on the test set. The domain specific training performed equally well compared to the more generalized training set used in the TRC3mt run (Table 11).

If we compare the T5 to the Multee predictions, the models differed on 36 different questions. While we can compare each question's linguistic features, each models exact strength and weakness are difficult to compare by simple accuracy measures because this analysis sheds little light on what the actual models have learned [9].

5.3 Discussion

Given BERT-style transformers topped the leader board this year, we need to discuss the statutory entailment tasks potential evolution. Although the transformer mechanisms demonstrate the ability to handle statutory entailment this year, they aren't actually demonstrating legal reasoning because they're mainly learning the legal's text linguistic form instead of learn the article's communicative intent [1]. Given the task binary nature, it's difficult to even assess what the models learned because attention-based models aren't transparent to legal experts [2]. Instead of a yes or no answer, the task should explain the conditions under a scenario matches the statutes and the resulting consequence [5].

6 Summary

In this paper, we demonstrated a variety of approach to all four COLIEE information extraction and entailment tasks. In the information extraction tasks (tasks 1 & 3) we examined TF-IDF and semantic similarity matching with mixed results. When dealing with small data sets, sometimes more simplistic approaches work best. While TRC3_1 leverage TF-IDF cosine-similarity and rudimentary word segmentation, it performed well with the MAP, R_5, R_{10}, and R_{30} metrics compared to other approaches.

In the entailment tasks (tasks 2 & 4), we explore similarity features and transfer learning. With the entailment tasks, the transfer learning approaches perform comparably well to the other participants. For task 4, we submitted two transfer learning approaches which fined-tuning on a generalize NLI dataset and domain specific task. Both approach tied for second with 0.6250 accuracy, yet yielded different answers on 36 questions.

References

1. Bender, E.M., Koller, A.: Climbing towards NLU: On meaning, form, and understanding in the age of data. In: Proceedings of ACL 2020 (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.463
2. Branting, K., Weiss, B., Brown, B., Pfeifer, C., Chakraborty, A., Ferro, L., Pfaff, M., Yeh, A.: Semi-supervised methods for explainable legal prediction. In: Proceedings of ICAIL 2019 (2019). https://doi.org/10.1145/3322640.3326723
3. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method (2014), http://arxiv.org/abs/1402.3722, cite arxiv:1402.3722
4. Hudzina, J., Vacek, T., Madan, K., Custis, T., Schilder, F.: Statutory entailment using similarity features and decomposable attention models. In: Proceedings of COLIEE 2019 (2019)
5. Kim, M., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of ICAIL 2019 (2019). https://doi.org/10.1145/3322640.3326742
6. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019), https://arxiv.org/abs/1907.11692
7. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of EMNLP (2014)
8. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
9. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: Behavioral testing of NLP models with CheckList. In: ACL. pp. 4902–4912 (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.442
10. Son, N.T., Phan, V.A., Minh, N.L.: Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models. In: Proceedings of COLIEE 2017 (2017), http://www.easychair.org/publications/paper/347231
11. Trivedi, H., Kwon, H., Khot, T., Sabharwal, A., Balasubramanian, N.: Repurposing entailment for multi-hop question answering tasks. In: Proceedings of NAACL 2019 (Jun 2019). https://doi.org/10.18653/v1/N19-1302
12. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: SuperGLUE: A stickier benchmark for general-purpose language understanding systems. arXiv preprint 1905.00537 (2019)
13. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of NAACL-HLT 2018 (2018), http://aclweb.org/anthology/N18-1101
14. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Huggingface's transformers: State-of-the-art natural language processing. ArXiv **abs/1910.03771** (2019)
15. Yoshioka, M., Kano, Y., Kiyota, N., Satoh, K.: Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018 (2018), https://sites.ualberta.ca/řabelo/COLIEE2019/COLIEE2018_SL_summary.pdf

UB_Botswana at COLIEE 2020 Case Law Retrieval

Tebo Leburu-Dingalo, Edwin Thuma, Nkwebi Peace Motlogelwa, and Monkgogi Mudongo

Department of Computer Science, University of Botswana
{leburut,thumae,motlogel,mudongom}@ub.ac.bw

Abstract. In this paper we present two approaches to measure similarity between documents as contribution towards the 2020 COLIEE legal case retrieval task. Given a query case the task requires the retrieval of noticed cases relevant to support the decision of the query case. Our methods implement learning to rank techniques with various sets of query dependent features. The first method uses term weighting models and term dependence proximity models as features while the second method uses token-based and vector based text similarity features. In context of the competition, the first method compared fairly with the other systems while performance for the second method was very low. The first method results indicate potential for improvement since it still managed to perform fairly with no fine-tuning of the parameters of the features used.

Keywords: Legal Information Retrieval, Learning to Rank, Text Similarity, Term Weighting

1 Introduction

Provision of efficient information access tools in the legal field has been an active area of research for many years dating back to when legal material was only available in manual format [26]. Subsequently as legal material grew in volume and complexity it became necessary to avail this material in electronic format. Hence, huge amounts of legal documents including case law reports, statute articles and legal research have been digitized over the years. However, the challenge is now in developing tools that can provide efficient and effective access to this electronic content. Existing Information Retrieval (IR) tools that have performed well in other fields of search have been found ineffective for legal content [4]. Hence, there is now concerted effort towards addressing the limitations and drawbacks of these through development of novel techniques and improvement of the state-of-the-art. In support of this effort international evaluation campaigns have been established whose aim is to promote collaboration among researchers of different backgrounds and expertise working towards this goal. One of this platforms is the Competition on Legal Information Extraction and Entailment (COLIEE)[1]

[1] https://sites.ualberta.ca/ rabelo/COLIEE2020/

competition which provides an environment for researchers to develop comparable IR systems in the legal field through the provision of standardized data sets [20]. For the current year the competition focused on four aspects of legal IR divided into different tasks. Task 1 addresses the retrieval of case law articles to support the decision of a current case. Task 2 seeks systems capable of comparing and identifying paragraph from existing cases that entail the decision of a new case. Task 3 focuses on extracting a subset of civil code articles that can answer a legal bar exam. Task 4 focuses on determining whether relevant articles retrieved for a legal question entail it or not. In this paper we propose two approaches towards Task 1. Based on the description of the task our assumption is that supporting cases to a current case are more likely to have similar character sequences as the current case [9]. Hence for our approaches we explore the use of methods geared towards measuring lexical differences between the documents as opposed to semantic differences. To this end we deploy existing Learning to Rank (LTR) techniques with different sets of lexical features. In the first instance we explore the use of term weighting models and term dependence proximity models, which rely on indexing structures to identify relationships between a query and a document. In the second instance we use a combination of two categories of similarity measures. The first category measures the similarity between sets of document tokens, while the second category measures the similarity by taking the distance between vector representations of documents.

The remainder of the paper is structured as follows: Section 2 discusses related work, Section 3 gives a brief background on approaches used in our experiments. Section 4 describes our experimental setup, Section 5 discusses our results, while Section 6 is the conclusion and way forward.

2 Related Work

Work towards the improvement of automated retrieval of relevant case law reports given a current case also known as precedence retrieval has gained interest as an area of research in recent years. The interest has been sparked by proliferation in the number of case law reports over the years which has resulted in the activity being both tedious and prone to errors. Several studies have thus been undertaken to tackle this issue. These include the work by Shao et al. [24] who developed a novel method that utilizes a BERT model fine-tuned on a case law entailment dataset. The model infers the relevance between two cases by capturing their paragraph-level semantic relationships. Gain et.al [8] proposed several methods which utilize a variety of approaches to retrieve relevant cases given a current case. In the first approach, they used Doc2vec to compute the similarity between the current case and the candidate cases. In the second approach, they used BM25 as the basis for measuring the similarity between the query case and the candidate cases. The last approach combined document scores obtained from the previous approaches to get a final ranking. Their system ranked third amongst systems that participated at FIRE 2019[2] Artificial Intelligence

[2] http://fire.irsi.res.in/fire/2019/home

for Legal Assistance (AILA)[3] track. In another study, Rossi et al. [23] used text summarization and a generalized language model(BERT) to generate pairwise embeddings that are submitted to a Multi-Layer Perceptron for pairwise classification. Their method performed reasonably well in the retrieval of noticed cases.

3 Description of Methods

Our general approach is based on the use of LTR techniques with query dependent features derived from term weighting models, term dependence proximity models, token-based and vector based measures. Below we give a brief description of approaches used.

3.1 Learning to Rank (LTR)

LTR techniques use supervised Machine Learning (ML) algorithms to learn an appropriate combination of features for constructing an optimal ranking model [13, 14]. Once constructed, the model can be used to re-rank a sample of retrieved documents for a given query to obtain a final ranking. The general steps for constructing the model are as follows [14]:

Given a set of training queries and documents and relevance assessments or ground-truth, retrieve a sample of k documents using an initial term weighting model such as BM25. For each top k document, generate a set of features. The features can be query dependent features such as term weighting models or term dependence proximity models. Query dependent features such as fraction of stop words or document length can also be used. The existing relevance judgments are then used to label a feature vector for each document. A model can then be learned from these feature vectors, This model can then be used to re-rank results of unseen queries obtained using the same retrieval strategy.

LTR techniques can be classified as pairwise, pointwise or listwise. Pairwise models transform ranking into a classification problem. Given a pair of documents a binary classifier is learned that attempts to obtain an optimal ordering for the pair. In a pointwise approach each query-document pair has a numerical or ordinal score. Therefore ranking can be approximated using regression, classification and ordinal regression. Listwise approaches learn the ranking model by optimizing an IR evaluation measure averaged over all queries in the training set such as MAP [13, 14].

In this work we deploy LambdaMART a state-of-the-art listwise approach that has been found effective when compared with other approaches [13, 15]. LambdaMART [2] is a combination of the tree-boosting optimization method MART [7] and the listwise ranking model LambdaRank [3, 1]. LambdaMART as Burgess et al. [2] explains uses gradient boosted regression trees to learn pairwise

[3] https://sites.google.com/view/fire-2019-aila/

preferences over documents based on NDCG gain obtained by swapping the rank position on any given pair. The learned model is the linear combination of outputs from a set of regression trees T, known as an ensemble [15]. To obtain a score for a document d the nodes of a particular tree t are traversed depending on a series of decisions based on the document's vector of feature values (fd) expressed as:

$$score(d, Q) = \sum_{t \in T} t(f_d) \tag{1}$$

3.2 LTR with Term Weighting Models and Term Dependence Proximity Models

Our first proposed approach uses term weighting models and term dependence proximity models as features. A total of five features are used. These are four term weighting models namely TF-IDF, DPH, BM25, PL2 and the markov random fields with the sequential dependence Score modifier.

TF-IDF is a numerical statistic that is calculated by taking the product of two components; term frequency (tf) and inverse document frequency (idf). tf refers to the number of times term t occurs in document d [21]. The basic idf calculation is as follows:

$$idf(t) = \log \frac{N}{n_i} \tag{2}$$

where N is the total number of documents in collection C, and n_i is the number of documents the term t occurs in.

The traditional BM25 weighting scheme is a bag of words model which incorporates the structure of a document in scoring [22]. Score for a document d for a query Q is computed as follows:

$$score_{BM25}(d, Q) = \sum_{t \in Q} w^{(1)} \cdot \frac{(k_1 + 1)tfn}{k_1 + tfn} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf}. \tag{3}$$

where qtf is the number of occurrences of a given term t in the query Q. k_1 and k_3 are parameters of the model. $w^{(1)}$ denotes the Robertson-Spark Jones (RSJ) weights, which is an inverse document frequency (IDF) factor and is given by:

$$w^{(1)} = \log \frac{N - dft + 0.5}{dft + 0.5} \tag{4}$$

Where N is the number of documents in the collection and dft is the number of documents in the collection that have a term t.

tfn is the normalised within document term frequency and is given by:

$$tfn = \frac{tf}{1 - b + b \cdot \frac{1}{avg_l}} \tag{5}$$

where tf is the frequency of the query term t in the document d and b is the term frequency normalization hyper-parameter, l is the length of the document and avg_l is the average document length in the collection.

DPH is a hyper-geometric parameter free model [12] which computes the score of a document d for a given query Q as follows:

$$score_{DPH}(d,Q) = \sum_{t \in Q} qtf \cdot norm \cdot \left(tf \cdot \log((tf \cdot \frac{avg\_l}{l}) \cdot (\frac{N}{tfc})) + 0.5 \cdot \log(2 \cdot \pi \cdot tf \cdot (1 - t_{MLE})) \right) \tag{6}$$

where qtf, tf and tfc are the frequencies of the term t in the query Q , in the document d and in the collection C respectively. N is number of documents in the collection C, avg_l is the average length of documents in the collection C and l is the length of the document d. $t_{MLE} = \frac{tf}{l}$ and $norm = \frac{(1-t_{MLE})^2}{tf+1}$.

The PL2 model [18] computes a score for a document d in collection C given query Q as follows:

$$score_{PL2}(d,Q) = \sum_{t \in Q} qtw \left(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn) \right) \tag{7}$$

where $score(d,Q)$ is the relevance score of a document d for a given query Q. $\lambda = \frac{tfc}{N}$ is the mean and variance of a Poisson distribution, tfc is the frequency of the term t in the collection C while N is the number of documents in the collection. The normalised query term frequency is given by $qtfn = \frac{qtf}{qtf_{max}}$, where qtf_{max} is the maximum query term frequency among the query terms and qtf is the query term frequency. qtw is the query term weight and is given by $\frac{qtfn}{tfn+1}$, where tfn is the Normalisation 2 of the term frequency tf of the term t in a document d and is expressed as:

$$tfn = tf \cdot \log_2 \left(1 + b \frac{avg\_l}{l} \right), (b > 0) \tag{8}$$

In the above expression, l is the length of the document d, avg_l is the average document length in the collection and b is a hyper-parameter.

DSM computes a term proximity score, this entails scoring pairs of terms co-occuring in a pre-defined window of text [17] the model calculates a score for document d given query Q as follows:

$$score(d,Q) = \lambda_1 \cdot \sum_{t \in Q} score(d,t) + \lambda_2 \cdot \sum_{p \in Q_2} score(d,p) \tag{9}$$

Where $score(d,t)$ is the score assigned to a query term t in the document d, p corresponds to a pair of query terms that appear within the query Q, $score(d,p)$ is the score assigned to a pair of query terms p in the document d, and Q_2 is a set of pairs of query terms.

3.3 LTR with Text Similarity Measures

Our second approach uses text similarity measures as features. These are four text distance measures based on the bag, levenshtein, sorensen-dice, and jaccard statistics [5, 6, 9–11, 25]. Also used as part of this feature vector are three

pairwise distance measures cosine, manhattan and euclidian [9, 26]. Bag distance computes the distance between two strings as follows [26]:

$$d_{bag}(x, y) = max\{|x - y|, |y - x|\} \tag{10}$$

where x , y denotes the set of symbols in the two strings respectively.

Levenshtein calculates distance between string sequences by counting the minimum number of character edits operations (insertions, deletions or substitutions) needed to transform one sequence into the other [9].

Given strings to compare, the Sorensen-dice coefficient gives a measure of twice the number of common terms in the strings, divided by the total number of terms in both strings [9]. Given string X and string Y Sorensen-dice distance is computed as follows:

$$d_{dice} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{11}$$

Jaccard similarity computes the number of shared terms over the number of all unique terms in both strings being compared [9]. Given string X and string Y the jaccard score is calculated as:

$$J(X, Y) = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \tag{12}$$

The Euclidean distance or L2 distance is a geometric measure used to measure distance between two vectors. The distance is calculated by taking the square root of the sum of squared differences between corresponding elements of the two vectors being compared [9]. Hence given two vectors x and y the distance can be computed as follows:

$$d\left(x, y\right) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2} \tag{13}$$

Manhattan, or city block distance computes distance by following a grid-like path from one data point to the other. If two items are being compared the distance is calculated as the sum of the differences of their corresponding components, so for two points x, y with any values the distance is calculated as follows:

$$d_{block}(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{14}$$

Cosine distance measures the cosine of the angle between two vectors being compared. The distance is inversely proportional to cosine similarity hence if the angle between two points is small then this indicates the points share similar properties than the points that are far apart [9]. Given vector x and vector y, cosine similarity is computed as follows:

$$Cosine_{Similarity}(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} \sqrt{\sum_{i=1}^{n} {y_i}^2}} \quad (15)$$

The cosine distance is then calculated as: $d_{cos} = 1 - Cosine_{Similarity}(x, y)$

4 Experimental Setting

Our experiments are conducted on the COLIEE 2020 Task 1 dataset. Task 1 is a legal case retrieval task, which requires reading a new case Q and extracting supporting cases S1;S2.....Sn that support the decision of case Q from a case law corpus. Cases are considered relevant or 'noticed' if they can support the decision of the query case. The dataset comprised of a total of 650 query cases each with 200 candidate cases. 520 of these were provided for training with their relevance judgments referred to as 'noticed cases' and the rest were test data with no 'noticed cases'.

In our first experimental evaluation, we use Terrier-4.2[4] , an open source IR platform [16]. All documents in the dataset are used as corpus for this experiment. The documents are pre-processed before indexing. This entails tokenizing the text, removing stop words, and stemming each remaining token using the Porter stemming algorithm [19]. All term weighting models are used with default parameters. For convenience, features are obtained by utilizing the FAT framework [15] which allows the computation of many features within one run. A jforests[5] LTR technique is then used to learn a LambdaMART model [2].

The Python[6] language in combination with various modules and libraries is used for our second experimental evaluation. Similar to the first experiment the documents are first tokenized, stop words removed and the remaining tokens stemmed with the Porter Stemmer [19]. The textdistance[7] library is then deployed to calculate bag, jaccard and sorensen-dice distances between the query case and each candidate case. To obtain the levenshtein distance we deploy the fuzzywuzzy[8] package. The cosine similarity, as well as manhattan and euclidian distances are computed using the sklearn[9] pairwise metrics on the query and document vectors obtained using the TF-IDF vectorizer. To obtain an optimal ranking model for the features we use Ranklib-2.8[10], which is a library of learning rank algorithms. From these we select, train and test a LambdaMART [2] model with the metric to optimize on the training data set to NDCG@20.

[4] http://terrier.org/
[5] https://github.com/yasserg/jforests
[6] https://www.python.org/
[7] https://pypi.org/project/textdistance/
[8] https://pypi.org/project/fuzzywuzzy/
[9] https://scikit-learn.org
[10] https://sourceforge.net/p/lemur/wiki/RankLib/

5 Results and Discussion

Our results from both experiments were submitted to the COLIEE competition for evaluation by the organizers. The evaluation for Task 1 uses precision, recall and the F-measure. The results of the task based on the F-measure are shown in Table 1. From these results, our first approach (UB_RUN1) which deploys LTR with term weighting and dependence scores obtains a modest F-measure of 0.5866 for the task. However the second approach (UB_RUN2) that uses text similarity features did not perform well managing to achieve only 0.0592. The first results show that the combination of features used has the potential to meaningfully compare long text documents. We thus expect that fine-tuning the models would yield even better performance from the system. The second model did not perform well even though it used a combination of features presumably thought to work well in text comparisons. We thus note the need to carefully consider the type of features to combine for a re-ranking problem regardless of their performance in other tasks. Also to take into consideration is the use of documents' semantic properties instead of relying solely on lexical features.

6 Conclusion

In this paper, we attempt to address the problem of legal case retrieval by employing methods that attempt to find optimal ranking of results obtained from several IR systems. We propose the use of learning to rank models with a variety of features including term weighting models, token based distance measures and similarity measures for the vector space. Although one of our original submission to the competition performed poorly, having reviewed the approach we are convinced that our approaches have potential to improve retrieval performance of legal case retrieval systems. Hence going forward we plan to perform more similar experiments and work on fine tuning parameters for our models to improve performance.

References

1. Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, October 2002.
2. Christopher J. C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical report, Microsoft Research, 2010.
3. Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 193–200, Cambridge, MA, USA, 2006. MIT Press.
4. Danilo Carvalho, Vu Tran, Van-Khanh Tran, and Le-Nguyen Minh. Improving legal information retrieval by distributional composition with term order probabilities. 06 2017.

Table 1. COLIEE 2020 Task 1 Results.

Team	**File**	**F1**
cyber	task1 cyber02.txt	0.6774
cyber	task1 cyber03.txt	0.6768
TLIR	t1_run3_thuir.txt	0.6682
cyber	task1 cyber01.txt	0.6503
JNLP	JNLP.task1.BMW25.txt	0.6397
TLIR	t1_run2_thuir.txt	0.6379
JNLP	JNLP.task1.W25.txt	0.6358
JNLP	JNLP.task1.W30.txt	0.6278
UB	UB_RUN1.res	0.5866
iiest	iiest_bm26_t1_3.txt	0.5288
iiest	iiest_bm25_t1_2.txt	0.5272
TLIR	t1_run1_thuir.txt	0.5148
iiest	iiest_ps_t1_1.txt	0.4821
TR	submission2	0.38
TR	submission3	0.3792
TR	submission1	0.3388
AUT99	AUTIRT1R1.txt	0.2658
DACCO	T1 DACCO.txt	0.2077
AUT99	AUTIRT1R2.txt	0.1617
AUT99	AUTIRT1R3.txt	0.0898
UB	UB_RUN2.res	0.0592
taxi	task1_TAXICATTFCV.txt	0.0457

5. A. Chao, R. Chazdon, R. Colwell, and Tsung-Jen Shen. A new statistical approach for assessing similarity of species composition with incidence and abundance data. *Ecology Letters*, 8:148–159, 2004.
6. Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
7. Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.
8. Baban Gain, Dibyanayan Bandyopadhyay, Arkadipta De, Tanik Saikh, and A. Ekbal. Iitp at aila 2019: System report for artificial intelligence for legal assistance shared task. In *FIRE*, 2019.
9. Wael H Gomaa, Aly A Fahmy, et al. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.
10. P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Bulletin de la Société vaudoise des sciences naturelles. Impr. Corbaz, 1901.
11. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics. Doklady*, 10:707–710, 1966.

12. Nut Limsopatham, C. MacDonald, I. Ounis, Graham McDonald, and Matt-Mouley Bouamrane. University of glasgow at medical records track: Experiments with terrier. In *TREC*, 2011.
13. Tie-Yan Liu, Thorsten Joachims, Hang Li, and Chengxiang Zhai. Introduction to special issue on learning to rank for information retrieval. *Inf. Retr.*, 13(3):197–200, June 2010.
14. Craig Macdonald, Rodrygo L. Santos, and Iadh Ounis. The whens and hows of learning to rank for web search. *Inf. Retr.*, 16(5):584–628, October 2013.
15. Craig Macdonald, Rodrygo L.T. Santos, Iadh Ounis, and Ben He. About learning models with multiple query-dependent features. *ACM Trans. Inf. Syst.*, 31(3), August 2013.
16. Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. In *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, ECIR'05, page 517–519, Berlin, Heidelberg, 2005. Springer-Verlag.
17. Jie Peng, Craig Macdonald, Ben He, Vassilis Plachouras, and Iadh Ounis. Incorporating term dependency in the dfr framework. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 843–844, New York, NY, USA, 2007. Association for Computing Machinery.
18. Vassilis Plachouras and Iadh Ounis. Multinomial randomness models for retrieval with document fields. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, page 28–39, Berlin, Heidelberg, 2007. Springer-Verlag.
19. M. F. Porter. *An Algorithm for Suffix Stripping*, page 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
20. Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. A summary of the coliee 2019 competition. In *JSAI International Symposium on Artificial Intelligence*, pages 34–49. Springer, 2019.
21. S. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *J. Documentation*, 60:503–520, 2004.
22. S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.
23. Julien Rossi and Evangelos Kanoulas. Legal information retrieval with generalized language models. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE*, 2019.
24. Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3501–3507. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.
25. T.J. Sørensen. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. I kommission hos E. Munksgaard, 1948.
26. Howard Turtle. Text retrieval in the legal world. *Artif. Intell. Law*, 3(1–2):5–54, March 1995.

Significance of Textual Representation in Legal Case Retrieval and Entailment

Arpan Mandal[1], Saptarshi Ghosh[2], Kripabandhu Ghosh[3], and Sekhar Mandal[1]

[1] Indian Institute of Engineering Science and Technology, Shibpur, India
[2] Indian Institute of Technology, Kharagpur, India
[3] Indian Institute of Science Education and Research, Kolkata, India

Abstract. This paper describes our methods for the legal case retrieval and entailment tasks of the Competition on Legal Information Extraction/Entailment (COLIEE) 2020. Almost all retrieval applications over textual documents involve some pre-processing whereby Natural Language Processing methods are used to construct representations of the documents, before further use of the representations by the retrieval models. We show that instead of just using a bag of all words as the document representation, a bag of carefully chosen n-grams can better represent the documents for the tasks of case retrieval and entailment. We observe that, using advanced filtering mechanisms to select a set of unigrams, bigrams and trigrams gives us the best representation for a document. We also propose a novel method that is built upon the popular BM25 retrieval model with a simple intuitive modification of the scoring function. Out of the 22 different supervised and unsupervised methods submitted in each task in the COLIEE 2020 competition, our unsupervised method achieved the 10^{th} position in Task 1 (case retrieval) and the 9^{th} position in Task 2 (case entailment).

Keywords: Legal Retrieval· Legal Entailment · Information Retrieval.

1 Introduction

With the growing number of digitally available legal documents, it is of utmost importance that the repetitive tasks of experts be aided with automatic legal case retrieval and case entailment tools. In this regard, the COLIEE 2020 tasks[4] aims at building better retrieval models for legal documents. COLIEE has been organizing the case retrieval and entailment tasks since 2018. The target of these tasks is *not* to replace the experts, but to aid them in a way that speeds up the manual work and makes it less intensive.

Previous attempts towards legal case retrieval have been made in [4,5,9]. The recent trends in legal entailment tasks include interest in using neural network models such as BERT [2], ELMo [8] and ULMFit [3]. Unsupervised attempts such as using Word2Vec centroids as document representations were also made. Apart from COLIEE, there have been other shared tasks / data competitions

[4] `https://sites.ualberta.ca/~rabelo/COLIEE2020/`

as well that had the task of legal case retrieval [1, 6]. Though several methods for legal case retrieval have been presented as part of these tracks, almost none of the methods have systematically explored the importance of the document representations for the task. In this work, we attempt to bridge this gap.

In [12], it was shown that a 'bag of entities' representation performed better than a 'bag of words' representation in similar tasks of document retrieval. Motivated by this observation, in this work, we build our textual representation of legal documents utilizing the bag of entities approach. Specifically, we adopt an approach similar to what is used by keyphrase extraction algorithms [7, 11] to construct the representations of legal documents. Our method is also inspired by the POS filtering done in [10]. Also, we propose an unsupervised scoring function for legal document retrieval. The main contributions of our work are as follows:

1. We develop a better representation of legal documents using a bag-of-ngrams approach, and
2. We propose a simple modification of the popular BM25 retrieval model for better legal retrieval performance.

The rest of the paper is organized as follows. Firstly, the task and the dataset are described in Section 2. Then we describe our methodology in Section 3. Section 4 shows a comparison of our methods with the other methods submitted to the COLIEE 2020 competition. Finally the paper is concluded in Section 5.

2 Dataset and Task Descriptions

The Competition on Legal Information Extraction/Entailment (COLIEE) 2020 had four tasks in total out of which we participated in Task 1 and Task 2. Task 1 corresponds with legal case retrieval and task 2 is related to legal case entailment.

2.1 Task 1: Legal Case Retrieval

In this task, the challenge was to retrieve a set of supporting cases $S_1, S_2, ...S_n$ for a given base case Q. The supporting cases that support the decision taken in base case are called *noticed cases*.

The dataset for this task has two parts – (1) the training set of base cases for each of which the set of noticed cases were given, and (2) a test set of base cases for which the participants are required to find the noticed cases. For each base case (either in the training set or in the test set), there was a search space of 200 candidate documents, i.e., for each base case the participants were provided a set of 200 candidate cases among which the participants were to retrieve the set of noticed cases. The summary of the dataset can be found in Table 1.

2.2 Task 2: The Legal Case Entailment Task

This task involves the identification of a paragraph from an existing relevant case that entails the decision of a recent case. Given a decision Q of a recent case and a relevant case R, a specific paragraph that entails the decision Q needs to be identified.

Training Set	
Number of base cases	520
Number of candidate cases per base case	200
Mean number of noticed cases per base case	5.15
Standard Deviation of number of noticed cases	4.49
Test Set	
Number of base cases	130
Number of candidate cases per base case	200
Mean number of noticed cases per base case	4.89
Standard Deviation of number of noticed cases	4.01

Table 1. Properties of Task 1 Dataset

Dataset for Task 2 Similar to the dataset in the first task this dataset also had two parts – (1) the training set of judgements and, (2) the testing set of judgements. Each judgement in the training set were provided with the entailed paragraphs. The participants were to find the entailed paragraphs for the judgements in the testing set. The dataset properties are described in Table 2.

Training Set	
Number of base judgements	325
Number of candidate paragraphs per base judgments	35.52
Mean number of entailed paragraphs per base judgement	1.15
Standard Deviation of number of entailed paragraphs	0.45
Testing Set	
Number of base case judgements	100
Number of candidate paragraphs per base judgments	36.72
Mean number of entailed paragraphs per base judgement	1.25
Standard Deviation of number of entailed paragraphs	0.46

Table 2. Properties of Task 2 Dataset

3 Methodology

Our method is strictly unsupervised in nature and has two main components. The first component deals with the representation of the document and the second deals with a novel scoring function built upon BM25. The following two subsections describe the two involved components in detail.

3.1 Representing text as a Filtered-Bag-of-Ngrams (FIBONG)

In most NLP applications, there are two popular representations of text that are generally used – (1) Bag-Of-Words and, (2) Sequence-of-Words. The second

representation preserves all information but is suited to neural network based models such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). Bag-of-words (BOW) is generally used for frequency based unsupervised models such as TF-IDF and others. In addition to BOW, bag-of-entities (BOE) has recently been shown to perform well with unsupervised frequency based models such as BM25. For this reason, we create a representation that we call FIltered-Bag-Of-NGrams (or FIBONG in short).

Our FIBONG representation is motivated by the observation that *keyphrases identified by keyphrase extraction algorithms [7,11] can plausibly be a good representation of legal documents*. Keyphrases of a document are meant to summarise the content of the document. Specifically, in the context of legal documents, keyphrases are also referred to as *catchphrases* [7]. So the keyphrases / catchphrases should serve as a good representation of a legal document. In fact, most keyphrase extraction algorithms first extract candidate phrases from a document, and then select a few of them as the final keyphrases. We propose to represent a legal document by these candidate phrases / keyphrases.

Our FIBONG representation is inspired by a well-known keyphrase extraction algorithm KEA [11] that uses a form of n-grams (after stopword removal) as candidate phrases. Also FIBONG uses stopword removal and parts-of-speech (POS) filtering that has been shown to perform well in case of legal question answering systems [10]. Specifically, we perform the following steps to get the final FIBONG representation of a document.

1. We get all unigrams, bigrams and trigrams as separate bags of n-grams.
2. *For the bag of unigrams*, we replace dates with some wildcard token, consider only those tokens that are purely alphabetic, consider only those that has three or more alphabets, remove all stopwords, remove anything that is not a verb or noun or adjective, lemmatize every word.
3. *For the bag of bigrams*, we consider only those that does not end or start with a stopword, consider only those that start or end with a verb or noun or adjective,
4. *For the bag of trigrams*, we apply all filters that were applied on bigrams also, do not consider anything that starts or ends with any word of size two or less alphabets, also do not consider anything that ends with a verb.

The code for extracting the FIBONG representation for a given document is implemented in Python and is available online on GitHub[5]. Examples of FIBONG representations for two given texts is shown in Table 3.

3.2 A novel scoring function based on BM25

After we represent the documents using FIBONG, the second step is to score a candidate document with respect to a query document. For this, we propose our own scoring function that can score a given document with respect to a given query text.

[5] `https://github.com/amarnamarpan/fibong-a-new-way-of-representing-text-documents`

Text	FIBONG representation
Candidate phrases of keyphrase extraction algorithms can be a good form of representation of the document.	candidate, phrase, keyphrase, extraction, algorithm, good, form, representation, document, candidate phrases, keyphrase extraction, extraction algorithms, good form, phrases of keyphrase, keyphrase extraction algorithms, form of representation
Our method is strictly unsupervised in nature and has two main components	method, strictly, unsupervised, nature, main, component, strictly unsupervised, two main, main components, method is strictly, two main components

Table 3. Examples of Fibong representations.

Our method is a simple modification of the famous BM25 model that has remained a pioneer in document retrieval techniques.[6] The BM25 scoring function looks like the log-normalized TF-IDF function along with two parameters k_1 and b:

$$BM25(t,d) = IDF(t) \cdot \frac{TF(t,d) \cdot (k_1 + 1)}{TF(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{1}$$

where $TF(t,d)$ is the frequency of the term t in document d, $|d|$ is the length of the document d in words, $avgdl$ is the average document length in the text collection, and k_1 and b are free parameters.

