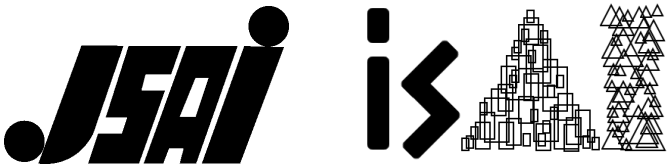
Proceedings of the Sixteenth International Workshop on Juris-informatics (JURISIN 2022)

hosted by JSAI-isAI2022

Workshop Co-Chairs

Makoto Nakamura
Satoshi Tojo



Kyoto International Conference Center, Kyoto, Japan and ONLINE,
June 13-14, 2022

Preface

The Sixteenth International Workshop on Juris-Informatics (JURISIN 2022) is held with a support of the Japanese Society for Artificial Intelligence (JSAI) in association with JSAI International Symposia on AI (JSAI-isAI 2022). JSAI-isAI is a collection of several workshops and two workshops were selected this year. Although JURISIN was organized to discuss legal issues from the perspective of informatics, the scope of JURISIN is more wide-ranging than that of the conventional AI and law. Thus, the members of Program Committee (PC) are leading researchers in various fields. The collaborative work of computer scientists, lawyers and philosophers is expected to contribute to the advancement of juris-informatics and it is also expected to open novel research areas.

Despite a short announcement period, 25 papers were submitted. Each paper was reviewed by three or more members of PC. This year, one paper was withdrawn and 22 papers were accepted in total. The collection of papers covers various topics such as legal reasoning, argumentation theory, legal compliance, dispute resolution, application of AI and informatics to law, application of natural language processing and so on. As a general tendency of these years, we notice that the interests in language-specific legal documents, security, semantic representation, as well as machine learning techniques become higher.

Following the previous years, JURISIN have a session on the Competition on Legal Information Extraction/Entailment (COLIEE 2022) which consists of a result report of the competition and 12 papers for each participant. Especially this year, the interest in COLIEE is strong. Among 22 accepted papers, 12 are the participants to COLIEE.

As invited speakers, we have Enrico Francesconi, Researcher of IGSG-CNR, Italy and Randy Goebel, Professor of University of Alberta, Canada.

Finally, I would like to thank all the members of PC and subreviewers for reviewing the submitted papers.

June, 2022 Makoto Nakamura
Kyoto Satoshi Tojo

Program Committee

Michał Araszkiewicz	Jagiellonian University
Ryuta Arisaka	Kyoto University
Marina De Vos	University of Bath
Kripabandhu Ghosh	Indian Institute of Science Education and Research (IISER) Kolkata, India
Saptarshi Ghosh	Indian Institute of Technology Kharagpur
Randy Goebel	University of Alberta
Guido Governatori	Independent researcher
Tokuyasu Kakuta	Chuo University
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Makoto Nakamura	Niigata Institute of Technology
María Navas-Loro	Universidad Politécnica de Madrid
Tomoumi Nishimura	Osaka University
Katsumi Nitta	National Institute of Advanced Indutrial Science and Technology
Yasuhiro Ogawa	Nagoya University
Ginevra Peruginelli	ITTIG-CNR
Juliano Rabelo	University of Alberta
Julien Rossi	Amsterdam Business School
Seiichiro Sakurai	Meiji Gakuin University
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Jaromir Savelka	Carnegie Mellon University
Yunqiu Shao	Tsinghua University
Akira Shimazu	JAIST
Yoichi Takenaka	Kansai University
Satoshi Tojo	Japan Advanced Institute of Science and Technology
Katsuhiko Toyama	Nagoya University
Vu Tran	The Institute of Statistical Mathematics, Japan
Josef Valvoda	University of Cambridge
Sabine Wehnert	Otto-von-Guericke-Universität Magdeburg
Yueh-Hsuan Weng	Tohoku University
Hannes Westermann	University of Montreal
Hiroaki Yamada	Tokyo Institute of Technology
Takahiro Yamakoshi	Nagoya University
Masaharu Yoshioka	Hokkaido University
Thomas Ågotnes	University of Bergen