The IDF (inverse document frequency) in the above function is the log normalized reciprocal of document frequency. The basic function of the IDF term is to capture the importance of the term t in the corpus. The main hypothesis behind its working is that, the lesser the number of documents a term appears in, the more important the term is. This hypothesis works well in general. But the IDF term alone does not suffice to give the accurate importance of the term in the corpus, especially when the retrieval is being done over documents from a single domain. For instance, in the legal domain, say a bigram like *"high court"* has appeared in many documents, due to the reason that we are only considering legal documents. This does not make the term *"high court"* any less important.

To mitigate this problem we introduce another factor CF which is essentially a standardized and normalized version of the collection frequency (cf) of a term. Collection-frequency, in comparison to IDF, gives us another form of corpus-specific importance that normalizes the pitfalls of using just IDF as the measure of term-importance. $CF(t)$ for a term t is calculated by first standardizing the collection frequency $cf(t)$ by Z-Score standardization[7], to even out the distribution of collection frequencies. Then MinMax scaling[8] is used to scale the values within the range $[0, 1]$. Thus $CF(t)$ for a term t is calculated as follows:

$$CF(t) = p_1 + \log(1 + \text{MinMax}(\text{Standardize}(cf(t))) \cdot p_2) \tag{2}$$

[6] `https://en.wikipedia.org/wiki/Okapi_BM25`
[7] `https://en.wikipedia.org/wiki/Standard_score`
[8] `https://en.wikipedia.org/wiki/Feature_scaling`

Note that the two parameters p_1 and p_2 control the effect of $CF(t)$ when the IDF term is augmented with it. The parameter p_2 directly influences the value of CF, and p_1 buffers the effect of p_2 on CF. Together they can determine the amount of impact that the CF term has upon IDF. After experimenting with values within a range of $[-5, 10]$ with increments of 0.1 in between, we determined the free parameters p_1 and p_2 as 0.3 and 0.5 respectively. These values gave us the best results for the COLIEE tasks, and may vary according to the dataset and the task involved.

So, in our proposed methodology, the final similarity score between a term t and a document d is computed as follows:

$$Score(t,d) = CF(t) \cdot IDF(t) \cdot \frac{TF(t,d) \cdot (k_1 + 1)}{TF(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{3}$$

The notations used in this equation are the same as those in Eqn. 1, and $CF(t)$ is the standardized and normalized collection frequency of the term t as computed using Eqn. 2. For the COLIEE retrieval task, we used the parameter values $k_1 = 1.5$ and $b = 0.75$.

For Task 1, our proposed method gives, for each base case (query case) Q, a ranked list of all the supporting cases in decreasing order of relevance (see Section 2). For Task 2, our proposed method gives, for each given decision Q, a ranked list of all paragraphs in the relevant case R in decreasing order of similarity with Q.

4 Experimental Results

In this section, we present the observed performance of our methodologies over the modified scoring function.

Evaluation: As per the rules of the COLIEE 2020 competition, the participants were required to submit a *set* of relevant cases for each query case in Task 1, and a set of relevant paragraphs for each given decision in Task 2. Hence, from the ranked lists output by our proposed methodology (as discussed in the previous section), we submitted the top $k = 5$ ranked entries for Task 1 and top $k = 1$ entry for Task 2. The values of k were chosen in accordance with the average number of gold-standard entries in the training dataset of respective tasks. In Task 1 we had an average of 5.15 noticed cases per base case so we chose $k = 5$. In case of Task 2 we had an average of 1.15 gold-standard entries so we chose $k = 1$. The methods were then evaluated by three popular set-based metrics for retrieval – Precision, Recall and F-Score. The metric calculations are detailed in the COLIEE 2020 website[9].

Performance comparison: To show the efficacy of our method, we first show the performance of the standard BM25 model over bag-of-words (BOW repre-

[9] `https://sites.ualberta.ca/~rabelo/COLIEE2020/`

Task 1 Results			
Method	**Precision**	**Recall**	**F-Score**
BM25 on filtered BOW	50.1%	51.7%	50.9%
BM25 on FIBONG	52.2%	53.3%	52.7%
Modified-BM25 on FIBONG (proposed)	52.3%	53.5%	52.9%
Task 2 Results			
Method	**Precision**	**Recall**	**F-Score**
BM25 on filtered BOW	64.1%	50.2%	56.3%
BM25 on FIBONG	66.0%	52.8%	58.7%
Modified-BM25 on FIBONG (proposed)	66.0%	52.8%	58.7%

Table 4. The results table shows performance of three methods for the two tasks of COLIEE 2020. FIBONG stands for our proposed representation Filtered-Bag-of-Ngrams. Modified-BM25 on FIBONG is our proposed method and shows improvements in performance over its predecessors.

sentation). Then we show what happens when we apply BM25 over the FIltered-Bag-Of-NGrams (FIBONG representation). We finally show the performance of our proposed best performing model, which uses FIBONG representations of documents and ranks them using the modified Scoring function given in Equation 3.

The results of this comparison are given in Table 4. First, we observe that there is a significant improvement by using the FIBONG representation instead of the BOW representation. For instance, the F-score for Task 1 improves from 50.9% (with BOW) to 52.7% (using FIBONG). Similarly the F-score for Task 2 improves from 56.3% (with BOW) to 58.7% (using FIBONG). Additionally, the modified scoring function also improves performance slightly in case of Task 1.

Thus, replacing existing models of document representation that uses the BOW representation with FIBONG representation might improve many downstream applications, without even changing the internal architecture of the model. Instead of using a regular word tokenizer we need a tokenizer that gives FIBONG representation for a given text. We have made such a tokenizer available[10].

Comparison with other COLIEE submissions: Table 5 compares the participant scores of all methods submitted to COLIEE 2020, as provided by the COLIEE organizers. In this case, we are limited to just showing the F-Score values (as provided by COLIEE organizers). Our unsupervised method (highlihted in boldface in Table 5) achieved the 10^{th} position in Task 1 (case retrieval) and the 9^{th} position in Task 2 (case entailment).

5 Conclusion

We have presented our methods for the case retrieval task (Task 1) and the case entailment task (task 2) of COLIEE 2020. We have found that representation

[10] `https://github.com/amarnamarpan/fibong-a-new-way-of-representing-text-documents`

Task 1 Participant Scores		**Task 2 Participant Scores**	
Submission	**F-Score (%)**	**Submission**	**F-Score (%)**
task1 cyber02.txt	67.74	JNLP.task2.BMWT.txt	67.53
task1 cyber03.txt	67.68	JNLP.task2.BMW.txt	62.22
t1_run3_thuir.txt	66.82	t2-taxiXGBaft.txt	61.80
task1 cyber01.txt	65.03	t2_run2_thuir.txt	61.54
JNLP.task1.BMW25.txt	63.97	JNLP.task2.WT+L.txt	60.94
t1_run2_thuir.txt	63.79	t2-taxiXGBaf.txt	59.92
JNLP.task1.W25.txt	63.58	t2-taxiXGB3f.txt	59.17
JNLP.task1.W30.txt	62.78	task2 cyber03.txt	58.97
UB_RUN1.res	58.66	**iiest_bm26_t2_1.txt**	**58.67**
iiest_bm26_t1_3.txt	**52.88**	iiest_bm25_t2_2.txt	58.67
iiest_bm25_t1_2.txt	52.72	task2 cyber02.txt	58.37
t1_run1_thuir.txt	51.48	task2 cyber01.txt	56.00
iiest_ps_t1_1.txt	48.21	t2_run3_thuir.txt	54.95
submission2	38.00	t2_run1_thuir.txt	54.28
submission3	37.92	UA1.txt	54.25
submission1	33.88	UA2.txt	51.79
AUTIRT1R1.txt	26.58	iiest_l2v_t2_3.txt	50.67
T1 DACCO.txt	20.77	UA_translate.txt	46.47
AUTIRT1R2.txt	16.17	submission1.txt	41.07
AUTIRT1R3.txt	8.98	submission3.txt	41.07
UB_RUN2.res	5.92	submission2.txt	40.18
task1_TAXICATTFCV.txt	4.57	T2 DACCOr.txt	6.22

Table 5. F-Score values of all submissions for Task 1 and Task 2 of COLIEE 2020, sorted in descending order. Our best submissions (names starting with 'iiest') are marked in bold.

of a document plays an important role in these tasks. Our experiments show that FIBONG (Filtered-bag-of-ngrams) is a better way of representing a legal text document when compared to the bag-of-words (BOW) representation. Also a slight improvement in performance is shown by incorporating a new collection frequency based term to weigh the IDF term in the BM25 scoring function. We incorporated all these improvements to propose an unsupervised algorithm gives F-Score of 52.88% in Task 1 and F-Score of 58.67% in the Task 2 of COLIEE 2020.

In future, there are several improvements we would like to incorporate in the methods. Instead of FIBONG, we can use graph based models where the documents would be represented by a Filtered-graph-of-ngrams. Also, no form of supervision has been included in this work. We can incorporate different forms of supervision to enhance the performance of our methods. We plan to explore these directions in future.

6 Acknowledgements

The authors thank the organizers of COLIEE for providing the dataset and an opportunity to present our work. Arpan Mandal is supported by the Visvesvaraya PhD Fellowship, Ministry of Electronics and Information Technology (MeITY), Government of India.

References

1. Bhattacharya, P., Ghosh, K., Ghosh, S., Pal, A., Mehta, P., Bhattacharya, A., Majumder, P.: FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance. In: Proc. Annual Conference of the Forum for Information Retrieval Evaluation (FIRE) (2019)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
3. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 328–339. Association for Computational Linguistics (Jul 2018). https://doi.org/10.18653/v1/P18-1031
4. Kumar, S., Reddy, P.K., Reddy, V.B., Suri, M.: Finding Similar Legal Judgements under Common Law System, pp. 103–116. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
5. Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., Ghosh, S.: Measuring similarity among legal court case documents. In: Proc. Annual ACM India Compute Conference (COMPUTE). p. 1–9 (2017)
6. Mandal, A., Ghosh, K., Bhattacharya, A., Pal, A., Ghosh, S.: Overview of the FIRE 2017 IRLeD Track: Information Retrieval from Legal Documents. In: Working notes of FIRE 2017 – Forum for Information Retrieval Evaluation. pp. 63–68 (2017)
7. Mandal, A., Ghosh, K., Pal, A., Ghosh, S.: Automatic catchphrase identification from legal court case documents. In: Proc ACM Conference on Information and Knowledge Management (CIKM). pp. 2187–2190 (2017)
8. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018), http://arxiv.org/abs/1802.05365
9. Tran, V., Le Nguyen, M., Tojo, S., Satoh, K.: Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. Artificial Intelligence and Law pp. 1–27 (2020)
10. Verma, A., Morato, J., Jain, A., Arora, A.: Relevant Subsection Retrieval for Law Domain Question Answer System, pp. 299–319. Springer International Publishing (2020)
11. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries. p. 254–255 (1999)
12. Xiong, C., Callan, J., Liu, T.: Bag-of-entities representation for ranking. Proc. ACM International Conference on the Theory of Information Retrieval (2016)

JNLP Team: Deep Learning for Legal Processing in COLIEE 2020

Ha-Thanh Nguyen*[1], Hai-Yen Thi Vuong*[1,2], Phuong Minh Nguyen[1], Binh Tran Dang[1], Quan Minh Bui[1], Sinh Trong Vu[1,4], Chau Minh Nguyen[1], Vu Tran*[1], Ken Satoh[3], and Minh Le Nguyen*[1]

[1] Japan Advanced Institute of Science and Technology
[2] VNU University of Engineering and Technology
[3] National Institute of Informatics
[4] Banking Academy of Vietnam
{nguyenhathanh,phuongnm,
binhdang,quanbui,sinhvtr,chau.nguyen,vu.tran,nguyenml}@jaist.ac.jp,
yenvth@vnu.edu.vn, sinhvtr@bav.edu.vn, ksatoh@nii.ac.jp

Abstract. We propose deep learning based methods for automatic systems of legal retrieval and legal question-answering in COLIEE 2020. These systems are all characterized by being pre-trained on large amounts of data before being finetuned for the specified tasks. This approach helps to overcome the data scarcity and achieve good performance, thus can be useful for tackling related problems in information retrieval, and decision support in the legal domain. Besides, the approach can be explored to deal with other domain specific problems.

Keywords: Deep Learning · Legal Text Processing · Pretrained Legal Text Encoders

1 INTRODUCTION

COLIEE is an annual competition to find automated solutions in the field of law. This competition is challenging because legal documents are often complex and require a high level of comprehension. The problems in law are even tricky for law experts. COLIEE tasks cover two of the most popular legal systems in the world, Case law and Civil law. COLIEE provides real data from the Canadian judicial system and the Japanese legal system. COLIEE organizes 4 tasks divided into 2 categories: retrieval and entailment. For retrieval tasks, the systems need to automatically find out the supporting cases of a given query case (Task 1) or the relevant articles of a given bar question (Task 3). For the entailment tasks, the systems need to find the paragraphs in a relevant case that entail a given decision (Task 2) or to conclude whether the statement of a given question is correct or incorrect (Task 4). These tasks can be solved with various

* Corresponding authors
The works are conducted at Nguyen Lab and Interpretable AI Center - JAIST

text processing methods. On one hand, over the years, systems using only lexical similarity of texts yield inferior performance. On the other hand, deep learning approaches start gaining superior performance recently.

Case Law In COLIEE 2018, UBIRLED team use Tf-idf for ranking the candidate cases and abolish almost 75% of lowest scoring candidate cases. Almost the same approach with UBIRLED team, UA team create a system which capture lexical matching between given base case and corresponding candidate cases. This approach is applied for both Task 1 and Task 2 in COLIEE 2018 by several teams, but we can see that the performance is quite low with F1 of 0.3741 on Task 1 and 0.0940 on Task 2. JNLP team combines lexical matching and deep learning, which achieved state-of-the-art performance on Task 1 with F1 of 0.6545.

In COLIEE 2019, several teams apply machine learning including deep learning to both of the tasks. JNLP team achieved the best system for Task 1 in COLIEE 2019 [10] by using the approach similar to theirs in COLIEE 2018. In Task 2, their deep learning approach achieved lower performance compared to their lexical approach. However, team UA's combination of lexical similarity and BERT model achieved the best performance for Task 2 in COLIEE 2019 [5].

Statute Law The retrieval task (Task 3) is often considered as a ranking problem with similarity features. In COLIEE 2019, JNLP[9] chose Tf-idf as the main feature and calculated based on n-gram, noun phrase and verb phrase. DBSE[8] and IITP[1] based on BM25 to build their methods. DBSE used BM25 and Word2vec to calculate the similarity score. The KIS team[6] represents article and query as a vector by generating a document embedding vector. Keywords are selected by Tf-idf and assigned high weight in the embedding process.

Regarding the entailment task (Task 4), approaches using deep learning are attracting more attention. In COLIEE 2019, KIS[7] used predicate-argument structures to evaluate similarity. IITP[1] and TR[3] apply BERT for this task. JNLP [4] classifies each query to follow binary classification based on big data.

2 CASE LAW

2.1 Method

In legal technical terms, supporting cases in Task 1 are "noticed" legal cases, which is considered to be relevant to a query case. Relevant cases are assumed to concern similar situations. A legal case document contains case details and metadata. The contents of the legal case are presented in a list of paragraphs that can be factual statements.

In Task 2, from given base case, an entailed fragment paragraph which extract from base case and the correctly paragraph from second case which support the entailed fragment. The mission is to discover which paragraphs within the second case support the entailed fragment from the base case.

In the previous works, they usually tackle finding the supporting relationship between query-case/fragment and candidate-case/candidate-paragraph indirectly through similarity measures. Therefore, we want to build a model that could capture the supporting relationship directly.

Supporting Model We train the supporting model on a task of supporting text-pair recognition in case law. The goal of the task is to determine whether a text supports a decision. A small number of training data brings obstacles to the process of training deep neural models while collecting a large dataset for supporting relations can be challenging. Therefore, we designed some heuristics to automatically extract supporting text-pairs from case documents in Task 1's training data, and construct a "weak-labeling" dataset. The supporting model is based on the BERT base model, then finetuned on the supporting text-pair classification task. The BERT includes 768 hidden nodes, 12 layers, 12-attention heads, 110M parameters, 512 max input length.

Lexical Matching The lexical similarity and the support relationship are distinct and can be complementary. Therefore, the lexical similarity is also potential in these tasks. We use BM25 to calculate lexical similarities. In Task 1, we separate the whole base cases and also candidate cases into set of paragraphs. We calculate the BM25 score where the query in this circumstance is each paragraph in set of paragraphs for given base case, and the documents is set of paragraphs of corresponding candidate cases. In Task 2, we have the same approach with Task 1 where the query is entailed fragment and the documents is set of candidate paragraphs. We use both of lexical score and supporting score as combine score:

$$score = \alpha * score_{supporting} + (1 - \alpha) * score_{BM25} \quad (1)$$

Task 1 Deep learning models often consume much time and resources. Therefore, we use lexical scoring to filter top n cases from the given set of candidate cases and combine scores to identify supporting relations. All of the scores are calculated in paragraph level. Our supporting model does not use case-case supporting relation in the provided training data.

Task 2 We try 3 settings as follows:

1. We directly use the already trained supporting model in Task 1 together with BM25 scoring for finding the support paragraphs for given entailed fragment.
2. We enhance the system in the setting 1 by finetuning the supporting model on the training data of Task 2.
3. We replace BM25 in the setting 2 with a BERT model finetuned on SigmaLaw dataset [5].

[5] https://osf.io/qvg8s/

2.2 Task 1 Results

In COLIEE 2020, an short analysis is indicated in Table 1, every single case has more than 2.4K-token long in average, the number of paragraphs over 20. Not only about length but also the unbalance in the training and development set compare to the test set. We can see that the maximum number of words and paragraphs in test set are ten times more than training and development set.

Table 1: Task 1 Data Analysis

	train 2020	dev 2020	test 2020
Word/Doc	2462	2443	3232
Paragraphs/Doc	23	23	28
Maximum Word	10827	10827	127263
Maximum Paragraph	119	119	1139
Sample	285	61	130
Candidate case	57000	12200	26000
Average notice case	5.21	5.41	4.89

As the results in the Table 2, we can see the lexical model (BM25) only achieves 0.5297 on F1. While the supporting model (W25 and W30) surpasses the lexical model by 0.1 on F1. Combining the lexical model and the supporting model (BMW25) get the best performance in our retrieval system.

Table 2: Results on Task 1.

Method	**Hyper-parameters**	**Precision**	**Recall**	**F1**
BM25	$\alpha = 0, topn = 25$	0.4071	0.7579	0.5297
JNLP.W25	$\alpha = 1, topn = 25$	0.5323	0.7893	0.6358
JNLP.W30	$\alpha = 1, topn = 30$	0.5146	0.8050	0.6278
JNLP.BM25	$\alpha = 0.85, topn = 25$	0.5603	0.7453	0.6397

Since Task 1 is a retrieval task, we want to optimize recall. That is the reason why our models obtain as many cases as possible and achieved high recall scores (the highest was 0.8). The average number of notice case in test set 2020 is less than that number in the train set and the dev set. This leads to the unbalance between recall and precision and low performance on F1. Although we didn't use the gold label from the training set, we achieved promising results in this retrieval task.

Table 3: Task 2 Data Analysis

	train 2020	dev 2020	test 2020
Case	181	44	100
Candidate/Query	32.12	32.91	36.72
Entailed fragment/Query	1.12	1.02	1.25

2.3 Task 2 Results

From Table 4, we notice that the lexical model achieve inferior performance, only 0.5764 F1. The setting 1 using the supporting model without Task 2 gold label performs better than the lexical model. Moreover, finetuning on the Task 2 gold label in the setting 2 further improves the performance and thus achieves the top performance of 0.6753 F1. Replacing BM25 by the BERT trained on SigmaLaw, however, reduces performance.

Table 4: Result on Task 2.

Method	**Approaches**	**Precision**	**Recall**	**F1 score**
BM25	Only BM25 score	0.6346	0.5280	0.5764
JNLP.BMW	Setting 1	0.7000	0.5600	0.6222
JNLP.BMWT	Setting 2	**0.7358**	**0.6240**	**0.6753**
JNLP.WT+L	Setting 3	0.6574	0.5680	0.6094

3 STATUTE LAW

3.1 Task 3

As our observation, the challenge of this task is the long content of articles and the implication of meaning.

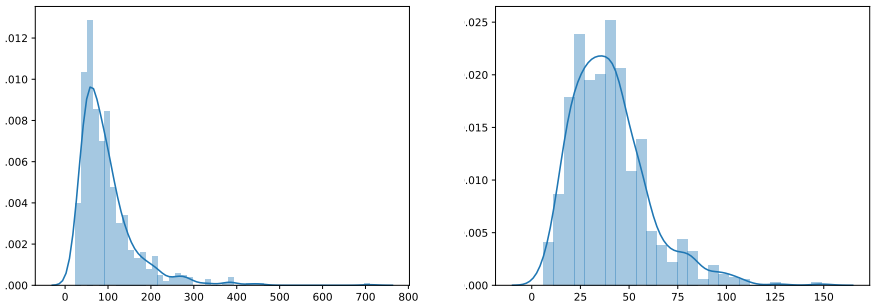
The content of the articles in some cases is a paragraph that contains many sentences. For example, the length distribution of articles and questions in the COLIEE Task 3 dataset is presented in Figure 1. The content of articles ($S_{i|1\leq i\leq n}$) is the abstract description containing the conditions and relevant actions. The content of legal bar exam question (Q) is a particular case in daily life (Table 5). Therefore, these pairs ($Q; S_i$) usually may not have much lexical overlap and it is difficult for methods using lexical matching and ranking to achieve high performance.

Recently, Transformer [11] and two-stage (pre-training + fine-tuning) learning using BERT pre-trained model [2] attracts a lot of attention because it is able to take advantage of pre-trained context before in many tasks in NLP. Therefore, in this work, we use BERT pre-trained model and fine-tune it with the

Table 5: Example of Task 3: pair of Query and Civil Code Article

Article	Id	"303"
	Part	"Part II Real Rights"
	Chapter	"Chapter VIII Statutory Liens"
	Section	"Section 1 General Provisions"
	Sumary line	"(Content of Statutory Liens)"
	Content	"The holder of a statutory lien has the rights to have that holder's own claim satisfied prior to other obligees out of the assets of the relevant obligor in accordance with the provisions of laws including this Act."
Question	Id	"H18-9-2"
	Content	"Statutory real rights granted by way of security exist, but statutory usufructuary rights do not exist."

Fig. 1: Sentence length distribution.



(a) Articles (b) Questions

pair $(Q; S_i)$ plays a role in two input sentences to classify the relevant between them. This approach changes the information retrieval problem into a classification problem that is similar to the MRPC subtask in the GLUE task [2]. Our hypothesis is that the learned knowledge in BERT pre-trained model is helpful for meaning representation of both legal bar exam question (Q) and article (S_i) and improves the weak points of the approaches using word face representation.

As we mentioned, we treat the task as classification to detect the relevant articles, so with each legal bar exam question, we need to classify each pair of the question with every article from the Civil Code. In fact, there are about 94% of the legal bar exam questions have the number of relevant articles less than 3 while the total number of considering articles is more than 750. It is so unbalanced between relevant and non-relevant labels if we use all articles for pairing with each question. To prevent the unbalance problem, we used a pre-processing method that limits the number of non-relevant articles by selecting the k (selected in fine-tune step) most appropriate articles using cosine similarity between Tf-idf vectors of the question and articles. In this way, we also reduce

the unnecessary prediction for the classification model and diminish the bad impact of the unbalance problem.

We build our system in two phases: data pre-processing and training/testing.

1. **Data pre-processing** We filter the k best appropriate articles corresponding to the legal bar question. We build a Tf-idf vectorizer based on all articles in the Civil Code. After that, we encode legal bar questions and articles into vectors and use Cosine similarity to the rank of all articles. For each filtered article, we create a pair of it with the legal bar question and provided labels from COLIEE Task 3 dataset. Besides, to choose the k value we do a small experiment to check the recall score of this step (Table 6). Based on this result, we choose $k = 150$ for the training/testing phase.

Table 6: The dependency between k and recall score.

k	30	50	70	100	120	**150**
Recall (%)	79.51	83.72	85.93	88.59	89.81	**91.25**

2. **Training/testing** In the training phase, we used BERT pre-trained model BERT-base-uncased) for fine-tuning data generated in the previous step. Besides, we assume that the BERT-base-uncased has strong knowledge in the general domain but weak in the legal domain. Therefore, we used all data from the Civil Code to fine-tune a special BERT model (BERT-CC) using the mask-language-model (MLM) [2] and fine-tune the BERT-CC model for measuring relevance. Our hypothesis is that the BERT-CC and BERT-base-uncased learned different linguistic features, so the ensemble of these models can improve the recall score as well as performance.

Experiments

From the raw text file Civil Code, we used pattern matching to parse the structure of each article (example in Table 5). To evaluate the performance of the model and optimize hyper-parameters, we separated a COLIEE Task 3 Questions dataset into two subsets train and dev. The dev subset contains about 10% of the total that is all Questions having id field start with "H29-*-*". This subset plays a role as a development set to optimize hyper-parameters. To get the final submit results, we used the best setting found based on the dev subset and re-train on all original COLIEE Task 3 dataset.

Firstly, we conduct experiments to find the best setting of some hyper-parameters such as training epochs, max sequence length, etc. (Table 7).

- The runs 1, 2, 3 show that if we increase the number of training epochs, the model tends to increase the Precision score and decrease the Recall score.
- The runs 3, 4, 5 show a similar variation, if we increase max sequence length, the model tends to increase the Precision score and decrease the Recall score.

Based on the F2 score, we found the best hyper-parameters setting is in "Run 3".

Table 7: Optimize hyper-parameters of BERT-base-uncased on dev subset.

Run	Hyper-parameters	Prec	Recall	F2
1	epochs=10; max_senquence_len=230; lr=$2e^{-5}$	0.5857	0.5694	0.5726
2	epochs=5; max_senquence_len=230; lr=$2e^{-5}$	0.5211	0.5139	0.5153
3	epochs=3; max_senquence_len=230; lr=$2e^{-5}$	0.5500	**0.6111**	**0.5978**
4	epochs=3; max_senquence_len=300; lr=$2e^{-5}$	0.5270	0.5735	0.5636
5	epochs=3; max_senquence_len=400; lr=$2e^{-5}$	**0.6000**	0.4853	0.5046
6	epochs=3; max_senquence_len=230; lr=$1e^{-5}$	0.5417	0.5417	0.5417

Secondly, we conduct experiments to compare performance of different methods/models (Table 8).

- **Tf-idf**. This is the simple method using cosine to calculate the similarity of Question Tf-idf vector (Q) with each Article Tf-idf vector (S_i) and choose top two articles having the best scores.
- **BERT-base-uncased (regression)**. In this run, we try to learn a regression model based on BERT-base-uncased. The results are the best performance after we optimized threshold values.
- **BERT-base-uncased**. This run used BERT-base-uncased pretrained model to fine-tune.
- **BERT-CC**. As mentioned before, this model we fine-tune a language model with MLM task from BERT-base-uncased and continue to fine-tune to detect relevant articles.
- **BERT-ensemble**. In this model, we ensemble the result of 2 models: BERT-base-cased and BERT-CC, the article is relevant to the legal bar question if one of these models predicts True.

These results show that the BERT-CC learned more information in the legal domain as our expectation with the highest precision. Besides, the BERT-ensemble can utilize the result of BERT-CC and BERT-base-uncased to improve the performance (recall and F2) of the overall system.

Table 8: The comparasion performance of models on dev subset.

Model	Prec	Recall	F2
Tf-idf (Top 2 best articles)	0.3276	0.5278	0.4703
BERT-base-uncased (regression)	0.4819	0.5882	0.5634
BERT-base-uncased	0.5500	0.6111	0.5978
BERT-CC	**0.5694**	0.6029	0.5959
BERT-ensemble	0.5111	**0.6389**	**0.6085**

Performance on the test data The performance on test set is 2nd in the COLIEE-2020 competition, showed on Table 9. Similar to the result on the dev subset, the model BERT-ensemble achieves the best result in comparing with BERT-CC and BERT-base-uncased. We show the improving predictions of

Table 9: Task 3 - final result on test set

Run	Prec	Recall	F2	MAP	R5	R10	R30
JNLP BERT-ensemble	**0.5766**	**0.5670**	**0.5532**	0.6618	0.6857	0.7143	0.7786
JNLP BERT-CC	0.5387	0.5268	0.5183	0.6847	0.7143	0.7714	0.7929
JNLP BERT-base-uncased	0.5409	0.5268	0.5149	0.6847	0.7143	0.7714	0.7929

BERT-CC when comparing with BERT-base versions in Table 10. When comparing with BERT-CC, BERT ensemble version predicts more than 20 relevant pairs of Question and Article but there are only 7 correct relevant pairs learned by BERT-base. When compare with BERT-base, BERT ensemble version predicts more than 7 relevant pairs and there are 5 correct relevant pairs that learned by BERT-CC. We hypothesis that the BERT-CC model is different from BERT-base to pay attention to the different important words in a sentence with higher accuracy.

Table 10: Improving predictions of BERT-CC missed by BERT-base.

Pair	Content
Q: *R01-1-E* A: *17*	An ... such consent or the permission of family court in lieu thereof. Part I General Provisions ... Requiring Person to Obtain Consent of Assistant) ... obtain the consent of the person's assistant ...
Q: *R01-3-O* A: *117*	When An unauthorized agent acted believing that ... ... Section 3 Agency (Liability of Unauthorized Agency) (1) A person who concludes ...
Q: *R01-10-O* A: *254*	A, B and C co-own Building X at the rate of one-third each. ... ... Ownership Section 3 Co-Ownership (Claims on Property in Co-Ownership) A claim that one of the co-owners ...
Q: *R1-20-I* A: *484*	In cases where the seller of the specified things retains them at the place ... the tender of the performance shall be sufficient ... ... Unless a particular intention is manifested with respect to the place where the performance should take place ...

3.2 Task 4

The goal of Task 4 is to find answers to given bar questions. Bar questions are the questions to judge paralegal examinees, making sure they are qualified to

practice legal activities. The reasoning sequence of an examinee when perceiving a question is as follows: The examinee reads and understands the question, delineates the scope of relevant legal knowledge, finds the articles that reject or support the proposition in question and decide the final answer. There are two possible outcomes, Yes and No, which the system has to decide on for each question.

A system designer's natural thought is to use the result of Task 3 as input to Task 4 in the form of a Texture Entailment problem. However, this approach may cause the outcome of Task 4 to be biased into the selection of relevant articles in Task 3.

Also, thinking about the paralegal's process of finding answers in the bar exam, we suppose that the examinee's thinking process might be different from conventional system design. A good examinee would not try to remember every word in the civil code to conduct inference. They will try to abstract out the principles in the law as well as situations they have learned earlier.

From the above observation, in terms of system design, we use two approaches: using the results of Task 3 together with the questions as input to the texture entailment problem and using only the content of questions as inputs of the classification problem. With the second approach, our hypothesis is that deep learning systems can automatically generate latent patterns at an abstract level to answer questions.

In the first approach, we use BERT-base-uncased to implement texture entailment. The question was linked to the relevant article by a *[SEP]* token and fed into BERT model. On our own development set, we found that using gold data with extra relevant articles extracted by Tf-idf directly from Civil Code brings better performance than using only gold data from Task 3. For each question, we applied the textual entailment with a new data named *Tf-idf extra data* that contains gold data and 2 top articles returned by *Tf-idf*. The system decides the answer is Yes if at least one pair is classified as positive textual entailment.

The second approach is firstly proposed by Nguyen et al. [4] in COLIEE-2019. By converting the texture entailment problem into the lawfulness classification problem, the data used to build the model becomes more abundant. In this approach, the model is trained directly from law sentences, bar questions, and their variants with two labels: Yes and No. Data sources are the Civil Code and previous years' bar questions provided by the COLIEE organizers.

We use two variants of BERT in this approach, pretrained BERT-base-uncased by Google and BERT Law trained from scratch by a corpus containing 8.2M sentences (182M words) from American case data in English.

Both variations contain 768 hidden nodes, 12 layers, 12-attention heads, 110M parameters. They are then finetuned and validate on the dataset, including their law sentences, bar questions, and their augmented samples. The maximum length in our setting is 128.

These two vocabulary sets share only 12,853 tokens. 17,669 tokens appear only in BERT Base vocabulary and 19,146 tokens appear only in BERT Law vocabulary. This is an interesting figure showing the difference in how the two

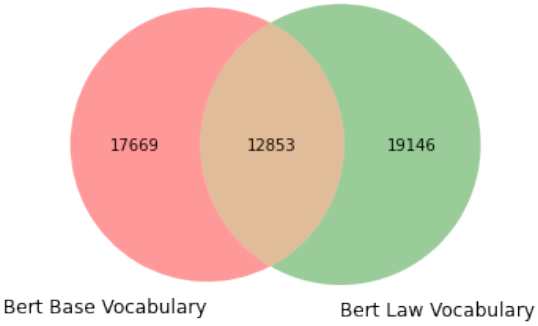


Fig. 2: The cardinality of two vocabulary sets of Bert Base and Bert Law.

models of the same architecture learn to use vocabulary in the general domain (BERT Base) and the specialized domain (BERT Law).

Textual Entailment Approach In the textual entailment approach, we conduct an experiment to choose the more suitable source of relevant articles: original gold data of Task 3 or Tf-idf extra data. The dataset to train and validate the model is the question bars from the previous years provided by the organizers. The development set is 10% of the entire dataset. Table 11 presents the performance on the two sources.

Table 11: Using relevant articles retrieved by *Tf-idf* yeilds better performance

Data Source	**Accuracy**
Gold data of Task 3	0.5278
Tf-idf extra data	**0.5417**

Using the *Tf-idf extra data* yields 1.39 % higher than the result from the gold data of Task 3. This result interestingly suggests the possibility that there exists a set of relevant articles other than gold data of Task 3 that can aid in deciding answers to bar questions. This is consistent with the actual phenomenon that occurs in the litigation process, lawyers can come up with different arguments to defend or reject the same opinion.

Lawfulness Classification Approach Compared with the Textual Entailment approach, the Lawfulness Classification approach has much more abundant data. All augmented data from law sentences and previous years' bar questions

are 5000 samples. We also use 10% of the dataset as the development set. Table 12 shows the performance of BERT Law and BERT-base-uncased in the development set.

Table 12: BERT Law overperforms BERT Base

Model	**Accuracy**
BERT Base	0.7784
BERT Law	**0.8168**

Table 11 and Table 12 report results evaluated on two different development sets of two different problems. The conclusions we draw from the above two tables are that: on our development set, using data from retrieved from *Tf-idf* will be better in the Textual Entailment problem and using BERT Law will be better in Lawfulness Classification problem. These are important information for us to choose the candidates for our final runs.

Performance on the gold data Our results on gold data of Task 4 are shown in the Table 13. This table of results shows that the second approach outperforms the first. In addition, BERT Law creates a significant gap compared to Google's pretrained BERT-base-uncased. The model comparison result is consistent with our experimental results on our development set.

Table 13: Final runs result on gold data of Task 4

Run	**Correct**	**Accuracy**
JNLP BERT Law	81	**0.7232**
JNLP BERT Base	63	0.5625
JNLP Tf-idf BERT Base	62	0.5536

Discussion According to our observations, the test data for this task in previous years are usually small, leading to significant variability of the same model from the development set to the test set and from year to year. Therefore, choosing the best model based on a development set does not guarantee it will work well on test data. Despite this, the COLIEE Task 4 results this year brought us by surprise with the fact that BERT Law made a big gap compared to other models.

As our observation from the experimental results, pretrained deep learning models can perform well on comprehensive tasks like law. And it achieves good

results with the right training methods as well as good data in both quantity and quality.

4 CONCLUSION

In this paper, we reported our methods and results in using different techniques with Deep learning for the COLIEE 2020 comprehensive legal text processing tasks. Our approaches are based on trained deep learning models with large amounts of data to support decisions for retrieval and entailment problems. Experimental results show that our systems yield competitive performance.

The results demonstrate that it is feasible to use the deep learning models for advanced tasks such as law document processing as long as the appropriate approach is taken. In future works, we will investigate to build different models with high stability, applicable to real-life applications.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR1513 and JSPS KAKENHI Grant Number JP17H06103. Vuong and Chau were supported in part by the Asian Office of Aerospace R&D (AOARD), Air Force Office of Scientific Research (Grant no. FA2386-19-1-4041)

References

1. B.Gain, D.Bandyopadhyay, T.Saikh, A.Ekbal: Iitp@coliee 2019: Legal information retrieval using bm25 and bert. (2019)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
3. J.Hudzina, T.Vacek, K.Madan, C.Tonya, F.Schilder: Statutory entailment using similarity features and decomposable attention models. (2019)
4. Nguyen, H.T., Tran, V., Nguyen, L.M.: A deep learning approach for statute law entailment task in coliee-2019. (2019)
5. Rabelo, J., Kim, M.Y., Goebel, R.: Combining similarity and transformer methods for case law entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. p. 290–296. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3322640.3326741, https://doi.org/10.1145/3322640.3326741
6. R.Hayashi, Y.Kano: Searching relevant articles for legal bar exam by doc2vec and tf-idf. (2019)

7. R.Hoshino, N.Kiyota, Y.Kano: Question answering system for legal bar examination using predicate argument structures focusing on exceptions. (2019)
8. S.Wehnert, S.A.Hoque, W.Fenske, G.Saake: Threshold-based retrieval and textual entailment detection on legal bar exam questions. (2019)
9. T.B.Dang, T.Nguyen, L.M.Nguyen: An approach to statute law retrieval task in coliee-2019. (2019)
10. Tran, V., Nguyen, M.L., Satoh, K.: Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 275–282 (2019)
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Application of Text Entailment Techniques in COLIEE 2020

Juliano Rabelo[1,3][0000−0002−2982−3401], Mi-Young Kim[1,2], and Randy Goebel[1,3]

[1] Alberta Machine Intelligence Institute, University of Alberta, Canada
[2] Dept. of Science, Augustana Faculty, University of Alberta, Canada
[3] Dept. of Computing Science, University of Alberta, Canada
{rabelo, miyoung2, rgoebel}@ualberta.ca

Abstract. We try to identify entailment relationships in the texts of case law documents, in the context of two tasks of the Competition on Legal Information Extraction and Entailment (COLIEE 2020). The first task consists in, given a 1) base case, 2) a text fragment from that base case, and 3) the list of paragraphs from a noticed case, identify which of those paragraphs entail the given text fragment. For that problem, we apply a combination of transformer-based and textual similarity techniques. The second task consists in identifying entailment relationships between yes/no questions extracted from the Japanese law bar exam and paragraphs extracted from the Japanese statute legislation. Our approach is to first define a set of classes of textual entailment that cover our analysis of the statute law data, and then implement a pre-processing step to construct the statute law-specific entailment types before exploiting them with deep learning.

Keywords: Legal textual entailment · Binary classification · Imbalanced datasets.

1 Introduction

Tools to help legal professionals handling the large volume of legal documents are becoming more and more necessary. The amount of information produced in the legal sector by its many different actors (such as law firms, law courts, independent attorneys, legislators, and many other sources) is overwhelming. To help build a legal research community, COLIEE was created, and focuses on four specific problems in the legal domain: case law retrieval, case law entailment, statute law retrieval and statute law entailment. In this paper, we provide details of our approaches for the two entailment tasks.

Initial techniques for open-domain textual entailment focused on shallow text features, and evolved to the usage of word embeddings, logical models and machine learning in general. The current state of the art, especially for problems which have access to enough labelled data, rely on deep learning based approaches (more notably those based on transformer methods), which have shown very good results in a wide range of textual processing benchmarks, including benchmarks specific to entailment tasks.

Our method applied for the task of case law entailment presented builds on our methods in the past two editions [26], [27]: it combines similarity based features, transformer methods and a simple post processing technique based on *a priori* probabilities. For the similarity calculations, we now used two methods: a sentence distributed vector representation and a simple bag of words representation using noun phrases as tokens. The cosine measure was used to calculate a similarity documents represented as described above.

The transformer methods relied on the BERT framework[10]: one by fine-tuning a pre-trained BERT model text entailment on the provided training dataset, another one leveraging two BERT models fine tuned on a generic entailment data set, then another one applying zero-shot techniques by using BERT fine tuned for paraphrase detection. We also used data augmentation techniques based on back translation to increase the size of the training data.

The rest of this paper is organized as follows: in Section 2 we briefly review open domain textual entailment and the special cases of case law entailment and statute law entailment. Section 3 describes our approaches to both case law and statute law entailment tasks in COLIEE 2020. Section 4 describes our experiments with an analysis of the results. Section 5 concludes the paper and comments on future work.

2 Related Work

Textual entailment is a logic task in which the goal is to determine whether one sentence can be inferred from another. In COLIEE, teams are challenged with the task of classifying two case law textual fragments possessing a "positive entailment" relationship or not (i.e., either they have "positive entailment" or "neutral entailment"). The statute law entailment task (Task 4) in COLIEE is the same: the participants are required to decide if a query is entailed from the relevant civil law articles or not.

In the following subsections, we will discuss related research on textual entailment in general and techniques developed specifically for case law entailment.

2.1 Open-domain Textual Entailment

Textual entailment is useful as a task per se or as a component in larger applications. For example, question-answering systems may apply textual entailment techniques to identify an answer from previously stored databases [3]. Textual entailment may also be used to enhance document summarization (e.g., being used to measure sentence connectivity or as additional features to summary generation [21]). Due to the recently increased interest on textual entailment, publicly available benchmarks exist to evaluate such systems (e.g., [4], [31]).

Early approaches for open-domain textual entailment relied heavily on exploiting surface syntax or lexical relationships, and then a wide range of tools, such as word embeddings, logical models, graphical models, rule systems and machine learning were applied [1]. A modern research trend for open-domain

textual entailment is an application of general deep learning models, such as ELMo [25], BERT [10] and ULMFit [14].

These methods build on the approach introduced in [9], which showed how to improve document classification performance by using unsupervised pre-training of an LSTM [13] followed by supervised fine-tuning for specific downstream tasks. The pre-training is done on very large datasets, which do not need to be labeled and are intended to capture general language knowledge (usually, the pre-training is considered as a language modeling task). Subsequently, supervised learning is used as a fine-tuning step, thus using a significantly smaller labeled dataset, aiming at adjusting the weights of the final layers of the model and making it suitable for the specific task. These models have achieved impressive results in a wide range of publicly available benchmarks of different common natural language tasks, such as RACE (reading comprehension) [17] , COPA (common sense reasoning) [30] and RTE (textual entailment) [8], to name a few.

2.2 Case Law Textual Entailment

The specific task of assessing textual entailment for case law documents is quite new. The first COLIEE edition which included this task was in 2018 [15]. Chen et al. [7] proposed the application of association rules for the problem. They applied a machine learning-based model using Word2Vec embeddings [22] and Doc2Vec [18] as features. This approach faces two main problems: the lack of sufficient training data to make the models converge and generalize, and the computational cost of training, which increases exponentially on the size of the dataset. To overcome that issue, they proposed two association rule models: (1) the basic association rule model, which considers only the similarity between the source document and the target document, and (2) the co-occurrence association rule model, which uses a relevance dictionary in addition to the basic model. Another approach [26] worth mentioning approached the task as a binary classification problem, and built feature vectors comprised of the measures of similarity between the candidate paragraph and (1) the entailed fragment of the base case, (2) the base case summary and (3) the base case paragraphs (actually a histogram of the similarities between each candidate paragraph and all paragraphs from the base case). Those feature vectors are used as input to a Random Forest [5] classifier. To overcome the problem of severe data imbalance in the dataset, the dominant class was under-sampled and the rarer class was over-sampled by SMOTE sample synthesis [6]. In [27], the method for case law entailment combines similarity based features which rely on multi-word tokens instead of single words, and exploits the BERT framework [10], fine-tuned to the task of case law entailment on the provided training dataset.

2.3 Statute Law Textual Entailment

In addition to the case law entailment task, past editions of COLIEE included a task on statute law entailment, whose goal is to identify entailment relationships between Japanese bar exam questions and relevant legal statute articles. The

best performance on that task for all past COLIEE editions has been achieved by a combination of legal information retrieval and textual entailment approach, which exploits semantic information using a logic-based representation [16]. In that approach, a meaning extraction process uses a selection of features based on a kind of paraphrase, coupled with a condition/conclusion/exception analysis of articles and queries, while additionally exploiting negation patterns extracted from the articles. The logic-based representation is then constructed as a semantic analysis, which is used to classify questions according to their difficulty level by analyzing the logic representation. If a question is in the "easy" category, the entailment answer is obtained in a straightforward manner from the logic representation; otherwise, an unsupervised learning method is applied.

3 Text Entailment in COLIEE 2020

3.1 Case Law Entailment

The task of case law entailment in COLIEE may be defined as follows: given a base case b and one fragment of text f contained within b, and a second case r which is relevant in respect to b, the task consists in determining which paragraph(s) of r entail f. More formally, given b, f and r as above (r represented by its paragraphs $P = \{p_1, p_2, ..., p_n\}$), we need to find the set $E = \{p_1, p2, ..., p_m \mid p_i \in P\}$ where $entails(p_i, f)$ denotes a relationship which is true when $p_i \in P$ entails the fragment f.

We frame that problem as a binary classification problem, by considering each paragraph p_i in r an entailment candidate, which must then be classified as entailing f or not. To do so, we created a classifier which processes each pair (p_i, f) and uses as features:

- Score from a BERT model fine-tuned with the COLIEE training data;
- Scores from BART and Roberta (also transformer-based models) fine-tuned to a different text entailment dataset;
- Score from a transformer-based model fine-tuned for paraphrase detection;
- Distance-based features between embedding representations of p_i and f;
- Distance-based features between bag of words representations of p_i and f which only take into consideration noun phrases.

These features are then fed to a Random Forest classifier, which provides the final classification score for each (p_i, f). Those scores are post processed before the final output is produced[4]. Details on each one of these components will be provided in the next subsections.

[4] We also tried data augmentation to generate more training samples, but due to computing and time constraints we could not generate all features for the augmented dataset and had to combine the output of the BERT model fine-tuned on that augmented dataset with the original model through simple heuristic rules.

BERT model fine-tuned on the COLIEE training data The main component of our method applies BERT [10] by fine tuning it on the provided traning dataset. BERT is a framework designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This leads to pre-trained representations which can be fine-tuned with only one additional output layer on downstream tasks, such as question answering, language inference and textual entailment, but without requiring task-specific modifications. BERT has achieved impressive results on well-known benchmarks such as GLUE [31], MultiNLI [32] and MRPC [11].

We used a BERT model pre-trained on a large (general purpose) dataset (the goal being make it acquire general language "knowledge"[5]) which can be fine-tuned on smaller, specific datasets (the goal being to make it learn how to combine the previously acquired knowledge in a specific scenario). This makes BERT a good fit for this task, since we do not have a large dataset available for training the model [6]. Our BERT model is then fine-tuned on the COLIEE training dataset, receiving as inputs pairs of entailment fragment f and candidate paragraph p_i, and outputting whether there is an entailment relationship or not.

Transformer models fine-tuned for text entailment on different datasets Even if one cannot immediately apply a general text entailment classifier in COLIEE (recall the difference between COLIEE's case law entailment task and the usual text entailment tasks, as explained in section 2, it is possible to easily adapt a system able to identify the 3 usual entailment classes to the COLIEE task. One immediate approach would be to consider contradiction and neutral to represent the "not entailment" class and combine those scores somehow (e.g., averaging or summing them up). In our approach, we used the 3 scores output by the entailment model (actually, since the models we used apply a softmax layer on its outputs, the scores sum up to 1 for each sample, so we discard the neutral score and keep the entailment and contradict scores, since for machine learning purposes the neutral score would be redundant). We used two transformer based models which were fine tuned on external textual entailment datasets:

- one based on BART [19], which uses a standard seq2seq architecture with a bidirectional encoder (like BERT [10]) and a left-to-right decoder (like GPT [28]). BART is commonly used for text generation text but also presents good results for text "comprehension" tasks.
- another one based on RoBERTa [20], which builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.

[5] Calling the kind of representations learned by BERT (or any other transformer based model) "knowledge" is a stretch and even some sort of anthropomorphization but, well, it seems to be appropriate in the context of machine "learning".

[6] As it is explained before, we tried to enlarge the training dataset through data augmentation techniques, since even though BERT is capable of grasping part of the correlations from the training dataset, it can certainly benefit from larger datasets.

BERT model for paraphrase detection A BERT model fine tuned on the MRPC paraphrase detection dataset [11], under the assumption that sometimes an entailment relationship consists of paraphrasing.

Distance-based features between sentence embedding representations Distributed vector representations [22] are a great tool to represent semantics of isolated words. In [26], we applied those word embeddings to larger text fragments by using the Word Mover's Distance, but that is quite expensive and does not capture the full semantics of those larger text fragments. In an attempt to generate a more accurate representation for our inputs f and p_i, we applied a sentence embedding technique, which creates distributed vector representations for sentences, thus theoretically capturing their semantics more appropriately. To apply the sentence embedding model, we first segmented the input text fragments into sentences and calculated the cosine distances [2] between each sentence from f and p_i. Out of that set of measurements, we extracted some distance based statistics (average, standard deviation, minimum and maximum distances), which were then used as features in the final classifier.

Distance-based features considering only noun phrases A classic technique for measuring similarity between documents is the representation of the text as a bag of words, and then calculating the cosine distance between those document representations in the vector space [2]. Often the text tokenization considers each word as a token (i.e., punctuation marks and spaces are seen as token delimiters), thus completely neglecting the possibility that sentences formed by the same words in different order may have different meanings. n-grams is one of the usual techniques which can be used to tackle that problem, but here we experimented with a different option: we detect the noun phrases [7] and use those as tokens. Then we created a bag of words representation of the entailed fragment f and each paragraph p_i then calculated the distance between the text fragments using the cosine distance. The similarity score generated by this process is used as a feature in the end Random Forest classifier.

Data Augmentation Being aware that the provided training data in COLIEE is not large enough to satisfactorily inform transformer-based models, we performed data augmentation by applying back translation on the training dataset using two separate pipelines with different intermediate languages. The first pipeline used a convolutional based model [12] for English $\rightarrow$ German $\rightarrow$ English translation. The second one applied a transformer based model [23] for back translating with Russian as the intermediate language. However, due to time constraints, we could not generate all features for the augmented dataset. Thus, the produced BERT model fine-tuned on the augmented training dataset and the model which combined all the features trained on the original training

[7] The actual implementation detects noun chunks, a good enough approximation. See https://spacy.io/ for more details.

dataset were applied separately to the test dataset. We changed the response of the original model by applying simple heuristic rules, which consists of changing the score produced by the full featured model if it was not too high (< 0.8) and the corresponding score produced by the augmented BERT model was very high (> 0.9). Our hypothesis was that we could take advantage of the scores produced by a separate model when that model presented a high confidence and the original model was not too sure.

Final Classifier and Post-Processing At the end of the pipeline, all the calculated scores are used as features and fed to a Random Forest [5] classifier. The confidence score output by this classifier is then post processed as such: by analyzing the *a priori* probabilities of the dataset, we see that the majority of the cases have exactly one entailing paragraph among all candidates. So we establish that, for each case, we will return at least one candidate, even if its confidence score is lower than the threshold. Moreover, we also establish that at most 2 answers should be returned for each case, no matter how many candidates have confidence scores higher than the threshold.

3.2 Statute Law Entailment

Textual entailment, which is also called Natural Language Inference (NLI), is an important problem in language understanding, and we need a system that can learn various semantic fragments such as Boolean coordination, quantification, conditionals, and monotonicity reasoning. [29] said the state-of-the-art models, including BERT, that are pre-trained on existing NLI benchmark datasets perform poorly on these semantic fragments, even though these phenomena are central to the natural language inference.

Table 1: NLI types in the statute law entailment

Type	**Civil law article**	**Query**	
Example case	If a manager engages in benevolent intervention in another's business in order to allow a principal to escape imminent danger to the principal's person, reputation, or property, the manager is not liable to compensate... (omitted)	In cases where an individual rescues another person from getting hit by a car by pushing that person out of the way, causing the person's luxury kimono to get dirty, the rescuer does not have to compensate damages for the kimono.	Y

Background knowledge required	A juridical act performed by an adult ward is void-able;provided, however, that this does not apply to the purchase of daily necessities or to any oth-er act involved in day-to-day life.	In cases where an adult ward receives the gift of a building, the adult ward cannot rescind the relevant gift contract.	N
New legal term	If the land and a building on that land belong to the same owner, a mortgage is created with respect to that land or building, and the en-forcement of that mortgage causes them to belong to dif-ferent owners, it is deemed that a su-perficies has been created with respect to that building.In this case, the rent is fixed by the court at the re-quest of the parties.	Statutory real rights granted by way of security exist, but statutory usufructuary rights do not exist.	N
'only if' con-dition	The principal secured by a revolving mortgage is crystallized in the following cases:. . . (omitted). . . pro-vided, however, that this provision applies only if the commencement of either auc-tion procedures or execution procedures against earnings from immovable collateral, or an attachment has been effected;	Even if procedures for auc-tions petitioned by a lower rank security interest holder have begun, since the first rank -revolving mortgagee has priority over the lower rank security interest holder regarding selection of the tim-ing of the auction, the first rank revolving mortgagee can stop the auction procedures without fixing the principal.	N
Connection to the previous paragraph	In order to effect an assign-ment under the provisions of the preceding paragraph, the approval of the person that holds the rights for which that revolving mortgage is the subject matter must be obtained.	Since a revolving mortgage before the fixing of principal can be described as a right which dominates the value of the maximum amount detached from the secured claim, it can be transferred in whole or in part, but since the obligor and the secured claim may change, approval must be obtained from the re-volving mortgagor.	Y

Exceptional case	...(omitted) provided, however, that this does not apply if the third party knew or did not know due to negligence that the other person has not been granted the authority to represent.	a person who has manifested to a third party that he/she has granted certain authority of agency to other person(s) shall be relieved of his/her liability of apparent authority, if there is any proof that the third party knew or was negligent that the other person had not been granted authority of agency.	Y
Addition of conditional	A juridical act performed by an adult ward is voidable; provided, however, that this does not apply to the purchase of daily necessities or to any other act involved in day-to-day life.	In cases when the adult ward performs juristic acts other than acts related to everyday life, even if the adult ward obtains approval from the guardian of the adult in advance for the relevant juristic act, the adult ward can rescindthe juristic act.	Y
Relations between new entities	Except in cases prescribed in Article 438, Article 439, paragraph (1), and the preceding Article, any circumstances which have arisen with respect to one of the joint and several obligors is not effective in relation to other joint and several obligors;	When jointly and severally liable guarantor person C acknowledges a claim from person A to person B before the prescription period has elapsed, the effect of interruption of prescription also affects main obligor person B.	N
Reference to other articles	Except in cases prescribed in Articles 438, Article 439, paragraph (1), and the preceding Article, any circumstances which have arisen with respect to one of the joint and several obligors is not effective in relation to other joint and several obligors;	When jointly and severally liable guarantor person C acknowledges a claim from person A to person B before the prescription period has elapsed, the effect of interruption of prescription also affects main obligor person B.	N

Different condition, same conclusion	A person that commences the possession of movables peacefully and openly by a transactional act acquires the rights that are exercised with respect to the movables immediately if the person possesses it in good faith and without negligence.	A seller that cancelled a movable sale contract on the basis of duress can demand return of the movable based on its ownership from the person who bought the movable from the buyer prior to the cancellation without knowledge or any negligence.	N

In addition to these phenomena, we have found more NLI types in the statute law textual entailment dataset. Some of the semantic phenomena came from the characteristics of the law field. Table 1 shows the category of the found semantic fragments in statute law NLI.

According to the types of the NLI in the statute law textual entailment as shown in Table 1, we need to recognize condition, conclusion, and exceptional case. Also, we need to re-generate the conclusion of the exceptional case by taking the negation of the conclusion of the general case. References to other articles or previous paragraphs also need to resolved.
To solve the problem of the small training data size, we use the SNLI [8] dataset, a standard benchmark data in NLI. Here, we see another challenge: while SNLI dataset has one sentence for each premise and hypothesis, our COLIEE dataset has an unbalanced length between the two input texts: premise (statute law, here relevant civil law article) and hypothesis (query). While a query usually consists of one sentence, a civil law article consists of many paragraphs. Even some queries have multiple civil law articles as input. This unbalanced length can be an issue when we use an attention model to align between the two inputs. To make the alignment easier, we choose the most relevant sentence with the query amongst many sentences and paragraphs in the relevant civil law articles. To help the detection of various NLI types in statute law textual entailment, we use the following pre-processing step:
(1) Split a civil law piece into condition, conclusion and exceptional case by [16].
(2) Referring to preceding paragraphs or cases of other articles is replaced by the corresponding paragraphs or case descriptions in other articles.
(3) Re-generate a sentence for the exceptional case by adding the negated conclusion of the general case.
(4) Extract only one sentence relevant to a query that shares the most terms with the query.

We submitted the results of following three models in COLIEE 2020:
(1) a decomposable attention model [24], which is a simple neural architecture for natural language inference. This model is decomposing a problem into sub-

[8] https://nlp.stanford.edu/projects/snli/

problems that can be solved separately. The approach consists of the 'Attend', 'Compare', and 'Aggregate', and outperformed considerably more complex neural methods aiming for text understanding.
(2) RoBERTa, a robustly optimized BERT pretraining approach [20]. This model modified the original BERT to measure the impact of many key hyperparameters and training data size. Their modifications are (a) training the model longer, with bigger batches, over more data, (b) removing the next sentence prediction objective, (c) training on loner sequences, and (d) dynamically changing the masking pattern applied to the training data. This model established a new state-of-the-art on NLI over the original BERT.
(3) [16]'s approach that showed the best performance in COLIEE 2019.

4 Results

4.1 Case Law Entailment

The official COLIEE 2020 results for this task are shown in table 2. Our two best submissions (0.45 and 0.51 f1 scores) were based on our fine tuned BERT model and use the full set of additional features. The only difference between them is the hyperparameters used while training the BERT model: the first one was trained for 5 epochs and the second one for 3 epochs. Our worst submission (f1 score of 0.46) consists of the BERT model trained for 5 epochs combined with a separated BERT model fine tuned for 1 epoch on the augmented training dataset, as explained in section 3.1. As it can be seen from table 2, the applied heuristics actually compromised the model performance.

Table 2. Official COLIEE 2020 results on the test dataset.

Team	**F1**	**Team**	**F1**	**Team**	**F1**	**Team**	**F1**	**Team**	**F1**	**Team**	**F1**
JNLP	0.6753	JNLP	0.6094	iiest	0.5867	TLIR	0.5495	iiest	0.5067	TR	0.4018
JNLP	0.6222	taxi	0.5992	iiest	0.5867	TLIR	0.5428	**UA**	**0.4647**	DACCO	0.0622
taxi	0.6180	taxi	0.5917	cyber	0.5837	**UA**	**0.5425**	TR	0.4107		
TLIR	0.6154	cyber	0.5897	cyber	0.5600	**UA**	**0.5179**	TR	0.4107		

4.2 Statute Law Entailment

Table 3 shows the comparison of the accuracy between applying the pre-processing step and without the pre-processing step. For this experiment, we used 70 pairs out of the training dataset as the validation data. Through this result, we can conclude that applying the pre-processing step is helpful improving the NLI performance in the statute law entailment.

In Task 4 of COLIEE 2020, the training data consists of 13 xml files with 696 pairs of <query, relevant civil law article(s)>, and the test data consists of 112 pairs. Table 4 shows the results of the statute law entailment competition. Our

Table 3. Comparison of accuracy according to the usage of a pre-processing step

	model	Validation accuracy
Without using our pre-processing step	Attention	0.486
	RoBERTa	0.457
Using our pre-procesisng step	Attention	0.657
	RoBERTa	0.629

two models (attention and RoBERTa) were ranked No. 2, and the third model was ranked No. 9 amongst the 30 submissions.

Table 4. Official COLIEE 2020 results on Task4

Team	Acc.	Team	Acc.	Team	Acc.
JNLP.BERTLaw	0.7232	linearsvm.HONto	0.5625	CUGIVEN	0.5179
TRC3mt	0.6250	JNLP.BERT	0.5625	CUPLUS	0.5179
TRC3t5	0.6250	KIS	0.5625	linearsvm-no-ngram-nofuzz.HONto	0.5089
UA-attention-final	**0.6250**	linearsvm-no-ngram.HONto	0.5536	POS-simneg.OvGU	0.5089
UA-roberta-final	**0.6250**	JNLP.TfidfBERT	0.5536	taxi-le-bigru	0.5089
KIS_2	0.6161	KIS_3	0.5446	TRC3A	0.5000
llntu	0.6161	sim_neg.OvGU	0.5446	UEC2	0.4911
cyber	0.6161	UEC1	0.5446	baseline-attention.OvGU	0.4821
UA_structure	**0.6071**	taxi_BERTXGB	0.5357	AUT99-BERT-MatchPyramid	0.4643
GK_NLP	0.5625	UECplus	0.5357	AUT99-LSTM-CNN-Attention	0.4464

5 Conclusions

In this paper, we described our approaches for two complex textual entailment tasks on the domain of legal documents: case law and statute law entailment. For case law entailment, our method relied on the usage of transformer models. The most relevant component was the one fine tuned on the training dataset provided in the competition. The other components in isolation presented lower scores, but the final f1 score from the combined components was lower than expected, probably because the components where not complementary. For statute law entailment, our pre-processing step considering the NLI types in the statute law improved the performance and our submission results were ranked No. 2.

We plan to extend this work by performing the following actions:

– Train BERT using a larger case law dataset and check whether it is capable of grasping law-related knowledge. There are publicly available case law corpus (e.g., Canadian Supreme Court Reports) which can be used as input;
– Fine tuning the framework with a larger dataset of case law entailment would be probably more effective, but there are not other case law entailment datasets known to the authors. We will generate such a dataset by applying simple heuristics or semiautomatic approaches on case law documents;
– Given the golden labels of the official COLIEE test dataset, perform an error analysis to identify the characteristics of the errors and if the different components are complementary or redundant;
– Use back translation fully integrated into the final model.

Acknowledgements

This research was supported by Alberta Machine Intelligence Institute (AMII), and would not be possible without the significant support of Colin Lachance from vLex and Compass Law, and the guidance of Jimoh Ovbiagele of Ross Intelligence and Young-Yik Rhim of Intellicon.

References

1. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. CoRR **abs/0912.3747** (2009), http://arxiv.org/abs/0912.3747
2. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
3. Ben Abacha, A., Demner-Fushman, D.: A question-entailment approach to question answering. CoRR **abs/1901.08079** (2019), http://arxiv.org/abs/1901.08079
4. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. ACL (2015)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (Oct 2001)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (Jun 2002)
7. Chen, Y., Zhou, Y., Lu, Z., Sun, H., Yang, W.: Legal information retrieval by association rules. In: Twelfth International Workshop on Juris-informatics (2018)
8. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: ML Challenges Workshop. pp. 177–190. Springer (2005)
9. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. CoRR (2015)
10. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
11. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proc. of the 3rd International Workshop on Paraphrasing (2005)
12. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. CoRR **abs/1705.03122** (2017)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735

14. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018), http://arxiv.org/abs/1801.06146
15. Kano, Y., Kim, M.Y., Yoshioka, M., Lu, Y., Rabelo, J., Kiyota, N., Goebel, R., Satoh, K.: Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In: 12rh International Workshop on Juris-informatics (2018)
16. Kim, M.Y., Goebel, R.: Two-step cascaded textual entailment for legal bar exam question answering. In: Proc. of the 16th Edition of the International Conference on Articial Intelligence and Law. pp. 283–290. ACM, New York, NY, USA (2017)
17. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.H.: RACE: large-scale reading comprehension dataset from examinations. CoRR **abs/1704.04683** (2017)
18. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. CoRR **abs/1405.4053** (2014)
19. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019)
20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
21. Lloret, E., Ferrández, Ó., Muñoz, R., Palomar, M.: A text summarization approach under the influence of textual entailment. In: NLPCS - 5th International Workshop on Natural Language Processing and Cognitive Science. pp. 22–31 (01 2008)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR (2013)
23. Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., Edunov, S.: Facebook fair's WMT19 news translation task submission. CoRR **abs/1907.06616** (2019)
24. Parikh, A.P., T.O.D.D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proc. of EMNLP. pp. 2249–2255 (2016)
25. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
26. Rabelo, J., Kim, M.Y., Babiker, H., Goebel, R., Farruque, N.: Legal information extraction and entailment for statute law and case law. In: Twelfth International Workshop on Juris-informatics (JURISIN) (2018)
27. Rabelo, J., Kim, M.Y., Goebel, R.: Combining similarity and transformer methods for case law entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 290–296. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019)
28. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2018)
29. Richardson K., Hu H., M.L.S., A., S.: Probing natural language inference models through semantic fragments. In: Proc. of AAAI. pp. 8713–8721 (2020)
30. Roemmele, M., Bejan, C., Gordon, A.: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In: AAAI Spring Symposium Series (2011)
31. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. CoRR **abs/1804.07461** (2018), http://arxiv.org/abs/1804.07461
32. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. CoRR **abs/1704.05426** (2017)

BERT-based Ensemble Model for The Statute Law Retrieval and Legal Information Entailment

Hsuan-Lei Shao[1[0000-0002-7101-5272]*], Yi-Chia Chen[2*] and Sieh-Chuen Huang[3[0000-0003-3571-5236]**]

[1] National Taiwan Normal University, Heping E. Rd., Taipei City 106, Taiwan
hlshao2@gmail.com
[2] National Taiwan Normal University, Heping E. Rd., Taipei City 106, Taiwan
40647045s@ntnu.edu.tw
[3] National Taiwan University, Roosevelt Rd. Rd., Taipei City 106, Taiwan
schhuang@ntu.edu.tw

Abstract. Competition on legal information extraction/entailment (COLIEE) is a renowned international information processing and retrieval competition. As an aid to future participants as well as question designers, this article describes how to connect legal questions taken from past Japanese bar exam to relevant statutes (articles of Japan Civil Code, Task 3) and how to construct a Yes/No question answering system for legal queries (Task 4) incorporating background materials on Japanese law. We restructured given data to a dataset which contains all possible combination of queries and articles to be continuous strings as our samples. In this way, the difficult pairing task has been turned to a simpler classification task and samples for training become sufficient in number. Next, we used three kinds of BERT-based models to solve binary questions in order to achieve a stable performance. As a result, the model achieved an F2-score of 0.6587 in Task 3 (rank 1st) and an accuracy of 0.6161 in Task 4.