Additional Reviewers

Nguyen, Ha-Thanh

Table of Contents

Profiles of Legal Knowledge Representation and Reasoning in the Semantic Web

Enrico Francesconi[0000−0001−8397−5820]

Institute of Legal Informatics and Judicial Studies
National Research Council of Italy (IGSG-CNR)
enrico.francesconi@igsg.cnr.it
http://www.igsg.cnr.it

Abstract. Machine readable, actionable rules represent a precondition for developing systems endowed with automatic reasoning facilities for advanced information services in the legal domain. In this talk we present an approach for legal knowledge representation and reasoning within a Semantic Web framework. It is based on the distinction between provisions and norms and it is able to provide reasoning facilities for advanced legal information retrieval (like implementing Hohfeldian reasoning) and legal compliance checking for deontic notions. It is also shown how the approach can handle norm defeasibility. Such methodology is implemented by decidable fragments of OWL 2, while legal reasoning is implemented by available decidable reasoners.

Keywords: Semantic Web · Decidable Legal Reasoning · Legal Information Retrieval · Legal Compliance

What is required to capture legal knowledge for automated legal reasoning?

Randy Goebel

Alberta Machine Intelligence Institute,Computing Science Department,
University of Alberta, Canada
rgoebel@ualberta.ca

Over the life of the Competition on Legal Information Extraction and Entailment (COLIEE), we have seen almost a decade of change in the development and application of natural language processing (NLP) and machine learning (ML), with a broad variety of methods to capture and represent the subject matter of legal information. We present a personal perspective on these changes, including approaches to the legal knowledge acquisition problem, and provide some speculative hypotheses on what AI-based legal reasoning systems must do to provide practical value.

COLIEE 2022 Summary: Methods for Legal Document Retrieval and Entailment

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

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

Abstract. We present a summary of the 9th 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 who could use 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 (26 runs) and 5 teams for Task 2 (14 runs). On the statute law task, there were 11 different teams participating, most in more than one task. Five teams submitted a total of 15 runs for Task 3, and 6 teams submitted a total of 18 runs for Task 4. We summarize their approaches, our official evaluation, and provide 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. The intention is 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, 2019, and 2021. Until 2018, 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, which are hypothesized to support the decision for Q. Task 2 is a 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 further understand details of the difficulty of the questions. For the entailment task (Task 4), we performed categorized analyses to expose a variety of issues with the problems and characteristics of the submissions, in addition to the evaluation accuracy as in previous COLIEE tasks.

The rest of our 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 provided 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 dataset is comprised of a total of 5,978 case law files. Provided is a labelled training set of 4,415, of which 900 query cases. On average, there are approximately 4.9 noticed cases per query case in the provided training dataset, which

should be identified among the 4,415 cases. To prevent competitors to merely use citations of past cases in order to identify cited cases, citations are suppressed from the case contents and replaced by a "FRAGMENT_SUPPRESSED" tag indicating that fragment was removed from the case contents.

A test set is given, consisting of a total of 1,563 cases, with 300 query cases and a total of 1,263 true noticed cases, which means there are on average 4.21 noticed cases per query case in the test dataset. Initially, the golden labels for that test set is not provided to competitors.

2.3 Approaches

We received 26 submissions from 9 different teams for Task 1. In this section, we present an overview of the approaches taken by the 7 teams which submitted papers describing their methods. Please refer to the corresponding papers for further details.