Keywords: Textual Entailment, Information Retrieval, Legal AI, BERT-based Ensemble Model, Legal Analytics, COLIEE 2020

1 Introduction

Competition on Legal Information Extraction/Entailment (COLIEE), known as a famous international information retrieval competition, had its seventh run in 2020 [1,2,3]. Four tasks were included in the 2020 competition: Tasks 1 and 2 were the case law competition, and tasks 3 and 4 were statute law competition. Task 1 was a legal case retrieval task, requiring the participants to develop an approach capable of reading a new case Q, and extracting supporting cases S1, S2, ..., Sn from the provided case law corpus, to support the decision for Q. Task 2 was a legal case entailment task, which involves the identification of a paragraph from existing cases that entails the decision of a new case. Similar to the tasks featured in previous COLIEE competitions, Task 3 is to consider a yes/no legal question Q and retrieve relevant statutes from a database

of Japanese Civil Code statutes; Task 4 is to confirm entailment of a yes/no answer from the retrieved Civil Code statutes. [4]

Our team, *Legal Analytics Laboratory of National Taiwan University (LLNTU)*, participated in Task 3 (the Statute Law Retrieval Task) and Task 4 (the Legal Question Answering Task). Although this is the first time we participated in this competition, we are no strangers to establishments of a Civil Code and a similar bar exam in Taiwan. Perhaps this is why we do not feel unfamiliar with the data. We provided a solution to Task 3 using classification of pair sequences and ensemble of three kinds of BERT models. Fortunately, the BERT-based structure built a fine model for retrieving information: the F2-score was 0.6587 on Task 3 (best score among all participants). Then we applied the same model on Task 4, as to which the accuracy was 0.6161 (rank 6).

2 Competition Data and Task Description

The COLIEE organizer provided the "*Statute Law Competition Data Corpus*," in which legal questions were drawn from Japanese bar exams, and all Japanese civil law articles were also provided in both Japanese and English translation. The format of the COLIEE competition corpora has been derived from an NTCIR work [4], offering confirmed relationships between questions and the articles and cases relevant to answering the questions, as in the following example:

"H18-1-2"

<pair label="Y" id="H18-1-2">
<t1>
(Seller's Warranty in cases of Superficies or Other Rights)Article 566 (1)In cases where the subject matter of the sale is encumbered with for the purpose of a superficies, an emphyteusis, an easement, a right of retention or a pledge, if the buyer does not know the same and cannot achieve the purpose of the contract on account thereof, the buyer may cancel the contract. In such cases, if the contract cannot be cancelled, the buyer may only demand compensation for damages…
</t1>
<t2>
There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.
</t2>
</pair>

The <t2> part is the bar exam question Q, and the <t1> part is the article of Japan Civil Code relevant to Q. The above is an example where the query id "H18-1-2" is confirmed to be answerable from the Article 566 (relevant to Task 3). The pair label "Y" in this example means the answer for this query is "Yes", which is entailed from the relevant articles (relevant to Task 4). [4] For Tasks 3 and 4, the training data will be the same. We use them for the following two sub-tasks:

1) Task 3: Legal Information Retrieval Task. The input is a bar exam 'Yes/No' question and the output should be the relevant civil law articles.

2) Task 4: Recognizing Entailment between Law Articles and Queries. The input is a bar exam 'Yes/No' question. After retrieving relevant articles using our method, we determine 'Yes' or 'No' as the output. [4]

For Task 3, the evaluation measures are precision, recall and the F2-measure (instead of the ordinary F1-measure). The COLIEE organizer placed more emphasis on recall because the goal of information retrieval process is to select candidate articles to be used in the next entailment process [4], which is:

$$F2 - measure = \frac{(5\ x\ Precision\ x\ Recall)}{(4\ x\ Precision\ +\ Recall)}$$

$$Precision = \frac{average\ of\ (the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query)}{(the\ number\ of\ retrieved\ articles\ for\ each\ query)}$$

$$Recall = \frac{average\ of\ (the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query)}{(the\ number\ of\ relevant\ articles\ for\ each\ query)}$$

The goal of Task 4 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. When a 'Yes/No' legal bar exam question is given, our legal information retrieval system retrieves relevant Civil Code articles. The task investigates the performance of systems that answer 'Yes' or 'No' to previously unseen queries by comparing the meanings between queries and retrieved Civil Code articles. For Task 4, the evaluation measure will be accuracy. Training data consist of triples of a query, relevant article(s), a correct answer "Y" or "N". Test data includes only queries, but no 'Y/N' label, no relevant articles. [4]

3 Data Preprocessing

Our research processed only the Japanese-language Data Corpus provided by the COLIEE organizer, not the English-language data corpus. In the beginning, we have 696 bar exam questions as <t2> from H18~H30, and 767 Civil Code Articles as <t1>. Our design is to input the combinations of “bar exam question Q” and “article of the Japan Civil Code” as continuous strings. That is to say, for the <t2> component for each pair, <t2> is combined with every article of the Civil Code, producing 767 samples (the number of articles in the Part I~III of the Civil Code). Among these, only the sample containing the relevant article is labeled as “1” and other samples containing the non-relevant articles are labeled as “0”. For example, as Table 1 shows, as specified by <pair label="Y" id="H18-1-2"> in the training data, the query id "H18-1-2" is deemed

relevant to Article 566, so we labeled it as 1. On the other hand, regarding the non-relevant articles such as Article 1 and 2, the label shall be 0.

Table 1. Reconstruction of the training dataset

index	Bar exam questions "H18-1-2" (<t2>)	Japan Civil Code Articles (<t1>)	Label
1	Bar exam questions "H18-1-2" 目的物に地上権による制限があった場合の担保責任追及には期間制限があるが，抵当権の行使によって買主が権利を失った場合の担保責任追及には期間制限がない。 (There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.)	Japan Civil Code Article 566 （目的物の種類又は品質に関する担保責任の期間の制限）第五百六十六条　売主が種類又は品質に関して契約の内容に適合しない目的物を買主に引き渡した場合において、買主がその不適合を知った時から一年以内にその旨を売主に通知しないときは、買主は、その不適合を理由として、履行の追完の請求、代金の減額の請求、損害賠償の請求及び契約の解除をすることができない。ただし、売主が引渡しの時にその不適合を知り、又は重大な過失によって知らなかったときは、この限りでない。 ((Limitation on Period of Warranty with Respect to Kind or Quality of Subject Matter) Article 566　If the subject matter delivered by the seller to the buyer does not conform to the terms of the contract with respect to the kind or quality, and the buyer fails to notify the seller of the non-conformity within one year from the time when the buyer becomes aware of it, the buyer may not demand cure of the non-conformity of performance, demand a reduction of the price, claim compensation for loss or damage, or cancel the contract, on the grounds of the non-conformity; provided, however, that this does not apply if the seller knew or did not know due to gross negligence the non-conformity at the time of the delivery.	1 (given correct pair)

2	Bar exam questions "H18-1-2" 目的物に地上権による制限があった場合の担保責任追及には期間制限があるが，抵当権の行使によって買主が権利を失った場合の担保責任追及には期間制限がない。 (There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.)	Japan Civil Code Article 1 （基本原則）第一条　私権は、公共の福祉に適合しなければならない。２　権利の行使及び義務の履行は、信義に従い誠実に行わなければならない。３　権利の濫用は、これを許さない。 ((Fundamental Principles) Article 1 (1) Private rights must conform to the public welfare. (2) The exercise of rights and performance of duties must be done in good faith. (3) No abuse of rights is permitted.)	0 (fictitious wrong pair)
3	Bar exam questions "H18-1-2" 目的物に地上権による制限があった場合の担保責任追及には期間制限があるが，抵当権の行使によって買主が権利を失った場合の担保責任追及には期間制限がない。 (There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.)	Japan Civil Code Article 2 （解釈の基準）第二条　この法律は、個人の尊厳と両性の本質的平等を旨として、解釈しなければならない。 ((Standard for Construction) Article 2 This Code must be construed in accordance with honoring the dignity of individuals and the essential equality of both sexes.)	0 (fictitious wrong pair)
…	…	…	…
533,831	Bar exam questions "H28-35-3” 地上権者は，土地所有者の承諾を得ることなく地上権を第三者に譲渡することができるが，賃借人は，賃貸人の承諾又はそれに代わる裁判所の許可を得なければ，土地賃借権を譲渡することができない。 (A superficiary may assign the superficies without obtaining the approval of the owner of the land but a lessee may not assign the lessee's rights without obtaining	Japan Civil Code Article 724 （不法行為による損害賠償請求権の消滅時効）第七百二十四条　不法行為による損害賠償の請求権は、次に掲げる場合には、時効によって消滅する。一　被害者又はその法定代理人が損害及び加害者を知った時から三年間行使しないとき。二　不法行為の時から二十年間行使しないとき。 ((Extinctive Prescription of Claim for Compensation for Loss or Damage Caused by Tort) Article	0 (fictitious wrong pair)

	the approval of the lessor or the court's permission in lieu of it.)	724 In the following cases, the claim for compensation for loss or damage caused by tort is extinguished by prescription: (i) the right is not exercised within three years from the time when the victim or legal representative thereof comes to know the damage and the identity of the perpetrator; or (ii) the right is not exercised within 20 years from the time of the tortious act.)	
533,832	Bar exam questions "H28-35-3” 地上権者は，土地所有者の承諾を得ることなく地上権を第三者に譲渡することができるが，賃借人は，賃貸人の承諾又はそれに代わる裁判所の許可を得なければ，土地賃借権を譲渡することができない。 (A superficiary may assign the superficies without obtaining the approval of the owner of the land but a lessee may not assign the lessee's rights without obtaining the approval of the lessor or the court's permission in lieu of it.)	Japan Civil Code Article 724-2 （人の生命又は身体を害する不法行為による損害賠償請求権の消滅時効）第七百二十四条の二 人の生命又は身体を害する不法行為による損害賠償請求権の消滅時効についての前条第一号の規定の適用については、同号中「三年間」とあるのは、「五年間」とする。 ((Extinctive Prescription of Claim for Compensation for Loss or Damage Arising from Death to Person or Injury to Person Caused by Tort) Article 724-2 For the purpose of the application of the provisions of item (i) of the preceding Article with regard to the extinctive prescription of the claim for compensation for loss or damage for death or injury to person caused by tort, the term "three years" in the same item is deemed to be replaced with "five years".)	0 (fictitious wrong pair)

After combining each query with all articles, we construct a dataset composed of 533,832 samples (the 696 questions times 767 articles). Next, we allocate the samples in the earlier 11 years (H18~H28) to the training set which has 435,656 samples (the 568 questions times 767 articles), samples in the year before the latest year (H29) as a validation set which contains 44,486 samples (the 58 questions times 767 articles), and samples in the latest year (H30) as a test set, containing 53,690 samples (70 questions time 767 articles). Because the positive samples (samples with label “1”) are very few (about 1/700) and hence the dataset is imbalanced, we simply upsample positive ones in the training set to be 100 times. The aforementioned process can be seen as a kind of

data augmentation. After data preprocessing, we transform the pairing task to be a binary classification task.

4 Model Training and the Result

We use above-mentioned samples as inputs, then adopt three BERT-based models to train them. The first one is *BERT-base_mecab-ipadic-char-4k_whole-word-mask* [5], which abbreviated *BERTjpcwwm*. We preserve only the first 384 words (maximum length, *maxlen* 384) which is sufficient to give a good performance. (Fig. 1) The second BERT model is *BERTjpcwwm* with maxlen 512 and the third one is *ALBERTjp* with maxlen 512 (v2) [6]. We use Adam [7] as optimizer with learning rate: 4e-5, batch size: 48, loss function: binary cross entropy, epoch: 1. The valid loss of our model is approximately 0.64.

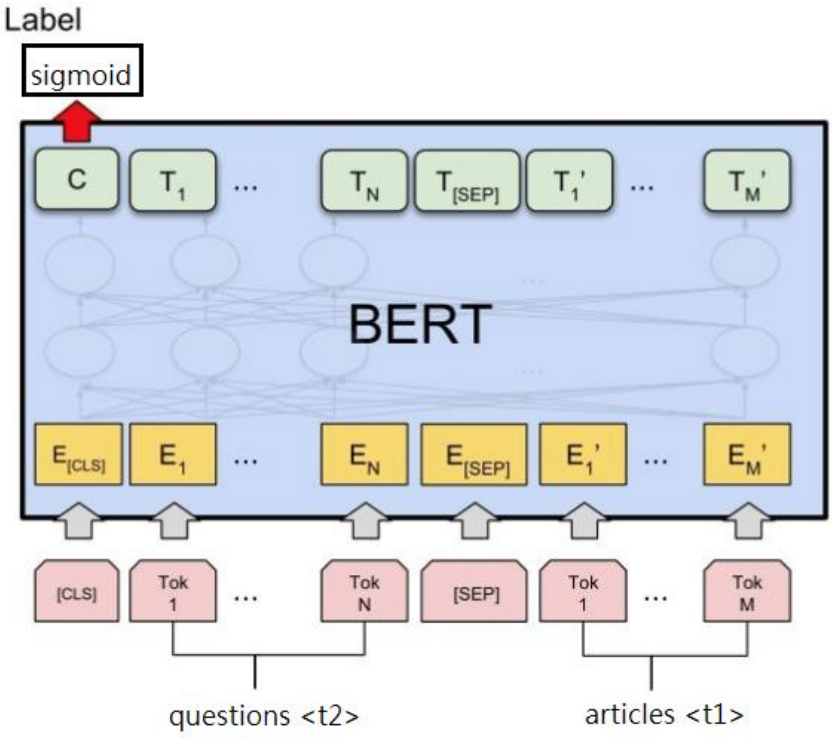


Fig. 1. Continious squence BERT model [8]

A sigmoid layer is put in the model finally. This model can indicate how much <t1> and <t2> are relevant. We train each model (*BERTjpcwwm* 384, *BERTjpcwwm* 512, *ALBERTjp* 512) independently as usual. Next, each validation sample or test sample is fed into these three models, yielding values between 0 and 1. Finally, in the ensemble

layer, these three values are averaged to provide the final prediction value (as shown in the following Fig 2.).

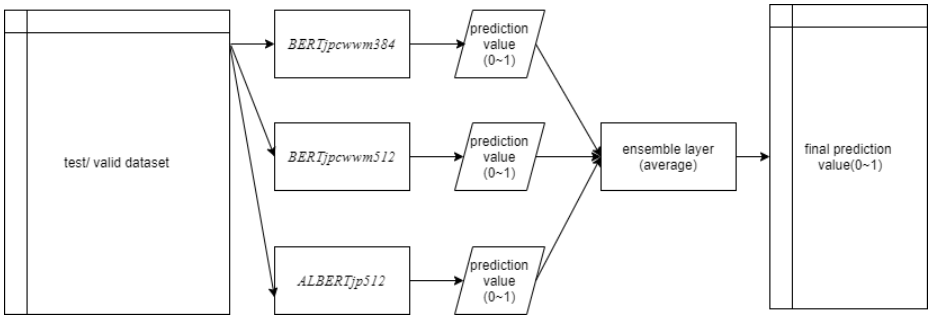


Fig 2. The structure of BERT-based ensemble model

The threshold to decide "relevant or irrelevant" pair is 0.8. We test different numbers for the threshold, ranging from 0.2 to 0.99, and record the F2-score performance of the validation set and the test set of each threshold to confirm model sensitivity (as shown in the following Fig. 3). The validation set curve (blue) reaches a peak at the threshold of 0.7. Meanwhile, the test set curve (orange) enters into a plateau period at the threshold ranging from 0.75 to 0.82. In order to acquire a stable result, an average curve (green) of the validation set and the test set is drawn, displaying that 0.8 is a relatively reliable and rational decision of the threshold.

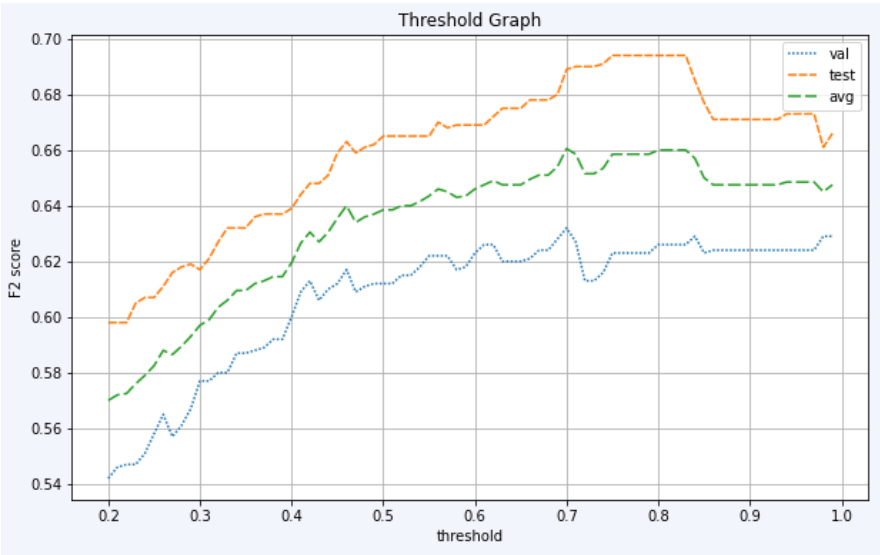


Fig. 3. The F2-score performance with different thresholds

With the threshold of 0.8, it may be possible that more than one articles are recognized as relevant at the same time. On the other hand, it also happens that no article has a F2-score value greater than 0.8. In this case, we obtain the maximum index to be the most possible relevant article using the arguments of the maxima (argmax) function. The COLIEE organizer also encouraged participants to submit a rank list with 100 candidates for each query as the "long answer", which gave us more opportunities to hit the target correctly. The "R_5" value means the recall by using top 5 ranked documents as returned documents, [1] "R_10" for our top 10, and "R_30" means our top 30, which are listed in the following Table 2.

Table 2. The result of BERT-based ensemble model on Task 3

Team	F2-score	Precision	Recall	MAP	R_5	R_10	R_30
LLNTU	0.6587	0.6875	0.6622	0.7604	0.8071	0.8571	0.9214

In terms of performance, as the Table 2 indicates, the F2-score of our model is 0.6587, the precision is 0.6875, and the recall is 0.6622. It is a stable model. In addition, with a wider rank list, we can provide 80% of the correct answers in the top five trial, and over 90% in top 30 hits.

5 TASK 4: Yes/No Question Answering

We use the same approach to solve Task 4. The goal of Task 4 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. When a 'Yes/No' legal bar exam question is given, our legal information retrieval system retrieves relevant Civil Code articles. The task is to answer 'Yes' or 'No' to queries based on these articles. [4] Training data consist of a triple set of a query, relevant article(s), a correct answer "Y" or "N". Test data include queries and relevant articles, but no 'Y/N' label.

Compared to sufficient samples in Task 3, in Task 4 we have only 568 samples to work with. Our model's Task 4 accuracy is 0.6161, which has 69 correct answers for all 112 questions. It is ranked around the 6th.

6 Discussion and Conclusion

The novelty we had is to combine <t2>s with both articles listed in <t1>s and articles not listed in <t1>s to be continuous strings as our samples. In this way, the difficult pairing task has been transformed into a simpler classification task, while producing sufficient number of samples for training. Next, we used three kinds of BERT-based models to solve binary questions in order to achieve a stable performance. However, there is further work to be done in the future.

6.1 Failure Analysis: Problems of Extraction-Based Question Answering

Our model shares the same limitations as extraction-based question answering, meaning that answer strings (often noun phrases or named entities) would be found in the query texts. [13] The model's failing answers can be categorized into two types.

Firstly, when there does not exist any common word for the correct <t1> and the respective query <t2>, our model has difficulties to recognize their relevancy and hence points to the wrong <t1>. For example, in <pair id='R01-1-O'>, the query is: 成年被後見人の行為であることを理由とする取消権の消滅時効の起算点は、成年被後見人が行為能力者となった時である (*The period of the extinctive prescription of the right to rescind about the act performed by an adult ward commences from the time of becoming a person with capacity to act).* The golden answer is that the query shall be related to Article 124 [9] and Article 126 [10], which are the rules of prescription. However, our model incorrectly identified Article 9, *the juristic acts of a ward is voidable*, as the most relevant article, with a low prediction value of 0.45. To answer this query correctly, as a human, it is necessary to understand the word "起算点(start point) " in the query has the same meaning as "時から(*from the time*)" in Article 126. The connection between these two words might have not been established in the BERT model. Furthermore, the query mentions "行為能力者となった(*becoming a person with capacity to act*)," which is the situation that a ward recovers the capacity so that his/her act is not voidable anymore but completely valid, which equals to "取消しの原因となっていた状況が消滅し(*the circumstances that made the act voidable cease to exist*)" stipulated in Article 124. In other words, it is necessary to pair the idea of "*becoming a person with capacity to act*" in the query with the idea of *"the circumstances that made the act voidable cease to exist"* in Article 124. However, our model does not succeed in doing this concept retrieval perhaps due to the lack of common words in the query and Article 124. This kind of mistake happens because the model cannot find sufficient hints to make a good decision. It needs additional information to derive the associations required to identify words belonging to similar concepts, such as legal domain knowledge.

Another incorrect answer pattern occurs when there exist several common words for the <t2> and <t1> pair so that it easily catches the model's attention. For example, in <pair id=' R01-2-E'>, the query is: Ａがその財産の管理人を置かないで行方不明となったことから、家庭裁判所は、Ｂを不在者Ａの財産の管理人として選任した。Ａが被相続人Ｆの共同相続人の一人である場合、ＢがＡを代理してＦの他の共同相続人との間でＦの遺産について協議による遺産分割をするためには、家庭裁判所の許可を得る必要はない(*The family court appointed B as the administrator of the property of absentee A, as A went missing without appointing the one. In cases where A is one of the joint heirs of decedent F, B doesn't need to obtain the permission of a family court when dividing inherited F's property by agreement between the other joint heirs as an agent of A*). Human would understand that this query is about the authority of an administrator. The golden answer is Article 28, which is exactly the provision governing the authority of an administrator. However, the word

“権限(authority)” does not exist in the query so that our model ignores Article 28 and the second answer, Article 103, which is cited by Article 28, providing also rules for the authority of an agent. In spite of the lack of the word “権限(authority)” in the query, Article 28 still has the same phrase “家庭裁判所の許可を得（obtain the permission of a family court)” as the query. But our model fails to recognize this information in the query to connect to Article 28. Instead, it gives Article 27 (stipulating that the administrator has an obligation to compile a property inventory) [11] as the wrong answer. Probably because the word “property（遺産・財産）”appears once in the query and three times in Article 27, our model incorrectly made the decision. Similar errors occur repeatedly in <pair id=' R01-2-O'>, <pair id=' R01-2-'>, <pair id=' R01-2-U'>, and <pair id=' R01-2-A'>, where the queries are all related to the powers or authority of the administrator. In these cases, the golden answers are Article 28 and Article 103, but our model incorrectly linked them to Article 27. It seems our model was not thorough enough when making the decision, then fell into the trap like a careless student.

The above-mentioned two patterns of errors in our model indicate that approaches using BERT may solve a certain kind of extraction-based question answering problem [14], but perform weaker when abstract legal concepts and reasoning are necessary. We are considering to integrate some legal domain knowledge with our approach as a mean to improve the performance of the model next year.

6.2 BERT and Y/N Questions

Our BERT-based ensemble model received the best score in Task 3, but it does not perform as well in Task 4. A direct and short answer is that BERT is not good enough at the yes/no questions [12], not to mention legal-related complex one. We consider that adding some rule-based models can perhaps improve the performance.

In addition, the first model that we tried in the beginning is exactly *BERT-base_mecab-ipadic-char-4k_whole-word-mask* [5]. Fortunately, it gave a good performance. Afterwards, we also attempted other models such as the XLM-RoBERTa. But compared to the initial *BERTjpcwwm384*, other models did not perform better. Finally we concluded that the best combination is: *BERTjpcwwm*(maxlen384), *BERTjpcwwm*(maxlen512), and *ALBERTjp*(maxlen512).

Acknowledgment

Hsuan-Lei Shao, “From Knowledge Genealogy to Knowledge Map-China Studies in Big Data and Machine Learning” (107-2410-H-003 -058 -MY3, *Ministry of Science and Technology, the MOST*), Taiwan.

Sieh-Chuen Huang, “Exploring Implications of Legal Analytics along with AI Technologies). (108-2628-H-002-005-MY2, *Ministry of Science and Technology, the MOST*), Taiwan.

References

1. Rabelo, J., Kim, M. Y., Goebel, R., Yoshioka, M., Kano, Y., & Satoh, K. A Summary of the COLIEE 2019 Competition.
2. Yoshioka, M., Kano, Y., Kiyota, N., & Satoh, K. (2018). Overview of japanese statute law retrieval and entailment task at coliee-2018. In Twelfth International Workshop on Juris-informatics (JURISIN 2018).
3. Kano, Y., Kim, M. Y., Yoshioka, M., Lu, Y., Rabelo, J., Kiyota, N., ... & Satoh, K. (2018, November). Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In JSAI International Symposium on Artificial Intelligence (pp. 177-192). Springer, Cham.
4. COLIEE organizer, COLIEE-2020 main webpage, https://sites.ualberta.ca/~rabelo/COLIEE2020/
5. Cl-tohoku, BERT-base_mecab-ipadic-char-4k_whole-word-mask, https://github.com/cl-tohoku/bert-japanese
6. Alinear-corp, albert-japanese, https://github.com/alinear-corp/albert-japanese.
7. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
8. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
9. Article 124: (1) The ratification of a voidable act does not become effective unless it is made after **the circumstances that made the act voidable cease to exist** and the person ratifying the act becomes aware of the right to rescind it. (2) In the following cases, the ratification referred to in the preceding paragraph is not required to be made after the circumstances that made the act voidable cease to exist: (i) if a legal representative or a curator or assistant of a person with qualified legal capacity ratifies the act; or (ii) if a person with qualified legal capacity (excluding an adult ward) makes the ratification with the consent of a legal representative, curator or assistant.
10. Article 126: The right to rescind an act is extinguished by the operation of the prescription if it is not exercised within five years **from the time** when it becomes possible to ratify the act. The same applies if 20 years have passed from the time of the act.
11. Article 27: (1) An administrator who is appointed by the family court pursuant to the provision of the preceding two Articles must prepare a list of the **property** he/she is to administer. In such case, the expenses incurred shall be disbursed from the **property** of the absentee. (2) In cases it is not clear whether an absentee is dead or alive, if so requested by any interested person or a public prosecutor, the family court may also order the administrator appointed by the absentee to prepare the list set forth in the preceding paragraph. (3) In addition to provisions of the preceding two paragraphs, the family court may issue an order to the administrator to effect any action which the court may find to be necessary for the preservation of the **property** of the absentee.
12. Clark, C., Lee, K., Chang, M. W., Kwiatkowski, T., Collins, M., & Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905.10044.
13. Barskar, R & Ahmed, G & Barskar, N. (2012). An Approach for Extracting Exact Answers to Question Answering (QA) System for English Sentences. Procedia Engineering. 30. 1187-1194. 10.1016/j.proeng.2012.01.979.
14. Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., ... & Li, J. (2019). Entity-relation extraction as multi-turn question answering. arXiv preprint arXiv:1905.05529.

THUIR@COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment*

Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu**, Min Zhang, and Shaoping Ma

Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
yiqunliu@tsinghua.edu.cn

Abstract. In this paper, we present our methodologies for tackling the challenges of legal case retrieval and entailment in the Competition on Legal Information Extraction/Entailment 2020 (COLIEE-2020). We participated in the two case law tasks, i.e., the legal case retrieval task and the legal case entailment task. Task 1 (the retrieval task) aims to automatically identify supporting cases from the case law corpus given a new case, and Task 2 (the entailment task) to identify specific paragraphs that entail the decision of a new case in a relevant case. In both tasks, we employed the neural models for semantic understanding and the traditional retrieval models for exact matching. As a result, our team ("TLIR") ranked 2nd among all of the teams in Task 1 and 3rd among teams in Task 2. Experimental results suggest that combing models of semantic understanding and exact matching benefits the legal case retrieval task while the legal case entailment task relies more on semantic understanding.

Keywords: Legal Case Retrieval · Legal Case Entailment · Neural Networks · Learning to Rank.

1 Introduction

Precedents are primary legal materials in the common law systems. For instance, retrieving and reviewing supporting precedents is fundamental for a lawyer who is preparing the legal reasoning in the countries that follows the common law system (e.g., USA, Canada). However, it takes a great number of efforts of legal practitioners to search for relevant cases and extract the entailment parts manually with the rapid growth of digitalized legal documents. Therefore, an efficient

* This work is supported by the National Key Research and Development Program of China (2018YFC0831700), Natural Science Foundation of China (Grant No. 61732008, 61532011), Beijing Academy of Artificial Intelligence (BAAI)and Tsinghua University Guoqiang Research Institute.

** Corresponding Author

legal assistant system that would alleviate the heavy document work has drawn increasing attention in the field of AI & Law.

Being held since 2014, the Competition on Legal Information Extraction / Entailment (COLIEE) is a series of evaluation competitions aiming to develop techniques for information retrieval and entailment using legal texts [3]. In COLIEE-2020[1], there are four tasks in total, among which, Task 1 & 2 are about the case law while Task 3 & Task 4 are about the statute law. Our team participated in the case law tasks (Task 1 & 2) this year. In previous studies, a variety of NLP and IR strategies have been developed and applied in these tasks. For example, Vu et al. [11] proposed a phrase scoring model and developed a summarization-based model for the legal case retrieval task. The application of the pre-trained language model BERT [2] has also been investigated. To deal with the long case documents, Rossi et al. [7] attempted to generate the summary of the case via unsupervised auto summarization algorithms (e.g., TextRank [6]) first and feed them into the BERT model. On the other hand, Shao et al. [8] proposed to model paragraph-level interactions to avoid information loss in the summarization process when dealing with long documents.

In this paper, we introduce our approaches to completing Task 1 and Task 2. Our models consider both exact matching and semantic understanding as well as the combination of them. To be specific, we utilize the word-entity duet framework [12] (exact matching) and the BERT-PLI model [8] (semantic understanding) in the first two runs in Task 1. Moreover, we extract features from these two methods and combine them through the learning to rank strategy (combination). In Task 2, we fine-tune BERT with symmetric and asymmetric truncation in the first two runs and combine the representation features with others, including the exact matching features and the meta-features. As a result, our team ranked 2nd among all of the teams in Task 1 and 3rd among teams in Task 2.

2 Task Description

2.1 Task 1: Legal Case Retrieval

Task 1 is a legal case retrieval task, which involves reading a new case (query case Q) and extracting supporting cases ($D = \{d_1, d_2, ..., d_n\}$) from the provided case law corpus. The "supporting case" (or "noticed case") is a legal term that denotes the precedent can support the decision of a query case, and thus in our paper, we consider it as the relevant case in the scenario of the legal case retrieval.

2.2 Task 2: Legal Case Entailment

Task 2 is a legal case entailment task, which involves identifying the paragraphs that entail the decision of a new case. Formally, given a decision fragment of a

[1] https://sites.ualberta.ca/~rabelo/COLIEE2020/

new case (denoted as q) and a relevant case composed of paragraphs (denoted as $R = \{p_1, p_2, ..., p_n\}$), the task is to find all of the p_i satisfying $entail(q, p_i)$. Different from the retrieval task, a relevant paragraph does not necessarily entail the query fragment.

2.3 Dataset

Data of Task 1 and Task 2 are both drawn from an existing collection of predominantly Federal Court of Canada case law. Table 1 gives a statistical summary of the dataset. In Task 1, the training and testing set consist of 520 and 130 query cases respectively, and each query case is provided with 200 candidate cases. In Task 2, the training and testing set have 325 and 100 base cases respectively, along with the corresponding decision fragment extracted from the base case. Besides, regarding each base case, a relevant precedent is given in the form of a paragraph list. The golden labels of the training set are provided in both tasks. Participants are required to submit the retrieved results on the testing set for evaluation.

Table 1. Summary of datasets in COLIEE 2020 Task 1 and Task 2.

Task 1	**Train**	**Test**
# query case	520	130
# candidate cases / query	200	200
# noticed cases / query	5.15	–
Task 2	**Train**	**Test**
# query case	325	100
# candidate paragraphs / query	35.52	36.72
# entailing paragraphs / query	1.15	–

2.4 Evaluation Measure

Both tasks are evaluated with micro-average of precision, recall, F1 score, where

$$Precision = \frac{Q_{TP}}{Q_{TP+FP}} \tag{1}$$

$$Recall = \frac{Q_{TP}}{Q_{TP+FN}} \tag{2}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

where Q_{TP} denotes the number of **correctly** retrieved cases (paragraphs) for all queries, Q_{TP+FP} is the number of retrieved cases (paragraphs) for all queries, and Q_{TP+FN} is the number of labels for all queries.

3 Methods

3.1 Task 1: Legal Case Retrieval

Run1 - Word-entity Duet. Traditional retrieval models are mostly calculated by term-level exact matching based on bag-of-words representations. Beyond bag-of-words, bag-of-entities (or bag-of-concepts) have been investigated in various scenarios, such as the ad-hoc text retrieval [12, 1] and medical retrieval [4]. Shao et al. [8] has pointed out that traditional bag-of-words IR models have competitive performances in legal case retrieval. Therefore, we would like to further study the exact-matching-based models and expand the bag-of-words representation to a mixed one. In particular, we employ the word-entity duet framework [12] in the first run. Each case document is modeled with word-based and entity-based representations. Then ranking features are generated to incorporate information from the word space, the entity space, and the cross-space connections. Due to the lack of public knowledge based in the Canadian legal domain, we simply utilize a general NER tool[2] to extract entities in the case texts. Through our pilot study, we find that the model performs better if we dismiss the term frequency in the entity-based features. Therefore, we follow this setting in this run.

Within the word-entity duet framework, a query case can interact with a candidate case document in different ways, including from query words to document words (Qw-Dw), from query entities to document words (Qe-Dw), and from query words to document entities (Qw-De). The interaction features are calculated by matching-based retrieval models, as shown in Table 2. In total, 11-dimensional features are utilized for ranking.

Table 2. Ranking features used in the word-entity duet model.

Interaction	Feature Description
Qw-Dw	BM25
Qw-Dw	TF-IDF
Qw-Dw	LM
Qw-Dw	Lm with JM smoothing
Qw-Dw	Lm with Dirichlet smoothing
Qw-Dw	Lm with two-way smoothing
Qe-Dw	BM25
Qe-Dw	TF-IDF
Qe-Dw	Lm with Dirichlet Smoothing
Qw-De	TF-IDF
Qw-De	Lm-Dirichlet

[2] https://www.nltk.org/

Based on the selected features, the relevance score between the query case and the candidate case is calculated by,

$$f(q,d) = g_a(W_{duet}^T \cdot v_{duet} + b_{duet}) \tag{4}$$

, where W_{duet} and b_{duet} denote the weight vector and bias respectively, and $g_a(\cdot)$ is the *sigmoid* function.

Following Xiong et al. [12], the model is trained by optimizing the pairwise hinge loss:

$$L_{hinge}(q,D) = \sum_{d \in D^+} \sum_{d' \in D^-} max(0, 1 - f(q,d) + f(q,d')) \tag{5}$$

, where D^+ and D^- are the set of relevant and irrelevant candidate cases. The loss function can be optimized using back-propagation in the neural network.

We consider it as a ranking problem in this run and return the top-5 ranked documents as the relevant cases. Before generating the rank features, we preprocess the documents by removing the punctuation and stop words via the NLTK tool. Parameters of the retrieval models (e.g. LM, BM25, etc.) in this run are kept as default. The learning rate of pairwise training is set as $1e-4$ without weight decay. The word-entity duet model is trained on the split training data for no more than 20 epochs and selected according to the F1 score on the split validation set. The division of the dataset would be detailed discussed in Section 4.

Run2 - BERT-PLI. Shao et al. [8] attempt to tackle the challenges caused by long and complex documents by modeling paragraph-level interactions and thus propose the **BERT-PLI** for legal case retrieval. Compared to traditional retrieval models, it has a better semantic understanding ability. Meanwhile, it can utilize the entire legal case document that is too long for the neural models developed for ad-hoc text retrieval to deal with. In COLIEE-2020, there still exist these challenges in the legal case retrieval task, including the extremely long documents, the complex definition of legal relevance, and the limited scale of training data. Therefore, we consider this task as a binary classification one and employ BERT-PLI in our second run.

As illustrated in Figure 1, for each input pair of query case and candidate case, (q, d_k), we first represent the case document by paragraphs, denoted by $q = (p_{q1}, p_{q2}, ..., p_{qN})$ and $d_k = (p_{k1}, p_{k2}, ..., p_{kM})$ respectively. In total, we obtain $N \times M$ pairs of paragraphs. Then we apply BERT to infer the semantic relationship of each paragraph pair, e.g., (p_{qi}, p_{kj}), and utilize the final hidden vector of [CLS] token as the representation of the corresponding paragraph pair. Once the interaction map through BERT inference is constructed, the *maxpooling* strategy is used to capture the strongest matching signals for each paragraph of the query case. The we utilize a forward RNN structure combined with the attention mechanism to further encode and aggregate the paragraph-based representation sequence, $\boldsymbol{p}'_{qk} = [\boldsymbol{p}'_{qk1}, \boldsymbol{p}'_{qk2}, \ldots, \boldsymbol{p}'_{qkN}]$, into a document-based representation, $\boldsymbol{d}_{qk}$. As for prediction, we pass the representation $\boldsymbol{d}_{qk}$ through a fully

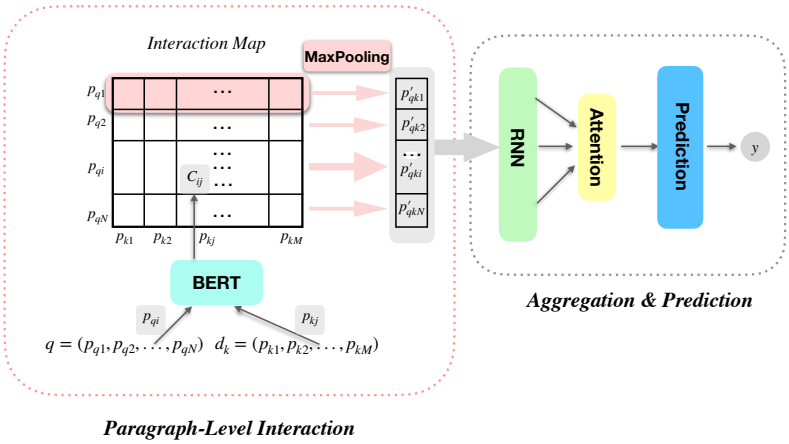


Fig. 1. An illustration of BERT-PLI [8].

connected layer followed by a *softmax* function and optimize the cross-entropy loss during the training process.

Following Shao et al. [8], we adopt a cascade framework considering the computational cost. Specifically, we rank all of the candidate documents ahead according to the scores given by bi-gram LMIR (**L**anguage **M**odel for **I**nformation **R**etrieval) [10] with a linear smoothing referring to prior work [9], and select the top-30 case as the candidates that would be further classified by BERT-PLI.

In our run, the uncased based BERT [2] version[3] is employed. Moreover, we do not update the parameters of BERT during the training process of BERT-PLI but utilize the parameters [8] that were fine-tuned with a legal paragraph dataset. The model settings as well as the training parameters all following the prior work [8], in which $N = 54$, $M = 40$, $lr = 1e-4$ with a weight decay of $1e-6$. In practice, we consider LSTM and GRU respectively as the RNN component and the results in the validation set suggest that GRU achieves better performance. Therefore, in the submitted run, a one-layer GRU with 256 hidden units is used. We train BERT-PLI on the split training data for no more than 60 epochs and select the best one according to the F1 score on the split validation set.

Run3 - Combination. Our first run is focused on the exact matching between two case documents based on traditional bag-of-words and bag-of-entities models, while our second run pays more attention to the semantic understanding through neural models. In this run, we attempt to combine the models of exact matching and semantic understanding. To be specific, we extract features of both kinds of models and employ learning to rank algorithms [5] to rank the candidate cases.

[3] https://github.com/google-research/bert

As for the features of exact matching, we use the 11-dimensional features listed in Table 2. Besides, we apply the SDR (**S**eed-driven **D**ocument **R**anking) [4], which is developed for the systematic review in medical retrieval, to the legal case retrieval task, and consider the similarity scores (calculated based on words and entities, respectively) as the additional two exact matching features. On the other hand, we extract the output vector (two-dimensional) of the *softmax* function in BERT-PLI as the features of semantic understanding. Besides, we use the first paragraph of the query and a candidate case as the input of BERT to fine-tune a sentence pair classification task [4] and extract the vector (two-dimensional) given by *softmax* function as additional features. In total, we obtain 13-dimensional (11+2) features based on exact matching and 4-dimensional features (2+2) based on semantic understanding. Note that BERT-PLI is only calculated on the top-30 candidate cases ranked by LMIR, so in this run, only the features of these top-30 candidates are considered.

We employ the RankSVM[5] to re-rank the top-30 candidates based on the 17-dimensional features. Specifically, we tune the parameter "-c" according to the F1 measure on the validation set and set $c = 20$ in this run. For each query, we consider the cases of non-negative predicted scores as the relevant ones. Meanwhile, if a query case has less than 3 relevant cases, we rank the remaining case based on the predicted scores and append the highly ranked ones to the result list until the query has 3 relevant cases.

3.2 Task 2: Legal Case Entailment

Run1 - BERT Fine-tuning. We fine-tune the BERT [2] with a downstream sentence pair classification task. As illustrated in Figure 2, the input pair of text is composed of the decision fragment in the query case and a candidate paragraph in the relevant case, denoted as (q, p_i), where p_i denotes the i-th paragraph in the corresponding relevant case. The final hidden state vector of [CLS] is then fed into a fully-connected layer for binary classification.

In this run, we make use of all of the candidate paragraphs in the given relevant case to construct the input text pair. If the total input tokens exceed the length limitation (512), the texts are truncated symmetrically. The model is training on the split training set by optimizing the *cross-entropy* loss. We update all of the parameters in an end-to-end way. The model is trained for no more than 5 epochs with $lr = 1e-5$ and selected according to the F1 measure on the validation set. The division of the dataset would be described in Section 4. As a result, we employ the one that trained for 4 epochs as the submitted run.

Run2 - BERT Fine-tuning with Asymmetric Truncation. The main difference from "Run1" is in that we truncate the text asymmetrically if the input tokens exceed the length limitation. In the training data, we observe that most of the decision fragment is no longer than 128 words while the paragraphs in

[4] The fine-tuning process is described in detail in Section 3.2, run1
[5] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

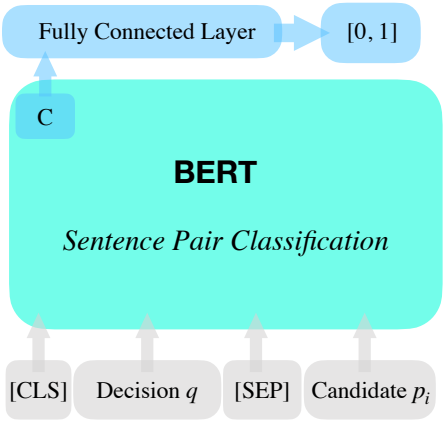


Fig. 2. Fine-tuning a sentence pair classification task.

the relevant case differ a lot. In the case where the candidate paragraph is rather long, symmetric truncation might lead to severe information loss in the decision fragment. Therefore, we are inspired to keep the tokens of decision fragment identical in each text pair corresponding to the query case.

In detail, we limit the tokens of decision fragment to 128, in which case, we truncate the decision fragment q ahead if it exceeds the limit. Then, we construct the input text pair combining with the candidate paragraphs, and we only truncate the tokens in the candidate paragraph p_i if the total length of the text pair exceeds 512 tokens. The model structure as well as the training process is the same as that in "Run1". With Asymmetric truncation, the training process converges faster and we select the model trained for 1 epoch as the submitted run.

Run3 - Combination. In this run, we also attempt to combine models of semantic understanding and exact matching along with some meta-information. Different from the retrieval task, the entailment task emphasizes the semantic understanding since the relevant/similar paragraphs might not entail the decision fragment. Therefore, we pay more attention to the features of semantic understanding models. To be specific, we extract the output vector of the fully-connected layer of "Run 1" and "Run 2" (4-dimensional in total). As for the features of exact matching, we only consider the BM25 scores (1-dimensional) implemented by gensim [6] with default parameters. Moreover, considering the legal document is somehow semi-supervised, we attempt to utilize the structure features including the position ID and the length of the paragraph (2-dimensional in total). To sum up, we obtain 7-dimensional features in this run and then RankSVM is employed based on them. We consider the paragraphs with non-negative predicted scores as the entailment paragraph, and if all of the predicted

[6] https://radimrehurek.com/gensim/

scores are negative, we select the top-ranked one as the result. Similarly, we tune the parameter "-c" according to the F1 measure on the validation set, and set $c = 1$ in this run.

4 Experiments and Results

For both tasks, we divide the training data into two separate parts randomly. 20% of queries as well as all of their candidates are treated as the validation set, and the remaining queries along with the candidates are utilized for training. We maintain the same division of training and validation set across all of the runs in the corresponding task.

4.1 Task1: Legal Case Retrieval

Table 3. Results on the test set of Task 1.

Team ID	Run ID	F1	Rank
TLIR	**Run 3**	**0.6682**	**3**
TLIR	Run 2	0.6379	6
TLIR	Run 1	0.5148	12
cyber	**Run 2**	**0.6774**	**1**
cyber	Run 3	0.6768	2
cyber	Run 1	0.6503	4
JNLP	BMW25	0.6397	5

Table 3 gives the results on the test data in Task 1, the legal case retrieval task[7]. Among our submitted runs ("TLIR"), Run 3, which combines the features of semantic understanding and exact matching, performs the best and outperforms the other single models in a nontrivial scale. This result suggests that the models of semantic understanding and the ones of exact matching focus on different aspects of legal documents, and thus can be complementary in the legal case retrieval task. Comparing the result of BERT-PLI (Run 2) with that of Duet (Run 1), the development of semantic neural models enhance the performance in the legal case retrieval a lot. We assume that this task calls for the ability of semantic understanding beyond the exact keywords matching due to the complexity of relevance definition in the legal domain.

The results are ranked according to the F1 score, and the top-2 runs are both from the team "cyber". We note that the difference between our method and the top-1 run in F1 is less than 0.01. Considering the limited data scale (130 test queries in total), we believe that our third method achieves rather close performance to the cyber's in this task.

[7] For simplicity, we only list parts of the runs here that we would like to further interpret

4.2 Task2: Legal Case Entailment

Table 4. Results on the test set of Task 2.

Team ID	Run ID	F1	Rank
TLIR	**Run 2**	**0.6154**	**4**
TLIR	Run 3	0.5495	13
TLIR	Run 1	0.5428	14
JNLP	**BMWT**	**0.6753**	**1**
JNLP	BMW	0.6222	2
taxi	XGBaft	0.6180	3
JNLP	WT+L	0.6094	5

As shown in Table 4, our second run, which utilizes the asymmetric truncation during BERT fine-tuning, achieves the best performance among our submitted three runs and the improvement is nontrivial. This result indicates the effectiveness of the asymmetric truncation. To be specific, the asymmetric truncation attempts to maintain the information in the decision fragment when constructing the input pairs. As an analogy, traditional retrieval models (e.g., BM25) tend to focus more on query items and thus we believe that the decision fragment (also denoted as "query") should be paid more attention to in this task. Unfortunately, our other two runs seem to be less effective in this task and the combination of different models does not help significantly. We interpret it from several perspectives. For one thing, considering the limited data scale, fine-tuning for 4 epochs might cause over-fitting. Since the division of training and validation set is constant in our experiments, the potential bias in the data division might mislead the model selection. For another, according to the task definition, a relevant paragraph does not necessarily entail the query fragment and thus the semantic understanding technique is required more than the exact matching one, which might explain why the combination does not work well in this task. Generally, our best run ranks 4th among all of the submitted runs in this task. The top-ranked one ("BMWT" from the "JNLP" team) does outperform all of the other runs considerably and is worth further discussion.

5 Conclusion and Future Work

In this paper, we introduce our methods in two case law tasks in COLIEE-2020, including a legal case retrieval task and a legal case entailment task. We investigate both semantic understanding and exacting matching methods and contribute three distinct runs in each task, respectively. In the legal case retrieval task, compared between the two types of models, the neural model that focuses on semantic understanding (BERT-PLI) outperforms the traditional retrieval model that is based on exact matching (Duet) a lot. Moreover, the combination

of two types of methods enhances each single one, which suggests that semantic understanding models and exact matching ones are complementary in this task. We rank 2nd in Task 1 and our best run ranks 3rd, which achieves rather close performance to the top-ranked ones. However, in the legal case entailment task, the semantic understanding ability is more important. We also find that reducing the information loss in the query fragment can help a lot in this task. In task 2, our best run ranks 4th among all runs and there still exists a performance gap from the top-1 run.

As an attempt to tackle the challenges in legal case retrieval and entailment, there still exist some limitations in our work that we would like to list as future directions. In this paper, different models are combined simply through RankSVM, a common learning to rank algorithm, while other more sophisticated mechanisms, such as pseudo-relevant feedback, are worth further investigation. Besides, the bias in the dataset division would mislead the optimization process of the supervised learning models. Cross-validation might be an alternative way in this scenario. Last but not least, we employ the BERT pre-trained on the common text by *Google* and merely fine-tune it on the limited legal data. With more available legal documents, pre-training a BERT in the legal domain might enhance most of the BERT-based models in this scenario.

References

1. Chen, J., Xiong, C., Callan, J.: An empirical study of learning to rank for entity search. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 737–740 (2016)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Juliano, R., Mi-Young, K., Randy, G., Masaharu, Y., Yoshi-nobu, K., Ken, S.: Coliee 2019 overview. In: COLIEE'19. pp. 1–9 (2019)
4. Lee, G.E., Sun, A.: Seed-driven document ranking for systematic reviews in evidence-based medicine. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 455–464 (2018)
5. Liu, T.Y.: Learning to rank for information retrieval. Springer Science & Business Media (2011)
6. Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: EMNLP'04. pp. 404–411 (2004)
7. Rossi, J., Kanoulas, E.: Legal information retrieval with generalized language models. In: COLIEE'19. pp. 16–19 (2019)
8. Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., Ma, S.: Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In: IJCAI-20. pp. 3501–3507 (2020)
9. Shao, Y., Ye, Z.: Thuir@ aila 2019: Information retrieval approaches for identifying relevant precedents and statutes. In: FIRE (Working Notes). pp. 46–51 (2019)
10. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proceedings of the eighth international conference on Information and knowledge management. pp. 316–321 (1999)

11. Tran, V., Nguyen, M.L., Satoh, K.: Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In: ICAIL'19. pp. 275–282 (2019)
12. Xiong, C., Callan, J., Liu, T.Y.: Word-entity duet representations for document ranking. In: Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval. pp. 763–772 (2017)

Recognizing Textual Entailment for Japanese Legal Text Using Lexical Simplification and Tuple-Based Matching and Similarity Features

Yusuke Suematsu, Suguru Matsuyoshi, and Akira Utsumi

The University of Electro-Communications, Tokyo, Japan
s2030054@edu.cc.uec.ac.jp, {matuyosi,utsumi}@uec.ac.jp

Abstract. The COLIEE2020 task 4 is a task of recognizing textual entailment (RTE) between a legal query (t_2) and its relevant articles (t_1). If t_1 entails t_2, an RTE system should answer "Yes." Otherwise, it should answer "No." We propose a method of recognizing textual entailment for Japanese legal text using lexical simplification and tuple-based matching and similarity features. *Kodomo Roppo* is a book explaining Japanese main laws in easy Japanese for children and contains 548 lexically-simplified articles. We have compiled a paraphrase dictionary by aligning original and their simplified articles manually. In a preprocessing step, we greedily substitute B for A in t_1 using noun paraphrase pairs $< A, B >$ in the dictionary. Our RTE system parses t_1 and t_2 and extracts from them tuples each of which consists of a subject, a predicate and an object. We introduce features based on the numbers of elements matched between tuples of t_1 and t_2, and those based on similarity between two different elements in the tuples. We use LightGBM, a gradient boosting decision tree, for binary classification. In the experiments with the training and formal run data, our system achieved accuracy of 0.593 and 0.545, respectively.

Keywords: Recognizing textual entailment · Lexical simplification of articles · Embedding vectors trained using judgment documents.

1 Introduction

The Recognizing Textual Entailment (RTE) task in natural language processing (NLP) is formulated as follows [1]:

> Textual entailment is defined as a directional relationship between pairs of text expressions, denoted by T and H. We say that T entails H if humans reading T would typically infer that H is most likely true.

This definition is based on common human understanding of language. The primary framework for evaluating textual entailment systems has been the series of the PASCAL RTE challenges [2]. Recently, multi-task benchmarks [17, 18] provide RTE tasks with large datasets for researchers of natural language understanding. The NTCIR-10 RITE2 and NTCIR-11 RITE-VAL workshops deal

t_1: 詐欺又は強迫による意思表示は、取り消すことができる。(A manifestation of intention based on fraud or duress is voidable.)
t_2: 消費者契約（消費者と事業者との間で締結される契約）において，事業者の詐欺により消費者がした意思表示は，取り消すことができる。(In Consumer Contract (a contract concluded between a Consumer and a Business Operator), manifestation of intention of Consumer which is induced by fraud of Business Operator may be rescinded.)

Fig. 1. An example (t_1, t_2) pair in the COLIEE2020 task 4.

with RTE tasks in simplified Chinese, traditional Chinese and Japanese in addition to English [10].

The COLIEE2020 task 4 is the RTE task between a legal query (t_2) and its relevant articles (t_1). Fig. 1 shows an example (t_1, t_2) pair in the task. If t_1 entails t_2, an RTE system should answer "Yes." Otherwise, it should answer "No."

In this paper, we propose a method of recognizing textual entailment for the COLIEE2020 task 4 (Japanese) using lexical simplification and tuple-based matching and similarity features. Our system translates t_1 into an easier one (t_1^p) with a paraphrase dictionary, and extracts features from the (t_1^p, t_2) pair as well as a (t_1, t_2) pair. It uses a Japanese parser for analyzing predicate argument structures and constructing tuple-based features as proposed by Kano et al. [6].

This paper consists as follows: Section 2 presents related work. In Section 3, we describe a compilation procedure of a paraphrase dictionary for lexical simplification of articles. We explain our classifiers, features, application of the paraphrase dictionary in Section 4. Section 5 reports hyper-parameter optimization of our classifiers using the training data. Section 6 provides our formal run results and discusses the proposed method. Section 7 summarizes the paper.

2 Related Work

2.1 Language Resources Available for Japanese Legal NLP

The Ministry of Internal Affairs and Communications of Japan provides textdata of Japanese laws in a XML format at the official web portal of Government of Japan[1,2]. The corpus has about 1.1 million sentences in Japanese. Nishino and Morikawa have published a Linked Open Data (LOD) format of the above textdata [12]. Their LOD includes RDF tuples representing reference relations between laws, definitions of legal terms described in laws, parallel structures in sentences, and semantic labels for text spans splitted with *touten* (comma).

[1] `https://elaws.e-gov.go.jp/download/lawdownload.html`
[2] Textdata translated into English of Japanese laws are also available at the web site of the Ministry of Justice of Japan: `http://www.japaneselawtranslation.go.jp/`.

The Ministry of Justice of Japan has published an electronic version of a Japanese-English (JE) dictionary of legal terms for JE translation tasks at its web site[3]. It contains 3,813 JE pairs of legal words, legal phrases and legal sentence patterns. It is also regarded as an authoritative list of Japanese legal terms available freely.

The Supreme Court of Japan provides a judgment search service at its web site[4]. We can download PDF files of judgments of the Supreme Court, High Courts, District Courts and Family courts, and obtain a large corpus of judgments by converting the PDF files to text files. It can be used for training language models and embedding word vectors.

2.2 Japanese Legal RTE and Simplification

There are several approaches proposed for Japanese legal RTE [5, 6, 11, 13], including extraction of requirement and effect parts, comparison between predicate argument structures, embedding vectors trained with judgment documents, multi-layer perceptron neural networks and a pre-trained language model BERT [3].

Shimazu et al. [8, 15] propose a legal paraphrasing system which rewrites legal paragraphs structually to increase their readability.

3 A Paraphrase Dictionary for Lexical Simplification of Articles in Japanese

Law articles (t_1) tend to include complicated terms while a legal query (t_2) mainly consists of simple legal terms and ordinary words. It is helpful to simplify t_1 lexically as a preprocessing step for aligning t_1 and t_2.

Kodomo Roppo [19] is a book explaining Japanese main laws in easy Japanese for children. It contains 548 lexically-simplified articles, whose original ones are extracted from Civil Code, Code of Civil Procedure, Penal Code, Code of Criminal Procedure, The Constitution of Japan, Juvenile Act, and Act for the Promotion of Measures to Prevent Bullying. It deals with only 79 Civil Code articles out of all 782. Thus, we decided to compile a paraphrase dictionary by aligning original and their simplified articles manually.

Except for The Constitution of Japan which we judged as far from Civil Code, we used the 493 articles for the four codes and two acts. Fig. 2 illustrates an example of extraction of paraphrase pairs from an original and its simplified articles. In the example, we extract paraphrase pairs < 過失 (negligence), 不注意 (carelessness)> and < 傷害する (cause NP to suffer injury), ケガをさせる (cause NP to get hurt)> by aligning Article 209 (1) of Penal Code and its simplified one manually. We have compiled a paraphrase dictionary consisting of 663 entries.

[3] http://www.japaneselawtranslation.go.jp/dict/download
[4] https://www.courts.go.jp/app/hanrei_jp/search1

original:
第二百九条　過失$_A$ により人を傷害した$_B$ 者は、三十万円以下の罰金又は科料に処する。(Article 209 (1) A person who causes another to suffer injury$_B$ through negligence$_A$ shall be punished by a fine of not more than 300,000 yen or a petty fine.)
Kodomo Roppo:
第二百九条　不注意$_A$ で誰かにケガをさせてしまった$_B$ 人は、30 万円以下の罰金か科料とします。(Article 209 (1) A person who causes another to get hurt$_B$ because of carelessness$_A$ shall be punished by a fine of not more than 300,000 yen or a petty fine.)

⇓

Original	Aligned	POS	Class
過失 (negligence)	不注意 (carelessness)	noun	legal-equivalent
傷害する (cause NP to suffer injury)	ケガをさせる (cause NP to get hurt)	verb	legal-equivalent

Fig. 2. Extraction of paraphrase pairs from Article 209 (1) of Penal Code and its simplified one.

Table 1. Numbers of paraphrase pairs belonging to each class.

Class	Num.	Example
legal-equivalent	391	< 酌量 (extenuation), 考慮 (consideration)>
legal-entailing	89	< 身体を拘束する (take NP into custody), 手錠をかける (handcuff)>
ordinary-equivalent	157	< 業務上 (in the course of business), 仕事中 (during work)>
ordinary-entailing	26	< 記録媒体 (recording medium), 写真 (photo)>
Total	663	

We classify paraphrase pairs into the following four classes: legal-equivalent, legal-entailing, ordinary-equivalent and ordinary-entailing, where "ordinary-" indicates that the source term of a paraphrase pair is not a legal term but a commonly-used word or phrase. Paraphrase pairs with the "legal-," "ordinary-," "-equivalent" and "-entailing" labels are linguistic resources for problems of "Legal terms," "Normal terms," "Paraphrase / Verb paraphrase" and "Entailment" categories listed in Table 8 of the COLIEE2019 summary paper [13], respectively. Table 1 shows the numbers of paraphrase pairs belonging to each class.

4 Proposed Method

Our system extracts several features from a given (t_1, t_2) pair and uses binary classifiers for outputting a Yes/No label. We explain classifiers, tuple extraction, features, and application of our paraphrase dictionary in Subsections 4.1, 4.2, 4.3 and 4.4, respectively.

4.1 Classifiers

We use the following three classifiers for binary classification:

- Support Vector Machine (SVM)[5]
- XGBoost[6]: It is an implementation of gradient boosting decision tree [4].
- LightGBM [7]: It is an implementation of gradient boosting decision tree with gradient-based one-side sampling and exclusive feature bundling.

We also adopt a voting ensemble of the above three, where a label with the most votes is predicted.

4.2 Tuple Extraction

In this study, we define a tuple as a combination of a subject, a predicate and an object. We adopt a verb, an adjective, an adjectival noun and "be" + a noun phrase as predicates. We assign nouns or compound nouns with no modifying phrase to subject and object slots.

We use JUMAN [9] and KNP [14] for parsing Japanese texts t_1 and t_2. They analyze the predicate argument structure for each predicate after ellipsis and anaphora resolution is performed. The upper side of Fig. 3 shows a predicate argument structure analysis of the second sentence of Article 547 of Civil Code, where the underlined expression "相手方 (the other party)" appears in the first sentence of the article and is selected as an argument of the verb "受ける (receive)" by KNP. The first numbers of the rows are the serial numbers (IDs) of phrases in the sentence, and the second numbers are the IDs of the parent phrases which they depends on in the dependency tree. We use expressions in PRED, SUBJ and OBJ for elements of a tuple. If SUBJ and/or OBJ have no expression, we assign a dummy word "UNKNOWN" to the slot(s), as shown in the lower side of the figure.

KNP can tag predicates in a sentence with "negated" labels. A label "否定表現 (negated)" is used for a predicate negated by a depending negative word, such as "ない (not)" and "ず (not)." A label "二重否定 (double-negated)" is used for a predicate negated by two depending negative words, such as "ないことはない (not the case that ··· does not ···)" and "なければならない (must; lit., not occur if ··· does not ···)." If the predicate in a tuple is tagged with "否定表現

[5] We use SVM tools in Scikit-learn library: `https://scikit-learn.org/`.
[6] `https://xgboost.readthedocs.io/`

Input:
この場合において、その期間内に解除の通知を受けないときは、解除権は、消滅する。(In this case, if no notice of cancellation is received within that period, the right to cancel is extinguished.)

⇓

ID	Dep.	Phrase	Predicate Argument Structure	Label
0	1	この (this)		
1	2	場合に (case)		
2	11	おいて、(in)		
3	4	その (that)		
4	7	期間内に (within period)		
5	6	解除の (of cancellation)		
6	7	通知を (notice)		
7	8	受けない (not receive)	PRED: 受ける (receive); SUBJ: 相手方 (the other party); OBJ: 通知 (notice); TIME: 期間内 (within that period)	否定表現 (negated)
8	11	ときは、(if)		
9	10	解除 (cancellation)		
10	11	権は、(right)		
11	-1	消滅する。 (be extinguished)	PRED: 消滅する (be extinguished); SUBJ: 解除権 (right to cancel); OBJ: –; TIME: とき (if); IN: 場合 (this case)	

⇓

(相手方 (the other party), 受ける (receive), 通知 (notice))$_{\text{neg,conditional}}$
(解除権 (right to cancel), 消滅する (be extinguished), UNKNOWN)$_{\text{pos,main}}$

Fig. 3. Tuple extraction from the second sentence of Article 547 of Civil Code.

(negated)," we label the tuple with "neg." Otherwise, we label it with "pos" for representing "no negation."

We also tag extracted tuples with clause labels, which consists of "main clause" and "conditional clause," by utilizing keywords such as "場合 (in case)" and "とき (if)" around the predicates of them. For example, the first tuple in Fig. 3 is tagged with "conditional clause" because the predicate "受ける (receive)" depends on the phrase "ときは (if)."

KNP can capture parallel structures in a sentence and distinguish a parallel relation between two phrases from other dependence relations. If an element of a tuple has a parallel structure in it, our system divides it into several different tuples which have no parallel structure in them. For example, it divides (債権及び債務 (claim and obligation), *pred*, *obj*) into (債権 (claim), *pred*, *obj*) and (債務 (obligation), *pred*, *obj*). It divides (UNKNOWN, 禁止し又は制限する (pro-

hibit or restrict), 譲渡 (assignment)) into (UNKNOWN, 禁止する (prohibit), 譲渡 (assignment)) and (UNKNOWN, 制限する (restrict), 譲渡 (assignment)). Our tuple extraction makes no distinction between AND and OR enumerations, because calculation of feature values mainly requires existence of at least one tuple combination meeting a condition, and a maximum value of similarity among all possible combinations of two tuples, as described in the next subsection.