- **TUWBR (2 runs)** [5] start from two assumptions: first, that there is a topical overlap between query and notice cases, but that not all parts of a query case are equally important. Secondly, they assume that traditional IR methods such as BM25 provide competitive results in Task 1. They perform document level and passage level retrieval, and also augment the system by adding external domain knowledge by extracting statute fragments mentioned in the cases and explicitly adding those fragments to the documents.
- **JNLP (3 runs)** [3] applies an approach that consists first splits the documents into paragraphs, then calculates the similarities between cases by combining term-level matching and semantic relationships on the paragraph level. They also propose an attention model to encode the whole query in the context of candidate paragraphs, then infer the relationship between cases.
- **DoSSIER (3 runs)** [1] combined traditional and neural based techniques in Task 1. The authors investigate lexical and dense first stage retrieval methods aiming for a high recall in the initial retrieval and then compare shallow neural re-ranking with the MTFT-BERT model to the BERT-PLI model. They then investigate which part of the text of a legal case should be taken into account for re-ranking. The achieved results show that BM25 shows a consistently high effectiveness across different test collections in comparison to the neural network re-ranking models.
- **LeiBi (3 runs)** [2] applied an approach which consists of the following steps: first, given a legal case as a query, they reformulate it by heuristically extracting various meaningful sentences or n-grams. The authors then use the pre-processed query case to retrieve an initial set of possible relevant legal cases, which are further re-ranked. Finally, the team aggregates the relevance scores obtained by the first stage and the re-ranking models to improve retrieval effectiveness. The query cases are reformulated to be shorter, using three different statistical methods (KLI, PLM, IDF-r), in addition to models that leverage embeddings (e.g., KeyBERT). Moreover, the authors investigate if automatic summarization using Longformer-Encoder-Decoder

(LED) can produce an effective query representation for this retrieval task. Furthermore, the team proposes a re-ranking cluster-driven approach, which leverages Sentence-BERT models to generate embeddings for sentences from the query and candidate documents. Finally, the authors employ a linear aggregation method to combine the relevance scores obtained by traditional IR models and neural-based models.

- **UA (3 runs)** [9] use a transformer-based model to generate paragraph embeddings, and then calculate the similarity between paragraphs of query and positive and negative cases. The calculated similarities are used to generate feature vectors (10-bin histograms of all pair-wise between 2 cases), and then using a Gradient Boosting classifier to determine if those cases should be noticed or not. The UA team also applies pre- and post-processing heuristics to generate the final results, which were ranked first in Task 1 of the current COLIEE edition;
- **nigam (3 runs)** [8] developed an approach which was a combination of transformer-based and traditional IR techniques; more specifically, they used Sentence-BERT and Sent2Vec for semantic understanding combined with BM25. First, the nigam team selects top-K candidates according to the BM25 rankings, and then they use pre-trained Sentence-BERT and Sent2Vec to generate representation features of each sentence. The authors also used cosine similarity with the max-pooling strategy to get the final document score between the query and noticed cases.
- **siat (3 runs)** [15] show how longformer-based contrastive learning is able to process sequences of thousands of tokens, thus overcoming a well-known limitation of common transformer-based methods which are usually restricted to between 512 and 1024 tokens. In addition to that longformer-based approach, the siat team also explores traditional retrieval models. They achieved second place overall in Task 1 of COLIEE 2022.

2.4 Results

Table 1 shows the results of all submissions received for Task 1 for COLIEE 2022. A total of 26 submissions from 9 different teams have been received. Similar to what happened in COLIEE 2021 [11], the f1-scores are generally low, which reflects the fact that the task is now more challenging than its previous formulation (for a description of the previous Task 1 formulation, please see the COLIEE 2020 summary [10]). However, in the current edition, we already could see a relevant improvement in those scores, with the top teams achieving scores above 0.35 (up from 0.19 as the best score in the COLIEE 2021 edition).

Most of the participating teams applied traditional IR techniques such as BM25, transformer based methods such as BERT, or a combination of both. The best performing team was UA, with an f1-score of 0.3715, with an approach that relied on creating an embedding representation for the cases, and then calculating the similarity between each query case and positives and negatives samples from the training dataset. The resulting distances are then bucketed into 10-bin histograms, which are used to train a Gradient Boosting classifier. Also

worth mentioning is the siat team, whose approach made use of a longformer-based model and achieved second place overall.