4.3 Features

Our system constructs tuple-based matching and similarity features for a given (t_1, t_2) pair.

Tuple-based matching features for COLIEE RTE tasks have been originally proposed by Kano et al. [6]. The main three matching features are MATCH3, MATCH2 and MATCH1. Suppose that $a_{1,i}$ and $a_{2,j}$ be tuples in the set of tuples extracted from t_1 and t_2, respectively.

- MATCH3: If there is at least one combination of $a_{1,i}$ and $a_{2,j}$ which meets the condition that each element in the former matches the corresponding one in the latter, this feature takes 1. Otherwise, it takes 0.
- MATCH2: If there is at least one combination of $a_{1,i}$ and $a_{2,j}$ which meets the condition that arbitrary two of the elements in the former match the corresponding ones in the latter, this feature takes 1. Otherwise, it takes 0.
- MATCH1: If there is at least one combination of $a_{1,i}$ and $a_{2,j}$ which meets the condition that arbitrary one of the elements in the former matches the corresponding one in the latter, this feature takes 1. Otherwise, it takes 0.

Restriction of the above definitions to $a_{1,i}$ and $a_{2,j}$ with the "main clause" label leads to other features. We call them MATCH3_MAIN, MATCH2_MAIN and MATCH1_MAIN. In a similar manner for the "conditional clause" label, we define MATCH3_CONDITIONAL, MATCH2_CONDITIONAL and MATCH1_CONDITIONAL.

We have the following two options for the above nine features:

- *_NOT: If the selected $a_{1,i}$ and $a_{2,j}$ have different "negated" labels, e.g. $a_{1,i}$ is a normal proposition and $a_{2,j}$ is a negated proposition, this feature takes 1. Otherwise, it takes 0.
- *_ELLIPSIS: A dummy word "UNKNOWN" is regarded as a normal word. This means that matching ellipses in $a_{1,i}$ and $a_{2,j}$ is allowed.

Thus, the number of the tuple-based matching features our system uses is 36 $(= 9 \times 2 \times 2)$.

We propose tuple-based similarity features for a given (t_1, t_2) pair, each of which takes a decimal between -1 and 1. The aim of introducing them is to soften the MATCH2 and MATCH1 features by measuring similarity between two different elements.

We define a set of tuple pairs $D^{(*,p,o)}$ as follows:

$$a_{1,i} := (s_{1,i}, p_{1,i}, o_{1,i}),\ a_{2,j} := (s_{2,j}, p_{2,j}, o_{2,j}),$$
$$D^{(*,p,o)} := \{(a_{1,i}, a_{2,j}) | s_{1,i} \neq s_{2,j},\ p_{1,i} = p_{2,j},\ o_{1,i} = o_{2,j}\}.$$

$D^{(s,*,o)}$ and $D^{(s,p,*)}$ are defined in similar fashion. Then, the values of the MATCH2_SIMS, MATCH2_SIMP and MATCH2_SIMO features are calculated below:

$$\text{MATCH2\_SIMS} = \max_{D^{(*,p,o)}} \text{sim}(s_{1,i}, s_{2,j}),$$
$$\text{MATCH2\_SIMP} = \max_{D^{(s,*,o)}} \text{sim}(p_{1,i}, p_{2,j}),$$
$$\text{MATCH2\_SIMO} = \max_{D^{(s,p,*)}} \text{sim}(o_{1,i}, o_{2,j}).$$

For the sim() function, we use cosine similarity between the embedding vectors of two words. We use Wikipedia Entity Vectors [16] for embedding word vectors, which were trained with skip-gram algorithm using preprocessed Japanese Wikipedia text as the corpus.

Similarly, we also define a set of tuple pairs $D^{(*,*,o)}$ as follows:

$$D^{(*,*,o)} := \{(a_{1,i}, a_{2,j}) | s_{1,i} \neq s_{2,j},\ p_{1,i} \neq p_{2,j},\ o_{1,i} = o_{2,j}\}.$$

The value of the MATCH1_SIMSP feature is calculated below:

$$\text{MATCH1\_SIMSP} = \max_{D^{(*,*,o)}} \frac{1}{2} \left(\text{sim}(s_{1,i}, s_{2,j}) + \text{sim}(p_{1,i}, p_{2,j})\right).$$

The values of the MATCH1_SIMPO and MATCH1_SIMSO features are calculated in similar fashion.

We have the following two options for the above six features:

- *_JUDGE: This option adopts embedding word vectors which were trained using legal texts as proposed in Morimoto et al. [11], instead of Wikipedia Entity Vectors. We collected 49,401 judgment documents from the web site described in Subsection 2.1, and trained embedding vectors using them by skip-gram algorithm[7] with vector dimension size of 300.
- *_NOT: If the selected $a_{1,i}$ and $a_{2,j}$ have different "negated" labels, this feature takes 1 (not decimal). Otherwise, it takes 0.

The number of the tuple-based similarity features our system uses is 24 ($= 6 \times 2 \times 2$).

In addition to tuple-based features, we introduce text similarity features for a given (t_1, t_2) pair. The value of the WHOLE_SIM feature is calculated as cosine similarity between the two averaged vectors of the embedding vectors of the words composing t_1 and t_2. Restriction of this definition to the "main clause" spans of t_1 and t_2 leads to the MAIN_SIM feature. The CONDITIONAL_SIM feature is also defined in similar fashion.

Text similarity features have the following option:

- *_JUDGE: Embedding word vectors which were trained using judgment documents are adopted.

The number of the text similarity features our system uses is 6 ($= 3 \times 2$).

Eventually, our system constructs 66 features for machine learning algorithms.

[7] We use Word2Vec tools in gensim library: `https://radimrehurek.com/gensim/`.

Table 2. Classification results of our classifiers with the best hyper parameters for the training data.

Classifier	Source	Hyper parameters	Corr.	Acc.
SVM	T_1	kernel=poly, C=100	375	0.539
	T_1^p	kernel=poly, C=10	368	0.529
	$T_1 \oplus T_1^p$	kernel=poly, C=100	360	0.517
XGBoost	T_1	min_child_w=1, subsample=0.5, max_depth=7	375	0.539
	T_1^p	min_child_w=1, subsample=0.5, max_depth=3	362	0.520
	$T_1 \oplus T_1^p$	min_child_w=1, subsample=0.5, max_depth=3	354	0.509
LightGBM	T_1	leaves=20, min_data=10, max_depth=8	396	0.569
	T_1^p	leaves=10, min_data=20, max_depth=4	**413**	**0.593**
	$T_1 \oplus T_1^p$	leaves=50, min_data=10, max_depth=16	403	0.579
Voting	T_1		395	0.568
	T_1^p		391	0.562
	$T_1 \oplus T_1^p$		374	0.537

4.4 Application of the Paraphrase Dictionary

We can translate an article t_1 into an easier one (t_1^p) with our paraphrase dictionary described in Section 3. Our system can extract 66 features from the (t_1^p, t_2) pair as well as a (t_1, t_2) pair. Therefore, we have the following three options for feature sources:

- T_1: extracts features from (t_1, t_2) pairs only. 66 features.
- T_1^p: extracts features from (t_1^p, t_2) pairs only. 66 features.
- $T_1 \oplus T_1^p$: extracts features from both (t_1, t_2) and (t_1^p, t_2) pairs. 132 features.

Using paraphrase pairs $< A, B >$ in the dictionary as exemplified in Table 1, we greedily substitute B for A in t_1. In a small preliminary experiment, substitution using only paraphrase pairs whose parts-of-speech (POS) are nouns or noun phrases provides our RTE system with good performance. Thus, we use only noun paraphrase pairs for substitution of t_1. The number of the noun paraphrase pairs in the dictionary is 401.

5 Hyper-Parameter Optimization

We optimized hyper parameters of the classifiers described in Subsection 4.1 by using grid search. The training data for the COLIEE2020 task 4 (Japanese) consists of 13 files, containing 696 (t_1, t_2) pairs in total. We used it for grid search with 13 cross validation. The hyper parameters and their candidate values for optimization are listed as follows:

SVM kernel: linear, poly; regularization parameter C: 0.0001, 0.001, 0.01, 0.1, 1, 10, 100.
XGBoost min child's weight: 1, 3, 5; subsample: 0.5, 1; max depth: 1, 3, 5, 7.
LightGBM leaves: 10, 20, 50, 100; min data in leaf: 10, 20, 40, 60, 100; max depth: 1, 2, 4, 8, 16.

Table 3. COLIEE2020 Task 4 results (Excerpt).

Rank	Run name	Corr.	Acc.
1	JNLP.BERTLaw	81	0.7232
16	UEC1	61	0.5446
19	UECplus	60	0.5357
–	Baseline	59	0.5268
27	UEC2	55	0.4911

Voting Arbitrary combinations of the above candidates.

Table 2 shows the best hyper parameters for each classifier and each feature source. The best hyper parameters for Voting are omitted due to space limitation. "Corr." and "Acc." stand for the number of correctly classifed (t_1, t_2) pairs and classification accuracy, respectively. The table indicates that LightGBM with the T_1^p source achieves the best performance for the training data. LightGBM with the T_1^p source and the second best hyper parameters (leaves=5, min_data=20, max_depth=8) has accuracy of 0.588 and is also dominant over SVM, XGBoost and Voting. Therefore, we firstly selected LightGBMs with the T_1^p source and the best and the second best hyper parameters as our proposed systems **UEC1** and **UEC2**, respectively. Then, we selected LightGBMs with the $T_1 \oplus T_1^p$ source and the best hyper parameters as our other proposed system **UECplus** from the viewpoint of diversity of features.

6 Task 4 Results and Discussion

6.1 Leaderboard

The formal run data for the COLIEE2020 task 4 contains 112 (t_1, t_2) pairs, out of which 59 are labeled with "Yes" and 53 with "No." The number of the submitted system runs, including our **UEC1**, **UEC2** and **UECplus**, is 30. Table 3 shows an excerpt of the leaderboard for the task 4, which consists of the results of our system runs, the state-of-the-art run and a baseline run. The ranks of our system runs are 16th, 19th and 27th.

6.2 Discussion

LightGBM can calculate importance of each feature with respect to the constructed decision trees. Table 4 shows the top 5 important features of **UEC1**, **UEC2** and **UECplus** and their importance values for the whole training data, where figures are normalized importance values which are decimals between 0 and 1. All the systems give priority to tuple-based similarity features over tuple-based matching ones. They also suggest that MATCH1_SIMSP should be accompanied with embedding word vectors trained using judgment documents, not with those trained using Japanese Wikipedia text.

Table 4. Top 5 important features of our systems.

UEC1		**UEC2**		**UECplus**	
MATCH1_SIMSP_JUDGE	0.33	MATCH1_SIMPO	0.19	MATCH1_SIMSP_JUDGE$_{T_1^p}$	0.12
MATCH2_SIMP	0.22	MATCH1_SIMSP_JUDGE	0.14	MATCH2_SIMP$_{T_1^p}$	0.10
MATCH1_SIMPO	0.22	MATCH2_SIMP	0.10	MATCH1_SIMPO$_{T_1^p}$	0.10
MATCH1_SIMSO_JUDGE	0.11	MATCH2_SIMS	0.05	MATCH2_SIMP_JUDGE$_{T_1}$	0.08
MATCH1_SIMPO_CONDITIONAL_JUDGE	0.11	MATCH2_SIMS_JUDGE	0.05	MATCH1_SIMPO$_{T_1}$	0.08

Table 5. Classification results of our classifiers with the best hyper parameters for the formal run data.

Classifier	Source	Hyper parameters	Corr.	Acc.
SVM	T_1	kernel=poly, C=100	62	0.554
	T_1^p	kernel=linear, C=1	61	0.545
	$T_1 \oplus T_1^p$	kernel=poly, C=0.1	62	0.554
XGBoost	T_1	min_child_w=1, subsample=0.5, max_depth=3	55	0.491
	T_1^p	min_child_w=1, subsample=0.5, max_depth=3	51	0.455
	$T_1 \oplus T_1^p$	min_child_w=1, subsample=0.5, max_depth=7	52	0.464
LightGBM	T_1	leaves=10, min_data=20, max_depth=1	63	0.563
	T_1^p	leaves=10, min_data=100, max_depth=2	63	0.563
	$T_1 \oplus T_1^p$	leaves=10, min_data=10, max_depth=4	63	0.563
Voting	T_1		**64**	**0.571**
	T_1^p		61	0.545
	$T_1 \oplus T_1^p$		**64**	**0.571**

We performed extra experiments using the formal run data for evaluating our other classifiers. Table 5 shows the classification results of them, where the best hyper parameters for Voting are omitted due to space limitation. From the table, we found that SVM, LightGBM and Voting have comparable accuracy for the formal run data, although LightGBM is superior to SVM and Voting for the training data.

The feature source T_1^p is not performing well for the formal run data. The numbers of t_1 in the formal run and the training data whose at least one word can be substituted using our paraphrase dictionary are 108 ($\frac{108}{112} = 0.964$) and 659 ($\frac{659}{696} = 0.947$), respectively. Therefore, it can be said that the dictionary has comparable application ratios for the two data. Increasing paraphrase pairs and entailing pairs in it and proposing a method of making effective use of it are our future work.

Our system deeply depends on a Japanese parser and tuple-based features as Kano et al.'s system [6] does. In future work, we investigate heuristic preprocessing and postprocessing approaches for better predicate argument structure

analysis, ellipsis and anaphora resolution, numerical normalization, and unification of characters between t_1 and t_2.

7 Conclusion

In this paper, we proposed a method of recognizing textual entailment for the COLIEE2020 task 4 (Japanese) using lexical simplification and tuple-based matching and similarity features. Our system translates t_1 into an easier one (t_1^p) with our paraphrase dictionary, and extracts features from the (t_1^p, t_2) pair as well as a (t_1, t_2) pair. In the experiments with the training and formal run data, it achieved accuracy of 0.593 and 0.545, respectively. The feature source $T_1^p = \{(t_1^p, t_2)\}$ is not performing well for the formal run data.

Our future work includes collecting paraphrase pairs and entailing pairs for Japanese legal text, investigating preprocessing and postprocessing approaches for better tuple construction, and incorporating pre-trained language models such as BERT in our framework.

Our paraphrase dictionary is applicable to a legal text simplification task as well as a legal RTE task. Developing a system of automatically simplifying legal text including, but not limited to, articles of Japanese Civil Law is other future work.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP18K11427.

References

1. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché Buc, F. (eds.) First PASCAL Machine Learning Challenges Workshop, pp. 177–190. Springer-Verlag (2006)
2. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.: Recognizing Textual Entailment: Models and Applications. Morgan & Claypool Publishers (2013)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186 (2019)
4. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
5. Hoshino, R., Kano, Y.: Recognizing textural entailment for Japanese legal bar exams using BERT. In: Proceedings of the 26th meeting of the Association for Natural Language Processing. pp. 577–580 (2020), (in Japanese)
6. Kano, Y., Hoshino, R., Taniguchi, R.: Analyzable legal yes/no question answering system using linguistic structures. In: Proceedings of the 4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017). pp. 57–67 (2017)

7. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30 (NIPS 2017). pp. 3149–3157 (2019), https://lightgbm.readthedocs.io/
8. Kimura, Y., Nakamura, M., Shimazu, A.: Treatment of legal sentences including itemized and referential expressions -towards translation into logical forms-. LNAI **5447**, 242–253 (2009)
9. Kurohashi, S., Nakamura, T., Matsumoto, Y., Nagao, M.: Improvements of japanese morphological analyzer JUMAN. In: Proceedings of The International Workshop on Sharable Natural Language Resources. pp. 22–28 (1994), http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
10. Matsuyoshi, S., Miyao, Y., Shibata, T., Lin, C.J., Shih, C.W., Watanabe, Y., Mitamura, T.: Overview of the NTCIR-11 recognizing inference in text and validation (RITE-VAL) task. In: Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies. pp. 223–232 (2014)
11. Morimoto, A., Kubo, D., Sato, M., Shindo, H., Matsumoto, Y.: Legal question answering system using neural attention. In: Proceedings of the 4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017). pp. 79–89 (2017)
12. Nishino, F., Morikawa, H.: Japanese laws LOD (2019), https://github.com/lod4all/e-laws-lod
13. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: A summary of the COLIEE 2019 competition. In: Proceedings of the Thirteenth International Workshop on Juris-informatics (JURISIN 2019) associated with JSAI International Symposia on AI 2019 (IsAI-2019) (2019)
14. Sasano, R., Kurohashi, S.: A discriminative approach to japanese zero anaphora resolution with large-scale lexicalized case frames. In: Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP2011). pp. 758–766 (2011), http://nlp.ist.i.kyoto-u.ac.jp/?KNP
15. Shimazu, A.: Structural paraphrase of law paragraphs. In: Proceedings of the Eleventh International Workshop on Juris-informatics (JURISIN) (2017)
16. Suzuki, M., Matsuda, K., Sekine, S., Okazaki, N., Inui, K.: A joint neural model for fine-grained named entity classification of wikipedia articles. IEICE Transactions on Information and Systems, Special Section on Semantic Web and Linked Data **E101-D**(1), 73–81 (2018), https://github.com/singletongue/WikiEntVec
17. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In: arXiv (2019), http://arxiv.org/abs/1905.00537
18. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355 (2018)
19. Yamasaki, S.: *Kodomo Roppo* (lit. Compendium of Japanese Laws for Children). *Kobundo* (2019), (in Japanese)

Legal Information Retrieval and Entailment Detection: Hybrid Approaches of Traditional Machine Learning and Deep Learning

Sabine Wehnert[1], Venkatesh Murugadas[1], Sachin Nandakumar[1], Anirban Saha[1], Tousifur Rahman Khan[1], Martin Urban[2], and Ernesto William De Luca[1]

[1] Otto von Guericke University Magdeburg, Germany
[2] Oberlandesgericht Naumburg, Germany

Abstract. Working with legal statutes is challenging, given the abstract nature of top-down legislation and the competences needed to identify applicable statutes for a situation. In course of the COLIEE competition, research has been directed to retrieving relevant statutes and classifying their entailment relationship with a hypothesis. We study hybrid methods of deep learning and traditional machine learning approaches, motivated by the observation that both fields could benefit from each other. We identify feature contributions and thus offer further interpretation on the predictions, which is often a requirement in the legal domain.

Keywords: Information Retrieval · Textual Entailment · Legal AI.

1 Introduction

Legal Artificial Intelligence is one of the hardest domains for applied Natural Language Processing (NLP) due to implicit relationships expressed in the texts, the high abstraction level and required background knowledge usually exclusive to experts. The field of Legal Artificial Intelligence is currently divided: On one hand, there are well-understood techniques, ranging from rule-based reasoning in ontologies to long time-studied machine learning techniques that are interpretable for the most part. On the other hand, there are newer emerging techniques which may lack interpretability, yet achieve high accuracy on several tasks, including information retrieval and classification. With the latest developments in Deep Learning, a lot of attention has been paid to contextual word embeddings. This trend has also been noticeable in the COLIEE competition of 2019, where contextual word embeddings created by BERT models have been employed by several of the participating teams [14]. The COLIEE competition includes four tasks, and we focus on Task 3 and Task 4, being statutory law retrieval and on the English version of the Japanese Civil Code and textual entailment classification on the same premise / hypothesis pairs. Since the deep learning approaches last year were not believed to solve tasks beyond syntactic similarity, we decided to combine deep learning with traditional approaches. We

had two groups participate in the competiton: One group (OvGU) focusing on a deep learning approach combined with traditional NLP features and another group (HONto) training classic machine learning algorithms with NLP features joint with word embeddings, usually applied in deep learning.

2 Related Work

Regarding the retrieval task, there are many traditional features which have proven to work well in the past. Using the Term Frequency - Inverse Document Frequency (TF-IDF) weighted cosine similarity is a common choice in combination with Support Vector Machines (SVMs), which can be seen for example in the contribution by Nguyen et al. [17]. For this reason, the OvGU group took traditional methods as a basis for their experiments. Nanda et al. used the topic model Latent Dirichlet Allocation (LDA) with partial string matching [12]. Probabilistic topic modeling captures semantic relatedness, which is useful for determining the relevance of a document to a query. We decided to combine it with local semantic similarity obtained from word embeddings and thus used a hybrid method for the retrieval task in the HONto group. Similar to Task 3, we proceed to review new and traditional methods for the textual entailment task.

Negations as well as conditions, conclusions and exceptions are vital features for the entailment task, for example as sucessfully performed by Kim et al. over the years [9,10]. Commonly those approaches include a method of chunking the text sequence in smaller parts and a lookup in dictionaries for the respective category. Regarding negations, they define a feature called negation level which is a binary flag switching depending on the even / odd number of occurrences of the negation words in the regarded category between premise and hypothesis. We adapt our own version from their proposed method within the HONto group. Akiba et al. describe other features for textual entailment, including n-gram matches and semantic dependency relation matches with the latter based on modifier-modifiee pairs of so-called Bunsetsu chunks [1]. Those chunks are the smallest units returned from a Japanese language parser, and we took inspiration from this feature and created a similar one with the HONto group. In addition to simple match features between the words, also the parse tree edit distance can be considered, as in the work by Montejo et al. [11]. Regarding works using deep learning together with traditional NLP features on legal text, we find the work by Nguyen et al. [13], who use BiLSTM-CRF models jointly with POS tags.

3 Legal Statute Retrieval Task

3.1 OvGU Group

The OvGU group implemented a legal retrieval system by analyzing the relevance between the queries and the relevant statute laws and implemented conventional models such as TF-IDF and BM25 [16], which is an improved version of TF-IDF that takes into consideration the term frequency saturation and document length for relevance ranking with various features.

3.1.1 Data Analysis To understand the property and nature of the corpus we carried out a manual and statistical analysis on the provided English translated corpus of Japanese statute laws. The manual analysis was done on the ground truth corpus which contains the query and the relevant statute laws to understand the features that relate both the queries and statute laws. The statistical analysis was conducted to find the average number of relevant articles per query, identifying the stopwords in the corpus, identifying the relevance of similar words between the query and relevant statute laws with and without query expansion, identifying the most common bigrams and trigrams, identifying spelling and typing mistakes. From the ground truth corpus the number of relevant statute laws per query was obtained to determine the number of statute laws to be retrieved for each query. The average relevant statute laws per query is 1.92 which indicates most of the queries have 1 or 2 relevant statute laws. 94% of the queries have two or less than two relevant statute laws and 6% have three or more relevant statute laws. An n-gram analysis was also performed to find the most commonly occurring bigrams and trigrams in the queries and statute laws for understanding the significance of n-grams.

3.1.2 Data Preprocessing The statute laws and the queries consisted of tokens that did not carry much information for the retrieval task such as the stopwords and punctuation. The stopwords were a combination of words which were obtained from the analysis and the NLTK package [3]. After stopword removal, the tokens are converted into their respective lemmas using the spaCy package [8]. The preprocessed corpus is then indexed for the retrieval task.

3.1.3 Feature selection For the task of information retrieval, we have considered the conventional linguistic features available for identifying the similarity between the queries and the statute laws. The data analysis provided the direction for the selection of features by indicating the relevance by using lemmas and n-grams. The features considered are lemmas, bigrams, trigrams, expansion of the queries and the combination of n-grams, lemmas and expanded queries.

3.1.4 Implementation The TF-IDF model was implemented using the gensim Python library [15]. The preprocessed data was used to create a corpus and the TFIDF model was created and indexed the corpus for search functionality. The BM25 model was implemented using a Python library named rank-bm25[3]. Both models were tested with above mentioned features and different parameters on each run to observe the performance of the models with varied numbers of retrieved documents. We included our top findings in the Results Section.

3.2 HONto Group

The HONto group built a retrieval system based on the Hierarchical Optimal Topic Transport (HOTT) metric proposed by Yurochkin et al. [19]. This meta-

[3] `https://pypi.org/project/rank-bm25/`

distance combines probabilistic topic modeling (Latent Dirichlet Allocation) with word embeddings. The model assigns a weight of each topic to a document. Those topics, in turn, contain a probability distribution over the corpus vocabulary. After truncating the distance measurement to the top 20 topic words, the 1-Wasserstein distance is computed between those words and re-weighted by the individual topic's contribution to the document. Yurochkin et al. formalize HOTT as follows:

$$HOTT(d^1, d^2) = W_1\left(\sum_{k=1}^{|T|} \bar{d}_k^1 \delta_{t_k}, \bar{d}_k^2 \delta_{t_k}\right), \tag{1}$$

with $\bar{d}^1$ and $\bar{d}^2$ being the document-topic distributions of two documents, re-weighting the 1-Wasserstein distance between the topic-word probability distributions t_k, that are truncated to the top n topic words. In that regard, the HOTT metric is more interpretable and computationally more efficient than the Word Mover's Distance between all word embeddings of two text sequences. Mostly for the interpretability aspect we decide to test HOTT on the statutory law retrieval task. For the word embeddings, we have three alternatives: Law2Vec by Chalkidis and Kampas [5], custom fine-tuned Law2Vec, and 300-dimensional gloVe embeddings. Furthermore, we test TF-IDF re-weighting on HOTT.

4 Textual Entailment Task

The textual entailment task consists of two text sequences: the premise and the hypothesis. In this case the premise is the article from the Japanese Civil Code and the hypothesis is some statement which may have a semantic relationship with the premise. The task is to perform a binary classification for a positive or negative entailment relationship between both texts. This task requires problem solving abilities in multiple problem categories, with most instances relating to conditions, person roles and person relationships. We approach this task from two different angles with our OvGU and HONto groups. The OvGU group implemented a deep learning approach with linguistic features, similarity measure and negation for the textual entailment task which is explained as follows.

4.1 OvGU Group

4.1.1 Data Analysis A basic exploration of the data set was done. There were 716 instances with an almost equal number of instances distributed among the two classes of Entailment yes ("Y") and no ("N"). The length of the premise and the hypothesis varied across instances. The average length of the premise is 97.35 words. The average length of the hypothesis is 40.64 words. During the exploration of the data set, there were a few spelling mistakes in the data set. There was also a noticeable pattern of the most frequently used words and trigrams in premises and hypothesis. Additionally, we also calculated the distance (Word Mover's Distance) between the premise and the hypothesis, and noted the distribution for the class labels. We infer that there might be a correlation between lower distance and the class label "Y".

Table 1. Distribution of Distances for Class Label "Y" and Class Label "N".

Distance	No. of Instances	
	Class "Y"	Class "N"
0.0 - 0.2	17	8
0.2 - 0.4	29	13
0.4 - 0.8	53	34
0.8 - 1.0	89	91
1.0 - inf	95	113
inf	73	99

4.1.2 Data Pre-processing We broke the pre-processing steps into an ensemble of steps. We did preliminary replacements of certain words, validation of each word and in case the validation failed, we tried replacing it with another valid word. A few replacements were done in the text before it was tokenized. The replacement words are provided in the Table 2. Once the preliminary replacements are made, the sentences are tokenized. The tokens are checked for their validity using the Law2Vec [5] model which is a pre-trained word embedding on a large number of legal corpora of 123,066 documents from various public sources in English. In case the word exists in Law2Vec, we assume that this is a valid word. There could be more than one possible combination of two words; we choose the one which occurs the most when comparing it with a bigram list we have manually prepared using legal texts from websites which have literature in the legal domain[4] (e.g.: tutorials, handouts, texts explaining the law). We check if the second word of the split token is "less" or "not". If the second word of the split token is "s", then we just assume that it is meant to be plural. If the word does not exist in Law2Vec, our first assumption is that it is missing the space between two words and has correct spelling. The token is then split into two words, each of which exist as a word in Law2Vec. In case there is no valid replacement of the word, then we assume that there is a spelling mistake. We try to get the correct spelling using the Python package "pyspellchecker"[5]. In case we do not find a replacement, we initialize it to a null vector later in the model. Given the training dataset, we observe that the out-of-vocabulary (OOV) words are less than 0.02% of the total number of tokens.

4.1.3 Experiments

4.1.3.1 Model Architecture We defined a simple Bidirectional LSTM [7] architecture as represented in the Figure 1. We noticed while exploring the dataset that over 87% of the instances had 200 words or less in premise and 80 or less

[4] German Civil Code: `https://www.gesetze-im-internet.de/englisch_bgb/englisch_bgb.html`
Hamburg Legal Guidance: `https://www.hk24.de/en/fairplay?param=fairplay`
USA Laws: `https://casetext.com/`

[5] `https://pypi.org/project/pyspellchecker/`

Table 2. Word / Expression replacements.

Token	Replacement	Description
".."	"."	Double dots to a single dot
"."	" . "	Padded with a space on both sides of the character.
","	""	Removed.
":"	""	Removed.
"/"	" / "	Padded with a space on both sides of the character.
"n't"	"not"	Expanded.
"cannot"	"can not"	Introduced space.
"sublessee"	"sub-lessee"	Same meaning.
"renumeration"	"remuneration"	Typographical error.
"supecify"	"specify"	Typographical error.

words in hypothesis. We took that to be the fixed window to process the premise-hypothesis pairs. After preprocessing, we encode the tokens to the Law2Vec vectors, the premise and hypothesis pair is then formed into a sequence. Each sequence has 4 special tags: ⟨PAD⟩, ⟨BOS⟩, ⟨EOS⟩, ⟨SEP⟩. Each of these tags are provided with embeddings. ⟨PAD⟩ was used to fit the premise and hypothesis, should it fall short of the fixed length. ⟨BOS⟩, ⟨EOS⟩ were appended at the Begin-Of-Sentence and End-Of-Sentence respectively. ⟨SEP⟩ was appended between the premise and the hypothesis to mark the end of the premise and the start of the hypothesis, following the work of Bowman et al. [4]. For the binary classification problem of recognizing textual entailment, we train the model with a softmax output layer. The model's complexity was kept low to tackle the issue of overfitting which was encountered with an increase in LSTM cells, hidden layers and batch size. We trained the model on an 80/20 split of train-validation split. We used a single layer of BiLSTM with 256 cells, batch size of 64, Adam optimizer with a learning rate of 1e-05 and cross-entropy loss. Also, we encountered gradient explosion by keeping track of validation loss which was tackled by introducing dropouts of 50% for each BiLSTM layer, L2 regularization over the network weights and gradient clipping. Each of these techniques was introduced through an exhaustive search to ensure minimal regularization was used. We used the early-stopping technique to stop training when the performance on the validation set starts to degrade. The architecture and hyperparameters defined for the baseline are the same for the remaining models we introduce in this report. The same random seed split was used for all the models for which a better performance was recorded. The difference lies in the features and the way they are introduced and encoded to this baseline. The network is implemented and trained using Python 3 on Tensorflow 1.0 framework.

4.1.3.2 Addition of the POS tag The first feature we introduce is information about the parts of speech (POS) of each word as a one-hot encoded vector, which is represented in the Figure 2. For the implementation of POS tags, we used the NLTK library [3] implementation to map the tagset of Penn Tree Bank (PTM) corpus to universal tagset retrieving 12 categories.

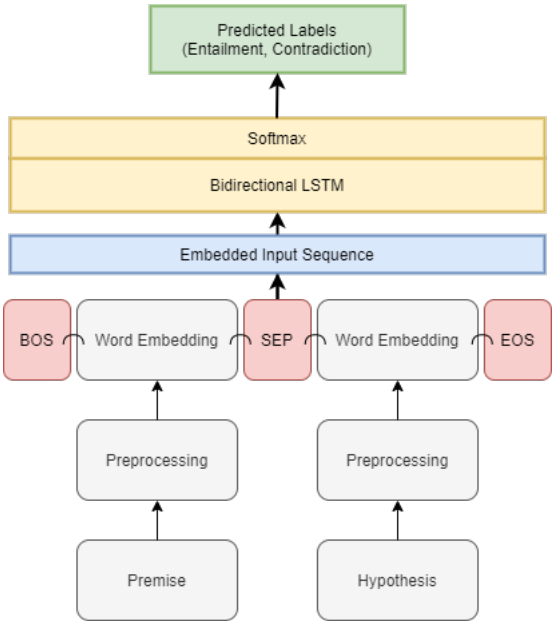


Fig. 1. Baseline Model Architecture.

4.1.3.3 Addition of the Similarity and Negation indication One-hot encoded vectors for intervals of 0.2 in the similarity score and a one-hot encoded vector for the presence of a negation word in premise and/or hypothesis were sent to the model as an input as represented in the Figure 3.

4.1.3.4 Addition of the POS Tag and Similarity and Negation indication In this, we sent the POS tag information in one-hot encoding, as well as the information about similarity and presence of the negation word as described in the previous two sections, as represented in the Figure 4.

4.1.3.5 Addition of the Attention Layer The attention mechanism proposed is inspired by the work of Peng Zhou's attention-based BiLSTM network for relation classification [20] and the original additive attention used in the neural machine translation [2]. The attention mechanism was integrated with the baseline model by adding an attention layer after the BiLSTM layer to produce a context vector that attended to the significant words in the whole input sequence rather than considering the final output state of the LSTM cell.

4.2 HONto Group

For textual entailment, the HONto group reviewed related work, especially from previous COLIEE editions and found several promising concepts in order to capture semantic meaning of the text in features. Common techniques we found

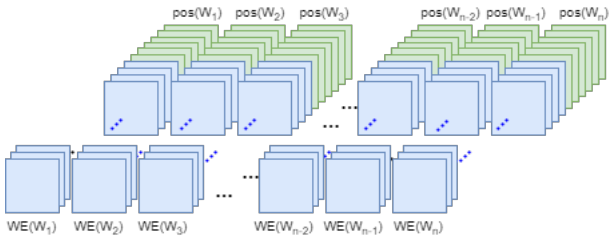


Fig. 2. Block diagram of the input sequence with POS tags.

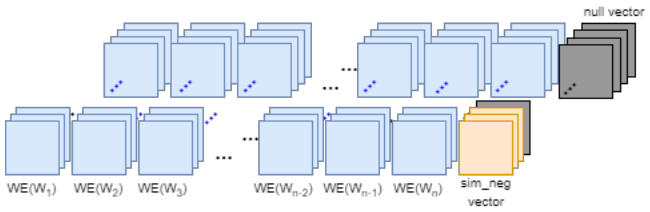


Fig. 3. Block diagram of the input sequence with Similarity and Negation.

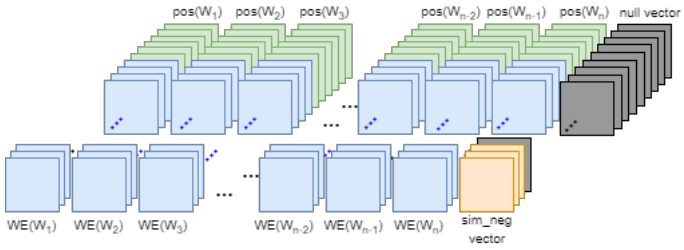


Fig. 4. Block diagram of the input sequence with POS, Similarity and Negation.

are dependency parsing, modeling of condition / conclusion / exception, modifier - modifiee relations and n-gram similarity. In addition to several text similarity features, we feed all extracted features to a Support Vector Machine (SVM) classifier. In detail, we first tokenize, sentence chunk, POS tag, and parse for dependencies in the premise and hypothesis with the Python library *spaCy*. Then, we add the cosine similarity for the TF-IDF representation of premise and hypothesis as a feature to the model. We consider a model variant with only unigrams as features (*tfidf_1*) and then another configuration with similarity scores from only unigrams, bigrams (*tfidf_2*) and trigrams (*tfidf_3*), respectively. We let a legal expert manually highlight the most important passages of five pairs in the dataset, according to what is subjectively relevant for his decision about the entailment relationship. Furthermore, a brief reasoning was added by the expert. The highlighted passages consist frequently of trigrams, other recurring patterns cannot be found in the small sample size.

In terms of further features, we include an overlap feature for the closest chunks of premise and hypothesis according to the cosine similarity (*overlap_spo*) and Word Mover's Distance (*chunk_wmd*), respectively, both performed on the chunks weighted by TF-IDF. The chunks are formed from subjects, predicates and objects, which are identified using the dependency parser. The dependency tags also include a negation marker, which we use for negating the subsequent word with an exclamation mark. Hence, overlapping terms in premise and hypothesis need to be both negated or both without negation prefixes in order to match. We define another set of six overlap features inspired by the approach taken by Kim et al. [9,10]. Thus, we compare the overlap of condition, conclusion and exception chunks separately and each of those scores becomes a feature (*overlap_con*, *overlap_conc*, *overlap_exp*), accompanied by a binary flag indicating the presence of the category in the given pair. We are using a custom lookup dictionary for each category which we formed by inspecting parts of the training data. Here, the chunks are created by searching for occurrences of each category and starting the span after a category term and ending in case another category's term occurs in the text sequence. The *negation_level* feature is re-implemented as described by Kim et al., see Section 2. The next feature is the WMD between the whole premise and hypothesis (*whole_wmd*). Following, we add the longest common subsequence feature *lcs*. The next feature is the token sort similarity from the *fuzzywuzzy* library [6]. This similarity is based on the edit distance and order insensitive, but allows for substring matching while respecting token boundaries. The last feature we employed is the overlap between modifier - modifiee relationships between premise and hypothesis (*overlap_mod*). We determined the pairs using the dependency parser, making the modifier be either a subject or a passive object, having a noun POS tag and a verb as the head of the dependency. The modifiee is an object or a passive subject, having the same POS and head category as the modifier. To summarize, our features consist of: *tfidf_1, tfidf_2, tfidf_3, overlap_spo, chunk_wmd, overlap_con, negation_level, whole_wmd, lcs, fuzzywuzzy, overlap_mod*.

We train two different types of SVM models (RBF / linear kernel) and include different features in each run. For the SVM models, we use scikit-learn[6].

Table 3. Dependency tags used for assigning SPO labels.

Subject	Predicate	Object
`nsubj, csubj, pobj, agent, expl`	`ROOT, aux, auxpass, ccomp, advcl, acl`	`obj, dobj, nsubjpass, csubjpass, dative, attr, oprd, npadvmod`

5 Results

5.1 Legal Statute Retrieval Results

5.1.1 OvGU Group The performance of the legal information retrieval system is evaluated on metrics such as precision, recall and F2-score. From the Table 4, the TF-IDF model with lemmas and combination of lemmas, query expansion and n-grams had similar F2-scores indicating that the addition of features did not yield improvement in the performance of the TF-IDF model. Instead, the BM25 model with lemmas alone was able to outperform the models with other features. Based on these results TF-IDF with lemmas, BM25 with lemmas and combined features were submitted.

Table 4. Performance of the OvGU Retrieval models.

Model	Features	Precision	Recall	F2
TF-IDF	Tokens	0.28839	0.41065	0.37855
TF-IDF	**Lemma**	0.34861	0.44629	**0.42260**
TF-IDF	Query Expansion	0.29245	0.44728	0.40445
TF-IDF	Bigrams	0.34922	0.39844	0.38752
TF-IDF	Trigrams	0.28947	0.32963	0.32073
TF-IDF	Bigrams + Trigrams + Query Expansion	0.33772	0.45504	0.42548
BM25	Tokens	0.29759	0.48057	0.42795
BM25	**Lemma**	0.33480	0.50610	**0.45912**
BM25	Query Expansion	0.25371	0.47391	0.40382
BM25	Bigrams	0.33938	0.41509	0.39736
BM25	Trigrams	0.26306	0.31853	0.30564
BM25	**Bigrams + Trigrams + Query Expansion**	0.34089	0.46725	**0.43500**

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

5.1.2 HONto Group Based on the training data, we selected three models which yielded an F2-score of 47%. Unfortunately, we had a mistake in our implementation for the test run, which heavily affected the ranking scores of our models. Table 5 lists the official competition results first, and second, the corrected results reflecting the actual model performance in boldface on the test data with micro averages. The best result after the correction has been achieved by the model with TF-IDF re-ranking of the top 100 HOTT results with 100 topics trained on the LDA. The other two methods were trained on 200 topics and are called hybrid because their relevance score was the HOTT metric re-weighted by the TF-IDF cosine distance. All models except *HONto_hybrid_ft* were trained using Law2Vec, whereas for *HONto_hybrid_ft*, we fine-tuned the Law2Vec model on the COLIEE training data, the English versions of the Japanese and German Civil Codes and on further literature on the Japanese civil code, consisting of 27 textbooks / online commentary. The observations raise questions regarding fitting the topic model on external resources. So far, the HOTT score has not outperformed the TF-IDF ranking. Further tests with 1000 topics did not improve results, so we conclude that HOTT might be more effective on larger documents and corpora.

Table 5. Performance of the HONto Retrieval models.

Model	F2	Precision	Recall
HONto_hybrid	0.2822	0.2545	0.2991
HONto_hybrid_ft	0.2822	0.2545	0.2991
HONto_tfidf	0.2114	0.1667	0.2411
HONto_hybrid	0.3878	0.3456	0.4000
HONto_hybrid_ft	0.3878	0.3456	0.4000
HONto_tfidf	0.4480	0.3417	0.4857

5.2 Textual Entailment Results

5.2.1 OvGU Group The performance of the models over the validation set and test set are discussed in this section. The SimNeg model and POS_SimNeg model performances are very comparable and are significantly higher than the baseline BiLSTM model without any additional features, when validation accuracy is taken into account. The BaselinewAtt & POS_SimNegwAtt has a significant improvement when recall is taken into consideration. The POS_SimNegwAtt model performs the best on validation data with an F1-score of 0.66. Although this model classifies the 'entailment' well (Recall: 0.83), it suffers from misclassifying 'contradiction' classes as 'entailment'. When the model is run on the test data set, we observed that we cannot reproduce results similar to results of the validation set. The POS model performs marginally better than the SimNeg model and POS_SimNeg model, when we take accuracy into consideration. SimNeg model and POS_SimNeg model performances are comparable but lower

than what was achieved in the validation set. When F1-score is taken into consideration, the POS_SimNegwAtt model and BaselinewAtt perform better than the rest. Based on the results presented on Table 6 and 7, we chose SimNeg, POS_SimNeg and BaselinewAtt models.

Table 6. Performance of all models based on the validation set.

Model	Val Accuracy	Val Loss	Precision	Recall	F1
Baseline	0.54	0.6911	0.611	0.17	0.266
POS	0.54	0.6905	0.5714	0.2580	0.355
SimNeg	0.667	0.6843	0.7083	0.5483	0.618
POS_SimNeg	0.683	0.6855	0.7391	0.5483	0.629
BaselinewAtt	0.563	0.6867	0.5492	0.6290	0.5864
POSwAtt	0.563	0.6863	0.5663	0.4838	0.5217
SimNegwAtt	0.563	0.6836	0.5555	0.5645	0.5599
POS_SimNegwAtt	0.587	0.6879	0.5531	0.8387	0.6666

Table 7. Performance of the OvGU models based on the test set.

Model	Test Accuracy	Test Loss	Precision	Recall	F1
Baseline	0.5918	0.6891	0.888	0.1702	0.2857
POS	0.6122	0.6869	0.909	0.212	0.3448
SimNeg	0.5918	0.6843	0.64	0.3404	0.4444
POS_SimNeg	0.5816	0.6871	0.6153	0.3404	0.4383
BaselinewAtt	0.5510	0.6905	0.5333	0.5106	0.5217
POSwAtt	0.5714	0.6891	0.5862	0.3617	0.4473
SimNegwAtt	0.5408	0.6825	0.5312	0.3617	0.4303
POS_SimNegwAtt	0.5204	0.6928	0.5	0.5957	0.5436

5.2.2 HONto Group Our HONto Group recorded three scores while evaluating their SVM-based approach. The first one is the Fit Accuracy, referring to the score achieved when fitting the training data and assessing the performance on all training data. This is an optimistic estimate for the performance on the test set. The second score is the validation accuracy, where a subset of the training data H18-H29 is seen by the model during training, while the H30 set is used for validation. This is a pessimistic estimate of the performance due to the reduction of the training sample size. The last score is the Test Accuracy on the test data. Our best model is the linearsvm with all regular features, except fuzzywuzzy for conclusions and exceptions. The validation accuracy during training is 51.4%, at the test run we obtain a score of 56.25%. We note that the Fit and Validation Accuracy of the model with all features called linearsvm_all model is surprisingly high, given the performance shown on the test data. The conclusion and exception fuzzywuzzy scores probably are not informative and lead to

Table 8. Performance of the HONto models, the boldfaced were submitted as runs.

Model	Fit Accuracy	Validation Accuracy	Test Accuracy
linearsvm_no_ngram_nofuzz	0.575	0.443	0.5089
linearsvm_no_ngram	0.564	0.514	0.5536
linearsvm_nofuzz	0.567	0.471	0.4911
linearsvm	0.567	0.514	0.5625
linearsvm_all	0.580	0.586	0.5179
rbfsvm_no_ngram_nofuzz	0.567	0.486	0.5000
rbfsvm_no_ngram	0.535	0.500	0.5268
rbfsvm_nofuzz	0.569	0.514	0.4911
rbfsvm	0.536	0.500	0.5268

overfitting. Interestingly, the SVM model without any fuzzywuzzy and n-gram feature exhibits similar behavior. Leaving out the fuzzywuzzy features causes the model performance to deteriorate despite having n-gram features along with all other standard features. We deduce that condition chunks can benefit from fuzzywuzzy, which is matching the shorter string of the pair, regardless of order. The performance of the RBF-kernel SVMs is mostly inferior to the linear SVM kernel, which is a common finding on text data. Since the linear SVM achieved the best score, we investigated feature importance from the model's coefficients. The five most important features were *overlap_mod*, *overlap_spo*, *chunk_wmd*, *lcs*, *whole_wmd*. These features are followed by the overlap features for condition, conclusion and exception. The next important feature is the trigram similarity, and the last remarkable impact stems from the fuzzywuzzy similarity scores between the conditions of premise and hypothesis, respectively. The negation feature and fuzzywuzzy similarity scores between conclusion and exception had the least impact. Given the observations on the feature importance, the WMD as well as the semantic features from the dependency parsing had the highest impact on the final prediction, which lets us conclude that both deep learning features as well as traditional NLP features were complementary in our experiments.

6 Conclusion

We approached the competition from two different angles: On one hand, we employed established machine learning techniques and on the other hand, we chose deep learning-based approaches. Overall, the best performance on the retrieval task was achieved by BM25 model on lemmas as features. For the entailment task, the linear SVM with fuzzywuzzy similarity and n-grams achieved the best score from all our submissions. We conclude that similarity and semantic features are helpful to solve the tasks. Given the average performance of our contributions, we seek to explore the further potential of these techniques. One avenue for future work is the use of external knowledge from ontologies or further related documents, as Taniguchi et al. have demonstrated by their use of FrameNet [18].

References

1. Akiba, Y., Taira, H., Fujita, S., Kasahara, K., Nagata, M.: Nttcs textual entailment recognition system for ntcir-9 rite. In: NTCIR (2011)
2. Bahdanau, D., et al.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015)
3. Bird, S., et al.: Natural Language ToolKit (NLTK) Book (2009)
4. Bowman, S.R., et al.: A large annotated corpus for learning natural language inference. In: Conference Proceedings - EMNLP 2015 (2015)
5. Chalkidis, I., et al.: Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artificial Intelligence and Law (2019)
6. Cohen, A.: Fuzzywuzzy: Fuzzy string matching in python (2011)
7. Hochreiter, S., et al.: Long Short-Term Memory. Neural Computation (1997)
8. Honnibal, M., et al.: spaCy2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. (2017)
9. Kim, M.Y., et al.: Answering yes/no questions in legal bar exams. In: JSAI International Symposium on Artificial Intelligence. pp. 199–213. Springer (2013)
10. Kim, M.Y., et al.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 283–289 (2019)
11. Montejo-Ráez, A., et al.: Combining lexical-syntactic information with machine learning for recognizing textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 78–82 (2007)
12. Nanda, R., et al.: Legal information retrieval using topic clustering and neural networks. In: COLIEE@ICAIL. pp. 68–78 (2017)
13. Nguyen, T.S., et al.: Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. Artificial Intelligence and Law **26**(2), 169–199 (2018)
14. Rabelo, J., et al.: A summary of the coliee 2019 competition.
15. Rehurek, R., et al.: Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks
16. Robertson, S., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval (2009)
17. Son, N.T., et al.: Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models. In: Satoh, K., Kim, M.Y., Kano, Y., Goebel, R., Oliveira, T. (eds.) COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment. EPiC Series in Computing, vol. 47, pp. 31–42
18. Taniguchi, R., et al.: Legal question answering system using framenet. In: JSAI International Symposium on Artificial Intelligence. pp. 193–206. Springer (2018)
19. Yurochkin, M., et al.: Hierarchical optimal transport for document representation. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 1601–1611. Curran Associates, Inc. (2019)
20. Zhou, P., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers (2016)

Paragraph Similarity Scoring and Fine-Tuned BERT for Legal Information Retrieval and Entailment

Hannes Westermann[1], Jaromir Savelka[2], and Karim Benyekhlef[1]

[1] Cyberjustice Laboratory, Faculté de droit, Université de Montréal, Montréal (Québec), H3T 1J7, Canada
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract. The assessment of relevance of legal documents and the effects of the application of legal rules embodied in legal documents are some of the key processes in the field of law. In this paper, we present our approach to the 2020 Competition on Legal Information Extraction/ Entailment, which provides researchers with the opportunity to find ways of accomplishing these complex tasks using computers. Here, we describe the methodology used to build the models for the four tasks and the results of their application. For Task 1, concerning the prediction of whether a base case cites a candidate case, we devised a method for evaluating similarity between cases based on individual paragraph similarity. This method can be used to reduce the number of candidate cases by 85%, while maintaining over 80% if the cited cases. We then train a Support Vector Machine to make the final prediction. The model is the best solution submitted for Task 1. We use a similar method for Task 2. For Task 3, we use an approach based on BM25 measure in combination with the identification of previously asked questions. For Task 4, we use a transformer model fine-tuned on existing entailment data sets as well as on the provided domain specific statutory law data set.

Keywords: Legal Entailment · Universal Sentence Encoder · Approximate Nearest Neighbor · Support Vector Machine · BM25 · BERT.

1 Introduction

Assessing the relevance relationship between textual documents is one of the key skills used in the legal profession. Judges and attorneys need to know which previous legal case is relevant to a current case, in order to craft their argument and decide on the merits of a case. Law students need to assess whether a piece of legislation is relevant to a fact pattern in an exam, and determine the effect of the legislation on which answer they should give. Far from being simple, this assessment of relevance can be very difficult, requiring complex analysis and determinations. (Semi-)Automating the tasks could have tremendous implications for legal profession and support judges, lawyers, researchers, student and the public in their use and understanding of the law.

The focal point for researchers to develop and compare their efforts in understanding and comparing legal texts is the Competition on Legal Information Extraction/Entailment (COLIEE). The competition consists of 4 tasks. Task 1 is the Legal Case Retrieval Task, concerned with determining which of 200 candidate cases are cited by a base case. Task 2, the Legal Case Entailment task, focuses on which paragraph in a cited case is cited by a specific fragment in a base case. Task 3, The Statute Law Retrieval Task, focuses on the retrieval of relevant legal articles for a questions in a bar exam. Task 4 focuses on providing the answer to bar exam questions, using the relevant sections of the law. We work with a number of methods and their combinations, including sentence encoding models based on deep neural networks, Support Vector Machines (SVM), nearest neighbour searches, rankings based on BM25 measure, and fine-tuned language models (BERT).

Our placements on the team level in the four challenges are:

- Task 1: **1st Place**
- Task 2: 4th Place
- Task 3: 3rd Place
- Task 4: 4th Place (shared)

The paper is structured in such a way that we go through the tasks one by one. For each task, we provide a short description, explain our methodology, and present and briefly discuss our results.

2 Task 1 - The Legal Case Retrieval Task

2.1 Task Description

The goal of Task 1 is to identify cases that are noticed (e.g. cited) by a base case. The data consists of 650 samples, each consisting of one base case and 200 candidate cases. The aim of the task is to identify which of the 200 cases was noticed by the judge in the base case.

From the classification perspective the data set is heavily unbalanced. Only around 2.5% of the provided candidate cases are noticed by a base case. Imbalanced datasets present a difficult problem for most machine learning methods. There is a number of approaches to tackle this issue, such as under-sampling and cost-sensitive learning. [12] This issue has been discussed in papers submitted for previous COLIEE editions. (see e.g. [6]).

Our solution is based on making the dataset more balanced via preliminary filtering. The filter excludes 85% of the candidate cases most of which are negative examples. On the remaining 15%, we train a traditional ML classifier to generate the final prediction.

2.2 Methodology

Our approach consists of three steps—data pre-processing, the preliminary neural filter that narrows the space of candidate matches from 200 to 30, and the

final Support Vector Machine (SVM) model that makes the final prediction as to which case is noticed.

The approach relies on several insights about the dataset. These are:

- Noticed cases might only be relevant to a single paragraph in the base case. Conversely, only a single paragraph in the noticed case might be relevant for the base case. Therefore, it is advantageous to work on the paragraph level. (Compare [27] which models interactions between paragraphs using BERT)
- Paragraphs in a certain section of the base case are more likely to cite other cases.

Pre-processing Each case is split into paragraphs. Each paragraph is turned into its vector representation, using a Universal Sentence Encoder. The model relies on deep transformer networks, trained on multiple tasks with the aim of being used for transfer learning. [2] We use the pre-trained implementation from tf-hub. [3] For each case paragraph the model produces a 512-dimension vector. The model has been used in previous COLIEE editions (e.g. [20], [17]).

Next, we identify the paragraphs that are the most similar to other paragraphs. For this, we use the Spotify Annoy [4] library, which implements a fast and scalable Approximate Nearest Neighbors algorithm. For each sample, we create a search tree for all paragraphs contained in the 200 candidate cases.

Paragraph Similarity Score The next step is to filter out cases that are unlikely to be noticed. The aim is to employ a low-precision filter with high recall. This enables us to create a more balanced sample suitable for traditional prediction. As discussed above, comparing the entire text between the base case and the candidate case at this stage might not give the desired results, as only certain paragraphs in the base case might be relevant for the candidate case, and vice-versa. Therefore, we identify the candidate cases with the most paragraphs that are similar to paragraphs in the base case.

First, we identify the section of the case where the judge is likely to cite other cases, by taking paragraphs with keywords indicating suppressed citations and a few surrounding paragraphs. The paragraphs with suppressed citations are used as a proxy to identify something akin to the "reasoning" section of a case, where the judge applies the law to a factual situation. This is the area where the judge is likely to refer to jurisprudence to explain a decision. While the use of keywords for suppressed citations is specific to the COLIEE dataset, it could be replaced by a method scanning for normal citations in a real-world dataset.

We step through these paragraphs and for each we identify the 10 most semantically similar paragraphs, across all the paragraphs in the candidate cases, using the nearest neighbor method and sentence encoders described above. Whenever a paragraph of a candidate case appears in the top 10 most similar

[3] tfhub.dev/google/universal-sentence-encoder/4
[4] github.com/spotify/annoy

	Train			Eval			Test		
	P	R	F1	P	R	F1	P	R	F1
S@1	.69	.13	.22	.75	.14	.24	.75	.15	.25
S@7	.37	.51	.43	.45	.60	.51	.42	.61	.50
S@30	.14	.81	.24	.16	.91	.27	.15	.93	.26
Run 1				.79	.48	.59	.81	.54	.6503
Run 2				.72	.66	.69	.69	.66	**.6774**
Run 3				.72	.65	.68	.69	.66	.6768

Table 1. Results for runs using only the similarity score (S1, S7 and S30), and the official runs using the similarity score and support vector machine.

paragraphs for a paragraph in the base case, we increase the score for that candidate case by an amount corresponding to the euclidian proximity of the two paragraphs. This aims to create a many-to-many comparison of paragraphs in the base case and candidate cases. The 30 candidate cases with the highest similarity score are retrieved for subsequent processing.

Support Vector Machine As the next step, a traditional ML model is trained to determine whether one of the 30 candidate cases is noticed or not. First, we transform the base cases and candidate cases to a single vector, by taking a Bag of Word representation of words and bigrams and running them through a Term Frequency and Inverse Document Frequency (TF-IDF) transformer. This method aims to determine the relevancy of terms by weighting them depending on how often they appear in a document and across the corpus. We use the implementation from the python library scikit-learn [18]. We cap the maximum amount of features at 6000. Next, we subtract the vector for the base case and the candidate case from each other.

Finally, we train an SVM model. SVM maps examples with different labels into a high-dimensional space, to find clear divisions between examples of different classes. [5] We use the SVM implementation from scikit-learn [18], using the "rbf" kernel. Further, we balance the predictions by weighting the classes in Run 2 and Run 3. The output of this model is the final prediction, with all cases outside of the top 30 being set to negative. The case with the highest similarity score is always predicted as positive.

2.3 Results

Table 1 shows the results of a hypothetical scenario where only the similarity measure is used for prediction (S1, S7 and S30), and the final three runs we submitted. We split the data into the following partitions: Train (Sample 1 - 449), Eval (Sample 450 - 520), Test (Sample 521 - 650) set. For the evaluation of the Eval split, the model is trained on the training split. For the evaluation of the test sample, Train and Eval data is used.

Specifically, the Table presents the following results:

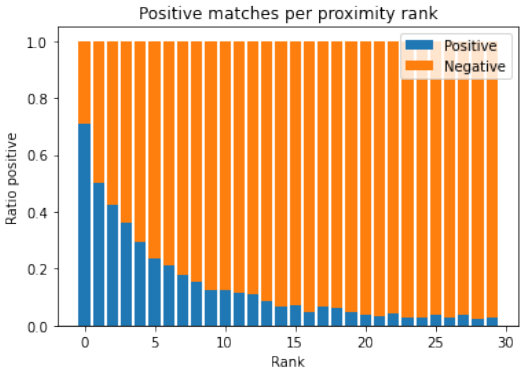


Fig. 1. Positive examples at different ranks as determined by paragraph similarity assessment.

- S@1 - The hypothetical results of predicting the most similar case only as noticed.
- S@7 - Hypothetical results of predicting the top seven most similar cases as positive. This is the highest-scoring approach that uses the similarity score only.
- S@30 - Hypothetical results of using the top 30 most similar cases. These 30 cases are used by the SVM for the three submitted runs.
- Final run 1 - Support Vector Machine, without balancing via class-weights.
- Final run 2 - The same as run 1, with balancing via class-weights.
- Final run 3 - The same as run 2 using TF-IDF transformer trained on all available text, including the text of test samples.

Figure 1 shows the distribution of positive and negative examples along the axis of the cases with the highest similarity score. The x-axis shows the rank of the candidate case, while the y-axis shows the percentage of cases at that rank that are positive (e.g. the candidate case is noticed) vs negative (e.g. the candidate case is not noticed).

2.4 Discussion

The proposed architecture was the best performing one submitted for COLIEE 2020 Task 1 (the winning model).

The main problem to overcome in this task is the unbalanced dataset (from the classification perspective). As only 2.5% of the examples are positive, classification models trained on the full dataset tend to only predict the negative class. In order to overcome this, we developed the paragraph comparison to identify relevant cases, by comparing paragraphs from the base case to paragraphs from the candidate cases, The comparison results in picking the 30 closest cases. This reduces the size of the dataset by 85%, while retaining 80% to 90% of the positive examples, meaning that around 15% of the data consist of positive cases. As can

be seen in figure 1, the similarity scoring does a good job of placing cases that are likely to be noticed at the highest positions in the ranking. The candidate case ranked as the closest is around 70% likely to be noticed.

After we have selected the 30 top highest ranked cases using the similarity score, we train an SVM model on the TF-IDF vector between the base case and the noticed case.

The balancing of the classes improved the performance of the model significantly, as can be seen by the strong performance of Run 2 and Run 3, which scored first and second of all the submitted runs. Pre-training the TF-IDF vectorizer on the test samples texts gave no performance improvement, which could indicate that the train and test data has a similar distribution in terms of topics.

There are a number of ways this model could be improved. It currently uses entire cases to determine whether one case noticed another case in the final prediction step. This works well, however it is possible that the performance could be improved by focusing on similar sections. Furthermore, the use of neural networks and language models could improve performance. These tend to have trouble with long sequences, such as the cases used in Task 1, however recently models that are able to handle thousands of tokens have emerged. (e.g. [11])

3 Task 2 - The Legal Case Entailment Task

3.1 Task Description

Task 2 is somewhat similar to Task 1, in that it concerns the detection of citations from one case to another. However, while Task 1 focuses on the entire case, Task 2 focuses on paragraphs within a case. As such, the data provided is a base case, an entailed fragment of the base case, and a noticed case, split into its paragraphs. The task is to determine which paragraph from the noticed case is cited by the entailed fragment.

In total, the organizers provided 325 training samples, and 100 test samples. Again, the data is sparse. 85% of the entailed fragments cite only a single paragraph, while 13% cite two paragraphs. Thus, for each pair between the entailed fragment and a candidate cited paragraph, only 3.3% are positive. Just like for Task 1, overcoming the imbalance was the key step in building a well-performing model. The method we use is similar to the one we used for Task 1, except that we based the model on sentences instead of entire paragraphs.

3.2 Methodology

The methodology is very similar to that we use for Task 1. We create a many-to-many sentence comparison method that narrows the potentially matching paragraphs down to the top ten most likely matches. We then train an SVM model to make the final predictions.

Pre-processing Paragraphs are read from the files and grouped by sample. Each paragraph is split into sentences, using the SpaCy library.[5] Each sentence is turned into a 512-dimension vector representation, using the Google Universal Sentence Encoder model. Finally, we use the Annoy library to create a nearest-neighbour search method of all sentences contained in the candidate paragraphs for a specific sample. (see section 2.2 for pointers to the individual methods)

Sentence similarity score Next, we perform a per-sentence comparison between the entailed fragment and the candidate paragraphs, to determine the semantic similarity between the two. We iterate through the sentences in the entailed fragment, and retrieve the ten closest sentences from all sentences in the candidate paragraphs. For each candidate paragraph, we keep track of how often a contained sentence appears in the top ten most similar sentences of the sentences in the entailed fragment. Finally, we select the ten paragraphs with the highest score. (see section 2.2)

Support Vector Machine Classifier We train an SVM model to predict the candidate paragraph that is the most likely to be cited by the entailed fragment. SVM does not return probabilities. However, we use the Platt-scaling implemented in scikit-learn, which trains an additional sigmoid function on top of the SVM model in order to turn the raw scores into probabilities. [19] We train a TF-IDF vectorizer on the case texts, and use it to transform all paragraphs into their TF-IDF representation. We use only unigrams for the vectorization, and cap the number of features at 6000. Next, for each pair of an entailed fragment and a candidate paragraph, we subtract the TF-IDF vector of the candidate paragraph from that of the entailed fragment.

We also introduce two additional features. First, we calculate the average cosine distance between all the sentences in the entailed fragment and the sentences in the candidate paragraph, to capture semantic relatedness beyond vocabulary overlap. Further, we introduce the length of the candidate paragraph as a feature, based on the experimental observation that cited paragraphs are often long.

For prediction we obtain the probability for each pair of entailed fragment and candidate paragraph. We then accumulate these based on the entire sample, and take the pair with the highest probability as the prediction. Further, for Run 2 and 3 we introduced a system that accumulates prediction probabilities for all paragraphs not initially selected. This list is ordered by probability, and the top 10% of these paragraphs are also predicted to be matching, to capture some of the cases where two paragraphs are cited. (see [20] which used post-processed BERT confidence scores to determine the cited fragments).

3.3 Results

We split the data into Train (Samples 1-284), Eval (Sample 285 - 325) and Test (Sample 326 - 425).

[5] https://spacy.io/

	Train			Eval			Test		
	P	R	F1	P	R	F1	P	R	F1
S@1	.44	.39	.41	.60	.49	.54	.48	.38	.43
S@10	.10	.88	.18	.12	.96	.21	.11	.88	.20
Run 1				.75	.61	.67	.63	.50	.5600
Run 2				.70	.61	.65	.63	.54	.5837
Run 3				.74	.65	.70	.63	.55	**.5897**

Table 2. Results for runs using only the similarity score, and the official runs using the similarity score and support vector machines.

Table 2 shows the following model runs:

- S@1 - The hypothetical results of only predicting the most similar paragraph from the sentence-to-sentence comparison as entailed.
- S@10 - The hypothetical results of only predicting the top ten paragraphs from the sentence-to-sentence comparison as entailed. This is further used as input data for the SVM models.
- Final run 1 - Regular run of the similarity many-to-many matchin combined with an SVM model.
- Final run 2 - The same as Run 1, but the TF-IDF vectorizer is fitted on both the training and test data.
- Final run 3 - The same as Run 2, except that the cases between 182 and 225 (corresponding to the test data of COLIEE 2019) are excluded from the training data. Oddly, this particular partition seems to confuse the model (determined heuristically), and we found that training without this data consistently improved the performance.

3.4 Discussion

Overall, the model performs reasonably well, enough to place us at the fourth place. The sentence similarity comparison works well as a way to make the dataset more balanced. When taking the top ten paragraphs, it is able to retain around 90% of the positive examples, while excluding 72% of the candidates.

The final model based on SVM, however, did not perform as well as it did for Task 1. This could potentially be explained by less data being available overall, and each sample (in the form of paragraphs) being shorter than the whole cases used in Task 1, making it more difficult to find a vocabulary overlap.

There are many ways the model could be improved upon. First of all, the spaCy model used to separate the sentences is likely not ideal for case law. Alternatives, e.g. [26], could be used. Further, the SVM model does not seem ideal for detecting entailment between short sequences. Instead, modern language models such as BERT could be used, as in [20]. We tried these, but have not yet been able to use them to outperform SVM. Finally, information from the entire base case could be used additionally to the entailed fragment.

4 Task 3 - Statutory Law Retrieval Task

4.1 Task Description

The task consists of a set of statements that could be either true or false with respect to the provisions of the Japanese civil code translated to English. The civil code is divided into 782 articles. The task is to retrieve one or more relevant articles in response to a statement. An article is considered to be relevant if a query statement can be evaluated as true or false on the basis of the article. As such, the task is an example of an ad hoc document retrieval where a statement is a query and the civil code articles are the documents. The competition organizers provided 696 statements paired with the relevant articles as the training set and 112 statements without the pairing as the test set.[6]

4.2 Methodology

As the first step, we segment the statements and civil code articles into sentences using a sentence segmentation model from [26], which is specifically trained to work well on legal texts. This transforms the data set of 696 statements to 751 statement sentences and the data set of 782 civil code articles to 1,415 article sentences. In the next step, we measure similarity between each statement sentence and each article sentence. For the three runs we submitted we have retrieved $n = \{1, 2, 3\}$ top similar article sentences for each statement sentence. Notably, we have also matched each statement sentence to other statement sentences from the training set. Again, for the three runs we have retrieved $n = \{1, 2, 3\}$ top similar statement sentences for each statement sentence. As the training statements are associated with civil code articles we retrieve the respective articles associated with the retrieved sentence.

The above described method results in the initial set of candidate articles that we subject to subsequent filtering. We first examine the word overlap between the matched sentences. If the ratio of the shared words is higher than 0.7 for at least one of the sentences, the match with the highest ratio is picked as the only result. The other retrieved sentences are discarded. If the first filter is not applicable, the maximum similarity match score is detected (max). Any sentences that were matched with a score that is lower than $0.7 * max$ are discarded. These filters improve precision while causing a rather minimal hit to recall.

Our approach to measuring similarity of statement sentences to article sentences is based on the Okapi BM25 function, from the well known TF-IDF family. The function is defined as follows:

[6] sites.ualberta.ca/ rabelo/COLIEE2020/

	F_2	P	R	MAP	R_5	R_{10}	R_{30}
cyber1	.5290	.5058	.5536	.5540	.5500	.6929	.8000
cyber2	.5251	.4353	.5967	.5540	.5500	.6929	.8000
cyber3	.5175	.4134	.6235	.5540	.5500	.6929	.8000

Table 3. Results of the 3 runs submitted for COLIEE 2020 Task 3 on the test set.

$$\text{BM25} = \sum_{t \in q} TF \cdot IDF \cdot QTF$$

$$\text{TF} = \frac{(k_1 + 1) \cdot tf_{td}}{k_1 \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right) + tf_{td}}$$

$$\text{IDF} = log\left(\frac{N - df_t + \frac{1}{2}}{df_t + \frac{1}{2}}\right) \quad \text{QTF} = \frac{(k_3 + 1) \cdot tf_{tq}}{k_3 + tf_{tq}}$$

Here, N is the size of the collection (i.e., the number of article sentences), df_t is the number of sentences in which t occurs, tf_{td} is the frequency of term t in document d (i.e., an evaluated article sentence), tf_{tq} is the frequency of term t in query q (i.e., an evaluated statement sentence), L_d and L_{avg} are the length of d and the average document length for the whole collection. k_1, k_3 and b are tuning parameters. We use the typical values of $k_1, K_3 = 1.2$ and $b = 0.75$. [14, p. 233] In [15] the authors emphasize that BM25 is an example of a probabilistic approach to IR, [21] where the 2-Poisson model forms the basis for counting term frequency. [8, 22] The idea is that there are two types of documents having term frequencies from different Poisson distributions—documents that are about a term and documents that only mention the term. The goal in IR is to distinguish between the two types. In order to achieve the goal the common version of BM25 considers query terms only, assuming that non-query terms are less useful for document ranking. [15]

4.3 Results

We submitted 3 runs, the technical details of which are described in the preceding section. The best F_2 score of 0.529 was achieved by the model focused on precision. However, the differences among the performance of the three models are minimal as the models are quite similar to each other. The most successful cyber1 model ended up as the third out of the total number of 28 submitted runs, while the cyber2 model ended up fourth and cyber3 model 6th.

4.4 Discussion

The articles are relatively short pieces of text that typically consist of one or at most several sentences. It appears to be well-established that the traditional

approach to ranking, based on computing similarity between the query and retrieved documents, is less effective for very short documents (see, e.g., [16]). Usually the main cause of the decreased performance is the lack of a robust overlap between a query and a document. In larger documents an unusually high occurrence of a term strongly indicates that the document might be "about" the term. This "aboutness" assumption often works surprisingly well for longer documents. In shorter documents there is typically just one exact occurrence of the term. Even in case of more than one occurrence it is questionable if the "aboutness" assumption would still be as solid as for longer documents.

5 Task 4 - Statutory Law Entailment Task

5.1 Task Description

The task uses the same data as Task 3 (Section 4). This means that there is a set of statements that could be either true or false with respect to the provisions of the Japanese civil code (782 articles). Here, the task is to predict if a statement is true or false given a set of relevant articles. Hence, a method developed for Task 3 would constitute a first step in a full question answering (QA) system where a statement would be considered a question. A method developed for this task would then be the second step. The QA system would directly provide a 'yes' or 'no' answer to a question (i.e., a statement). The task is an example of entailment detection where a statement and an article constitute a document pair. The 696 training statements paired with the relevant articles were also equipped with the 'Y' or 'N' label encoding the entailment relation. The relevant articles for the 112 statements from the test set were also provided.

5.2 Methodology

We leverage a bidirectional encoder representation from transformers (BERT) model that has been recently shown as an effective solution for a general entailment task. The SemBERT [29] model is currently reported as the state of the art in the task based on the Stanford Natural Language Inference (SNLI) data set.[7] We employ the base model resulting from the Robustly Optimized BERT Pretraining Approach (RoBERTa base) with 25 million parameters. [13] RoBERTa is using the same architecture as BERT. However, the authors of [13] conducted a replication study of BERT pre-training and found that BERT was significantly undertrained. They used the insights thus gained to propose a better pretraining procedure. Their modifications include longer training with bigger batches and more data, removal of the next sentence prediction objective, training on longer sequences on average (still limited to 512 tokens), and dynamic changing of the masking pattern applied to the training data. [13]

We fine-tune the pre-trained RoBERTa base model on the task of sentence pair classification. First, we attempted to fine-tune the model directly on the

[7] https://nlp.stanford.edu/projects/snli/

	Correct	Accuracy
cyber	69	.6161

Table 4. Results of the single run submitted for COLIEE 2020 Task 4 on the test set.

COLIEE Task 4 data. The resulting performance was rather underwhelming. Since one of the causes was likely the small size of the data set we decided to fine-tune the model on the SNLI data set first. Only after that we proceeded to fine-tune the model on the COLIEE data. This lead to a major improvement in performance. For both of the fine-tuning stages we trained the model for 10 epochs using a batch size of 12. After the end of each epoch we recorded the accuracy of the model on the validation set. In both stages we picked the model with the best accuracy to work with. In case of the fine-tuning on SNLI the best model was then used for further fine-tuning on the training data of COLIEE Task 4. The best model resulting from the second stage of fine-tuning was then used to generate the predictions on the test set.

5.3 Results

We only submitted one run of the model described in the preceding section. While the accuracy of 0.6161 is quite underwhelming for a binary task the run ended up as the sixth best performing one (shared with other two runs) out of the total number of 30 submitted runs. This clearly shows that the task is extremely difficult and much more complicated than the general entailment as encoded in the SNLI data set. There the accuracy of various models based on BERT appears to be very close to .90 in most cases.

5.4 Discussion

The most important takeaway is that one has to be careful about interpreting the results of various methods evaluated on general language data sets. For example, the performance of models based on BERT on general language entailment tasks, such as SNLI [1] or MNLI [28], gives an impression that the models deal with it very effectively (accuracy above 0.90). However, those data sets mostly consist of short sentences, often with considerable word overlap. The COLIEE Task 4 data are obviously much more complex and the application of the same methods on those data does not appear to lead to similar levels of performance. In the previous run of COLIEE several teams based their solutions to Task 4 on BERT. [7, 10] Although, theirs were not the winning submission they performed fairly well in relative terms ([7] ended up third). In [7] the accuracy on the test set was 0.5918 and here we achieved the accuracy of 0.6161. These numbers clearly show that the evaluation of statements' entailment with respect to statutory provisions is much more challenging than general language entailment.

An alternative explanation could be that BERT models are not well-suited for tasks performed on domain-specific legal texts. However, this does not appear likely since there has been numerous examples of successful applications

of BERT on legal texts. In [4] BERT is evaluated on classification of claim acceptance given judges' arguments. A task of retrieving related case-law similar to a case decision a user provides is tackled in [23]. The authors demonstrate the effectiveness of using BERT for this task while focusing on mitigating the constraint on document length imposed by BERT. In [3] BERT is evaluated as one of the approaches to predict court decision outcome given the facts of a case. BERT has been successfully used for classification of legal areas of Supreme Court judgments. [9] The authors of [20] combine BERT with simple similarity measure to tackle the challenging task of case law entailment. BERT was also used in learning-to-rank settings for retrieval of legal news [24] and case-law sentences interpreting statutory concepts [25].

6 Conclusions

In this paper we presented four systems focused on statutory and case law retrieval and entailment. The models were submitted to the COLIEE 2020 competition. The model that we submitted for Task 1, focused on case law retrieval, was the best model submitted for the task. The key idea behind the system is to evaluate the similarity between cases on the paragraph level. We used similar model for Task 2 on case law entailment. For Task 3 focusing on statutory retrieval we created a system based on the BM25 measure. Finally, we tackled the task on statutory entailment (Task 4) via fine-tuning a sentence pair classification model based on BERT.

References

1. Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint* arXiv:1508.05326.
2. Cer D., Yang Y., Kong S., Hua N., Limtiaco N., St. John R., Constant N., Guajardo-Cespedes M., Yuan S., Tar C., Sung Y., Strope B. & Kurzweil R. (2018). Universal Sentence Encoder. *arXiv preprint* arXiv:1803.11175.
3. Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2019). Neural legal judgment prediction in english. *arXiv preprint* arXiv:1906.02059.
4. Condevaux, C., Harispe, S., Mussard, S., & Zambrano, G. (2019, December). Weakly Supervised One-Shot Classification Using Recurrent Neural Networks with Attention: Application to Claim Acceptance Detection. In JURIX (pp. 23-32).
5. Cortes C. & Vapnik V., (1995). Support-vector networks, *Mach Learn*, 20, 3, (pp. 273–297).
6. El Hamdani, R., Trousssel, A. & Houvenagel, C. (2019) COLIEE Case Law Competition Task 1: The Legal Case Retrieval Task.*COLIEE'2019*. (pp. 10-15).
7. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: Iitp@coliee 2019: Legal information retrieval using bm25 and bert. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
8. Harter, S. P. (1975). A probabilistic approach to automatic keyword indexing. Part I. On the distribution of specialty words in a technical literature. *Journal of the american society for information science*, 26(4), 197-206.

9. Howe, J. S. T., Khang, L. H., & Chai, I. E. (2019). Legal area classification: a comparative study of text classifiers on Singapore supreme court judgments. *arXiv preprint* arXiv:1904.06470.
10. Hudzina, J., Vacek, T., Madan, K., Tonya, C., Schilder, F. (2019). Statutory entailment using similarity features and decomposable attention models. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019.
11. Kitaev, N., Kaiser, Ł. & Levskaya, A. (2020). Reformer: The Efficient Transformer. *arXiv preprint* arXiv:2001.04451.
12. Kotsiantis S., Kanellopoulos D., & Pintelas P., (2016). Handling imbalanced datasets: A review, *GESTS International Transactions on Computer Science and Engineering*, 30, 1 (pp. 25–36).
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint* arXiv:1907.11692.
14. Manning, C. D., Schütze, H., & Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge university press.
15. Mitra, B., Nalisnick, E., Craswell, N., & Caruana, R. (2016). A dual embedding space model for document ranking. *arXiv preprint* arXiv:1602.01137.
16. Murdock, V. G. (2006). *Aspects of sentence retrieval*. Massachusetts University Amherst, Department of Computer Science.
17. Paulino-Passos, & G. Toni, F. (2019) Retrieving Legal Cases with Vector Representations of Text. In Proceedings of COLIEE'2019 (pp. 50-55).
18. Pedregosa F. et al., (2011) Scikit-learn: Machine learning in Python, *the Journal of machine Learning research*, 12, (pp. 2825–2830).
19. Platt J. C., (1999) Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, *Advances in Large Margin Classifiers* (pp. 61–74).
20. Rabelo, J., Kim, M. Y., & Goebel, R. (2019, June). Combining similarity and transformer methods for case law entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law* (pp. 290–296).
21. Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.
22. Robertson, S. E., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94* (pp. 232-241).
23. Rossi, J., & Kanoulas, E. (2019, December). Legal Search in Case Law and Statute Law. In JURIX (pp. 83-92).
24. Sanchez, L., He, J., Manotumruksa, J., Albakour, D., Martinez, M., & Lipani, A. (2020, April). Easing Legal News Monitoring with Learning to Rank and BERT. In *European Conference on Information Retrieval* (pp. 336-343). Springer, Cham.
25. Savelka, J. (2020). Discovering sentences for argumentation about the meaning of statutory terms (Doctoral dissertation, University of Pittsburgh).
26. Savelka, J., Walker, V. R., Grabmair, M., & Ashley, K. D. (2017). Sentence boundary detection in adjudicatory decisions in the united states. *Traitement automatique des langues*, 58, 21.
27. Shao, Y. et al. (2020). BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval. *Electronic proceedings of IJCAI 2020* vol. 4 3501–3507.
28. Williams, A., Nangia, N., & Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint* arXiv:1704.05426.
29. Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., & Zhou, X. (2019). Semantics-aware bert for language understanding. *arXiv preprint* arXiv:1909.02209.

HUKB and the COLIEE 2020 Information Retrieval Task

Masaharu Yoshioka[1,2,3][0000−0002−2096−1218] and Youta Suzuki[2]

[1] Faculty of Information Science and Technology, Hokkaido University, N14 W9, Kita-ku, Sapporo-shi, Hokkaido, Japan
yoshioka@ist.hokudai.ac.jp
[2] Graduate School of Information Science and Technology, Hokkaido University
suzuki@eis.hokudai.ac.jp
[3] Global Station of Big data and cybersecurity, Hokkaido University

Abstract. Our HUKB group is participating in the legal information extraction task in COLIEE 2020. Our system uses BERT-based information retrieval for handling the issues involving word mismatch between queries and articles. We also use ordinal keyword-based information retrieval, improving the retrieval results by aggregating the outputs from these two methods. We estimate the difficulty of the question by comparing both systems' outputs and using this information to decide the number of relevant articles selected by the system. In this paper, we introduce the system we use for the submission of final results and discuss the characteristics of the system.

Keywords: Legal Information Retrieval · Query Analysis · Deep Neural Network

1 Introduction

The competition on Legal Information Extraction/Entailment (COLIEE) [3, 2, 9, 4] serves as a forum to discuss issues related to legal information retrieval (IR) and entailment. The IR task for Japanese statute law (Task 3) is one of the oldest tasks in COLIEE. Retrieval-system performance has improved during the course of the campaign, but there remain types of questions that are easily retrieved by all participants and others that are not retrieved by any participants [8]. Because the IR task for COLIEE statute law is a necessary preliminary to the entailment task (Task 4), it is important to provide a method for identifying the relevant articles for such difficult questions.

In this paper, we propose an ensemble-based IR system that combines an ordinal keyword-based IR system with a deep neural network (DNN) system derived from a natural language processing (NLP) system. In this framework, final retrieval results are generated by aggregating the output of the two systems and the number of articles returned for each question is determined in terms of the retrieval difficulty estimated by comparing the results from both systems. Finally, we evaluate our system based on the submission results and discuss the characteristics of the system via failure analysis.

2 An ensemble-based IR system

In earlier years of the COLIEE, the best-performing systems used ordinal keyword-based IR as a core component of the system. However, in COLIEE 2019, a DNN-based IR system [5] had better retrieval results when comparing systems via the mean average precision (MAP) calculated for the top 100 retrieval results for each question. This suggests that a DNN-based IR system is successful when finding relevant articles for those questions whose term overlap between question and answer is small. However, there are types of questions that explain the situation of the case with keywords that are important to identify the relevant articles. For such a question, since keyword-based IR system only use keywords used in the articles, the keyword-based IR system puts emphasis on such keywords and succeed to find the relevant article. However, the DNN-based IR system may fail to select articles if they made mistake on analyzing context defined by keywords for explaining the situation.

Based on this understanding, we propose a method that combines an ordinal keyword-based IR system and a DNN-based IR system, aiming to achieve better overall results.

For the ordinal keyword-based IR system, we adopt Indri [7][4], which is a state-of-the-art keyword-based IR system based on language modeling. We have confirmed this system performed well as a baseline system, even for out-of-the-box settings [10].

One of the difficulties of utilizing a DNN-based IR system is the small amount of training data available. To address this problem, BERT [1] was proposed as a DNN-based NLP tool that provides pretraining for the general semantic understanding task. This then forms the basis for fine-tuning the model using task-related data. Because the BERT model we used for our system is trained on a large Japanese text corpus to construct a language model that handles a wide variety of terms, it can estimate the semantic similarity between words that do not appear in the articles and those that do. This is why we adopted this BERT-based IR system [6] as a candidate for our DNN-based IR system.

Because of the different characteristics of these two approaches, aggregating the results generated by these two systems should improve the retrieval performance compared with using results from one system only. In combining the two systems' outputs, we use both sets of outputs to estimate the retrieval difficulty. Where there is an article that has a large term overlap with a question, both systems are likely to rank the article as the first-ranked relevant article. However, where there is no article with a large term overlap, the top-ranked retrieval result may be different between the systems. We, therefore, estimate the retrieval difficulty of the question based on a comparison of the top-ranked retrieval results from the two systems. The final output is then calculated by aggregating the results from the BERT-based IR system and Indri. Details of the system are discussed in Section 2.1.

[4] `https://www.lemurproject.org/indri/`

2.1 Implementation

For the ordinal keyword-based IR system, we used Indri with its out-of-the-box settings. For each article, we extracted the contents of the article, the title, and the section headings for the database. For the DNN-based IR system, we used a BERT-based IR system [6] [5]. This system is a Japanese-pretrained BERT model, which uses the Japanese Wikipedia for pretraining[6].) In this system, the IR model is trained as a binary classification task that checks each pair of questions plus relevant documents as being relevant or not. Based on the trained model, the system can calculate a score for all documents for a given question. This score is used in ranking the documents. In our case, we selected the question plus relevant article pairs in the training data as positive examples, with negative examples also being constructed.[OLE6] We selected 24 negative examples for each question, which is the default setting for the BERT-based IR system. We constructed training data using all questions plus article pairs in the training data. After generating both sets of retrieval results, we aggregated the results using two different procedures, called HUKB-1 and HUKB-2. Each of these procedures is discussed in the following.

HUKB-1 This is a simple aggregation procedure that classifies the retrieval difficulty into two types (easy and difficult).

1. **Aggregation of the results**
   Because the scoring format used by the two systems is different (Indri generates negative scores and the BERT-based IR system generates scores from 0 to 1), it is inappropriate to simply aggregate these scores. Therefore, we used the following formula to calculate the score using the rank of each article for the question as generated by Indri ($IndriRank_{article}$) and BERT ($BERTRank_{article}$).

$$score_{article} = \frac{1}{(IndriRank_{article} + \alpha) * 2} + \frac{1}{(BERTRank_{article} + \alpha) * 2} \quad (1)$$

   Here, α is a parameter to reduce the score of the first-rank articles compared with the second-rank articles. In addition, because BERT's task is binary classification, a difference in score of less than β is considered meaningless. Therefore, we set rank 800 instead of the original rank whenever the score was below β. We used $\alpha = 1$ and $\beta = 0.3$ in this experiment.
2. **Estimation of the retrieval difficulty**
   If both systems return the same article as the first-ranked relevant article, we classify the question as "easy" and return the first-ranked article in the overall result. If the results are different, we classify the question as "difficult" and return the top three ranked documents in the overall result.

[5] https://github.com/ku-nlp/bert-based-faqir

[6] http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT 日本語 Pretrained モデル (http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT%E6%97%A5%E6%9C%AC%E8%AA%9EPretrained%E3%83%A2%E3%83%87%E3%83%AB)

HUKB-2 The second (HUKB-2) approach attempts to select the articles by comparing the ranks from BERT and Indri. In addition to the rank generated by the two systems, the system sorts the articles based on a simple aggregated score $AggregatedScore_{article}$ in ascending order. The rank for an article in this sorted list is called $AggregatedRank_{article}$.

$$AggregatedScore_{article} = IndriRank_{article} + BERTRank_{article} \quad (2)$$

The system compares the results for these rankings to check the reliability of the two retrieval systems (BERT or Indri) in selecting the candidate articles. We now give a detailed explanation of the selection method based on the classification of the question. In this method, whenever the system cannot select candidates at a given step, the system goes to the next step to select the candidates.

1. **Easy case**
   If the top-ranked results from BERT and Indri are similar, the top-ranked document ($AggregatedRank_{article} = 1$) is an article that is ranked first in BERT and/or Indri. In such a case, we classify the question as easy and select the top-ranked article as a candidate. In addition to the top-ranked article, the second-rank article ($AggregatedRank_{article} = 2$) is also selected if the rankings by BERT and Indri are both below five ($BERTRank_{article} \leq 5$ and $IndriRank_{article} \leq 5$).
2. **BERT reliable case**
   Because of the small number of negative examples, our BERT system is not very selective and there are many cases where substantial numbers of articles have BERT scores greater than 0.9. In such cases, BERT can find the relevant articles, but they are not accurately ranked. To identify such cases, the system checks the BERT score of the 10th ranked article ($BERTRank_{article} = 10$). If this score is larger than 0.9, we classify the case as a "BERT reliable" case. The system then selects the BERT top 10 ranked articles and reranks them in terms of $AggregatedRank_{article}$. Finally, the top five articles are selected from the list.
3. **Ambiguous case**
   If the BERT score of the 10th-ranked article is below 0.9, this may indicate that BERT has misunderstood the context of the question. However, if the ranks generated by BERT and Indri are similar, the aggregated results generated by both systems may be reliable. Therefore, the system compares the top five ranked lists of BERT and Indri and counts the number of articles that belong to both lists. If this number of articles is at least two, the system classifies the question as an "ambiguous" case and returns the articles using the following substeps.
   (a) **Indri reliable case**
   If both lists include the top-ranked Indri document ($IndriRank_{article} = 1$), the system classifies the case as an "Indri reliable" case and uses Indri's top five ranked articles as candidates.

(b) **BERT reliable case**
Otherwise, the system uses BERT's top five ranked articles as candidates.

4. **Difficult case**
In other cases, where there is no clue to judge the reliability of the two sets of results, the system compares the top 15 ranked lists from BERT and Indri and selects those articles that belong to both lists as candidates.

2.2 Submission Results

We submitted results for three different versions of our system for Task 3. HUKB-1 and HUKB-2 is based on the ensemble approach discussed in Section 2.1. However, since the HUKB-2 had bugs on the program for making a submission results, we include the evaluation results of submitted result and debugged one. The HUKB-3 results were generated by the BERT-based IR alone. Because the macro F2 measure for HUKB-1 was almost the same as that for HUKB-2, but HUKB-1 performed better than HUKB-2, we discuss the characteristics of the system in terms of the HUKB-1 outputs.

Table 1 gives the evaluation results of the submitted runs. Precision, recall, and F2 are calculated as macro averages, with MAP being calculated via long lists. The Indri run selects the top-ranked article as a result for each question. Because HUKB-1 returns multiple answers, its recall is marginally better than those for the original two systems (HUKB-3 and Indri), but its precision is also not too bad in comparison.

Table 1. Evaluation results of submitted runs (Task 3)

Run	Return	Retrieved	Precision	Recall	F2	MAP
HUKB-1	250	75	0.4196	**0.5908**	**0.5160**	0.5687
HUKB-2 (submited)	248	**76**	0.4036	**0.5908**	0.5138	0.5744
HUKB-2 (debugged)	230	75	0.3924	0.5818	0.5119	**0.5815**
HUKB-3	112	49	0.4375	0.4107	0.4137	0.5435
Indri	112	50	**0.4464**	0.3899	0.3958	0.5253

Table 2 shows a contingency table of questions that are estimated as difficult and the results from HUKB-1 (whether the question was found by the system or not). Numbers in parentheses represent the ratio of found/not found articles for each category (Easy or Difficult).

From this table, we can note that difficult questions are comparatively difficult to retrieve even if we include three answers for each question. Using only the first top-ranked article, we can find answers for 24 questions only. A successful aspect of these merged results is that relevant articles were not missed, unlike those missed by using only one system (16 articles for HUKB-3 and 17 articles

for Indri). An additional side effect was the inclusion of additional secondarily relevant articles in the top three articles for two of the questions. Overall, HUKB-1 found at least one relevant article for 73 questions.

However, for the easy questions, even though we included the top three ranked articles for each question, we could only find relevant articles for four questions. This does improve the recall, but it adversely affects the precision for other easy questions.

Table 2. Evaluation results for HUKB-1 in terms of retrieval difficulty

	Found	Not found	All
Difficult	40 (0.58)	29 (0.42)	69
Easy	33 (0.77)	10 (0.23)	43

For the easy questions, we found most of the relevant articles in the top five (second: 3, third: 1, fourth: 3, fifth: 1), which represented at least one article for each query. There were only two questions for which relevant articles for the questions were not found.

For the difficult questions, the inclusion of the fourth and fifth rank of the merged results offers little improvement (fourth: 2, fifth: 6). There remained 21 questions for which relevant articles for the questions were not found.

2.3 Failure Analysis

There are two questions (R01-5-O, R1-14-U) that were estimated as easy questions, but for which the merged results did not find relevant articles in the top five. Figure 1 shows an example of the questions (R01-14-U) with the relevant article and top-ranked article from both systems. This is a typical example of a question for which our system fails to retrieve relevant articles. To understand the relationships between the question and the relevant answers, the most important issue discussed in the question is the relationship between the building (first thing) and the land (second thing). Such an interpretation is difficult to achieve using our system.

Another example is where the BERT-based IR finds the relevant article but Indri fails (see Figure 2). This is a typical example of partial word overlap. The answer covers keywords such as “賃貸借”(lease) and “効力を生ず” (becomes effective). However, the top-ranked article retrieved by Indri covers keywords such as “書面でしなければ，”(in writing) and “その効力を生じない。”(becomes effective). In this case, BERT can apply the appropriate context to finding the relevant article. However, there are questions for which BERT made an incorrect estimation of the context and Indri succeeded in retrieving relevant articles. Therefore, because our merged approach considers these two different context estimations by merging the results, its final performance in terms of recall is much better than the performance of one system (BERT or Indri) acting alone.

Question: R01-14-U

借地上の建物について抵当権が設定された場合，抵当権の効力は，敷地の賃借権に及ぶことはない。
In cases where a mortgage is created with respect to a building on leased land, the mortgage may not be exercised against the right of the lease.

Answer

（主物及び従物）
第八十七条　物の所有者が、その物の常用に供するため、自己の所有に属する他の物をこれに附属させたときは、その附属させた物を従物とする。
２　従物は、主物の処分に従う。
(Principal Things and Appurtenances)
Article 87
(1) If the owner of a first thing attaches a second thing that the owner owns to the first thing to serve the ordinary use of the first thing, the thing that the owner attaches is an appurtenance.
(2) An appurtenance is disposed of together with the principal thing if the principal thing is disposed of.

Top-ranked relevant article from both systems.

（抵当権の効力の及ぶ範囲）第三百七十条　抵当権は、抵当地の上に存する建物を除き、その目的である不動産（以下「抵当不動産」という。）に付加して一体となっている物に及ぶ。ただし、設定行為に別段の定めがある場合及び債務者の行為について第四百二十四条第三項に規定する詐害行為取消請求をすることができる場合は、この限りでない。
(Scope of Effect of Mortgages)
Article 370 A mortgage extends to the things that form an integral part of the immovables that are the subject matter of the mortgage (hereinafter referred to as “mortgaged immovables”) except for buildings on the mortgaged land; provided, however, that this does not apply if the act establishing the mortgage provides otherwise or the rescission of a fraudulent act may be demanded as prescribed in Article 424, paragraph (3) with regard to the act of the obligor.
Article 371 If there is a default with respect to a claim secured by a mortgage, the mortgage extends to the fruits of the mortgaged immovables derived after the default.

Fig. 1. Example of the failure of retrieval difficulty estimation

Question: R01-25-A

賃貸借は，書面でしなければ，その効力を生じない。
A lease becomes effective only when in writing.

Answer

（賃貸借）
第六百一条　賃貸借は、当事者の一方がある物の使用及び収益を相手方にさせることを約し、相手方がこれに対してその賃料を支払うこと及び引渡しを受けた物を契約が終了したときに返還することを約することによって、その効力を生ずる。
(Leases)
Article 601 A lease becomes effective if one of the parties promises to make a certain thing available for the other party to use and make a profit, and the other party promises to pay rent for the leased thing and return the delivered thing when the contract is terminated.

Top-ranked relevant article by Indri

（保証人の責任等）
第四百四十六条　保証人は、主たる債務者がその債務を履行しないときに、その履行をする責任を負う。
２　保証契約は、書面でしなければ、その効力を生じない。
３　保証契約がその内容を記録した電磁的記録によってされたときは、その保証契約は、書面によってされたものとみなして、前項の規定を適用する。
(Responsibility of Guarantor)
Article 446 (1) A guarantor has the responsibility to perform the obligation of the principal obligor when the latter fails to perform that obligation.
(2) No guarantee contract becomes effective unless it is made in writing.
(3) If a guarantee contract is concluded by an electronic or magnetic record that records the terms thereof, the guarantee contract is deemed to be made in writing, and the provisions of the preceding paragraph apply.

Fig. 2. Example of the failure of Indri

2.4 Discussion

Because of the different characteristics of Indri and BERT-based systems, the ensemble approach works well in improving the recall of results. It also works well in identifying the retrieval difficulty via a comparison between both individual system outputs. However, there remains a need to improve the basic IR system to identify more relevant articles in a top-ranked retrieval list. Currently, there are 15 questions for which none of the systems can identify the relevant articles among the top 10. This issue should be a focus for future development. In addition, it will also be necessary to consider cases with multiple answers in an appropriate manner.

3 Summary

In this paper, we introduced our system for participating in Task 3 of COLIEE 2020. This system uses a BERT-based IR system that considers the meaning of words that do not appear in the articles when selecting appropriate articles. In addition, recognizing the different characteristics of BERT-based IR and ordinal keyword-based IR, we can confirm that an ensemble approach that considers retrieval difficulty works effectively compared with systems that use BERT-based IR or ordinal keyword-based IR only.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number 18H0333808.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
2. Kano, Y., Kim, M.Y., Goebel, R., Satoh, K.: Overview of coliee 2017. In: Satoh, K., Kim, M.Y., Kano, Y., Goebel, R., Oliveira, T. (eds.) COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment. EPiC Series in Computing, vol. 47, pp. 1–8. EasyChair (2017)
3. Kim, M.Y., Goebel, R., Kano, Y., Satoh, K.: Coliee-2016: Evaluation of the competition on legal information extraction and entailment. In: The Proceedings of the 10th International Workshop on Juris-Informatics (JURISIN2016) (2016), paper 11
4. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: A summary of the COLIEE 2019 competition. In: The Proceedings of the 13th International Workshop on Juris-Informatics (JURISIN2019). pp. 69–82. The Japanese Society of Artificial Intelligence, (2019)

5. Raiyani, K., Quaresma, P.: Keyword machine learning based japanese statute lawretrieval and entailment task at COLIEE-2019. In: Proceedings of the 6th Competitionon Legal Information Extraction/Entailment. COLIEE ' 2019 (2019)
6. Sakata, W., Shibata, T., Tanaka, R., Kurohashi, S.: Faq retrieval using query-question similarity and bert-based query-answer relevance. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1113–1116. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331326, https://doi.org/10.1145/3331184.3331326
7. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. pp. 2–6 (2005)
8. Yoshioka, M.: Analysis of coliee information retrieval task data. In: Arai, S., Kojima, K., Mineshima, K., Bekki, D., Satoh, K., Ohta, Y. (eds.) New Frontiers in Artificial Intelligence. pp. 5–19. Springer International Publishing, Cham (2018)
9. Yoshioka, M., Kano, Y., Kiyota, N., Satoh, K.: Overview of japanese statute law retrieval and entailment task at coliee-2018. In: The Proceedings of the 12th International Workshop on Juris-Informatics (JURISIN2018). pp. 117–128. The Japanese Society of Artificial Intelligence, (2018)
10. Yoshioka, M., Song, Z.: Hukb at coliee2018 information retrieval task. In: The Proceedings of the 12th International Workshop on Juris-Informatics (JURISIN2018). pp. 183–193. The Japanese Society of Artificial Intelligence, (2018)

Author Index