Table 1. Task 1 results

Team	File	F1 Score	Precision	Recall
UA	pp_0.65_10_3.csv	0.3715	0.4111	0.3389
UA	pp_0.7_9_2.csv	0.3710	0.4967	0.2961
siat	siatrun1.txt	0.3691	0.3005	0.4782
siat	siatrun3.txt	0.3680	0.3026	0.4695
UA	pp_0.65_6.csv	0.3559	0.3630	0.3492
siat	siatrun2.txt	0.2964	0.2522	0.3595
LeiBi	run_bm25.txt	0.2923	0.3000	0.2850
LeiBi	run_weighting.txt	0.2917	0.2687	0.3191
JNLP	run3.txt	0.2813	0.3211	0.2502
nigam	bm25P3M3.txt	0.2809	0.2587	0.3072
JNLP	run2.txt	0.2781	0.3144	0.2494
DSSR	DSSR_01.txt	0.2657	0.2447	0.2906
JNLP	run1.txt	0.2639	0.2446	0.2866
DSSR	DSSR_03.txt	0.2461	0.2267	0.2692
TUWBR	TUWBR_LM_law	0.2367	0.1895	0.3151
LeiBi	run_clustering.txt	0.2306	0.2367	0.2249
TUWBR	TUWBR_LM	0.2206	0.1683	0.3199
nigam	sbertP3M3RS.txt	0.1542	0.1420	0.1686
nigam	s2vecP3M3RS.txt	0.1484	0.1367	0.1623
DSSR	DSSR_02.txt	0.1317	0.1213	0.1441
LLNTU	2022.task1.LLNTUtfidCos	0.0000	0.0000	0.0000
LLNTU	2022.task1.LLNTUtanadaT	0.0000	0.0000	0.0000
LLNTU	2022.task1.LLNTU3q4clii	0.0000	0.0000	0.0000
Uottawa	Task1Run3_UottawaLegalBert.txt	0.0000	0.0000	0.0000
Uottawa	Task1Run1_UottawaMB25.txt	0.0000	0.0000	0.0000
Uottawa	Task1Run2_UottawaSentTrans.txt	0.0000	0.0000	0.0000

For future editions of COLIEE, we intend to make the distributions of the training and test datasets more similar with respect to average and standard deviation of number of noticed cases. There are still some issues we need to fix in the dataset, such as two different files with the exact same contents (i.e., the same case represented as two separate files). This is a problem with the original dataset from where the competition's data is drawn, and knowing that dataset presents those issues we will improve our collection methods to correct them. Fortunately, those issues were rare and did not have a relevant impact on the final results.

3 Task 2 - Case Law Entailment

3.1 Task Definition

Given a base case and a specific fragment from it, together with 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

In Task 2, 525 query cases were provided for training against 18,740 paragraphs. There were 100 query cases against 3278 candidate paragraphs as part of the testing dataset. On average, there are 35.627 candidate paragraphs for each query case in the training dataset and 32.455 candidate paragraphs for each query case in the testing dataset. The average number of relevant paragraphs for Task 2 was 1.14 paragraphs for training and 1.18 paragraphs for testing.

3.3 Approaches

Five teams submitted a total of 14 runs to this task. They used LegalBERT, BM25, zero shot models, and some other heuristic approaches. More details on the approaches are shown below. Here, we introduce three teams' approaches that described their methods in their papers.

- **JNLP (3 runs)** [3] applied LegalBERT and BM25. In their first run, they combined the two scores from LegalBERT and BM25, and then ranked the outputs. In the second run, they used a knowledge representation technique called "Abstract Meaning Representation" (AMR) to capture the most important words in the query and corresponding candidate paragraph. In the third run, instead of combining the two scores from LegalBERT and BM25, they identified relevant paragraphs through the interaction between the top N candidates in LegalBERT and top M candidates in AMR + BM25.

- **nigam (2 runs)** [8] submitted two official runs both of which are based on the BM25 model. Run-1 uses only entailed fragment text as a query, and run-2 uses the filtered base case such that they combine search entailed fragment text into the base case, then provided as input to the model as searched results (along with previous and following sentences to capture the other relevant information). Every query case predicts one case law paragraph because the average number of relevant paragraphs is approximately 1.14 in the training dataset.

– **NM (3 runs)** [13] used monoT5, which is an adaptation of the T5 model. During inference, monoT5 generates a score that measures the relevance of a document to a query by applying a softmax function to the logits of the tokens "true" and "false." NM also extends a zero-shot approach, and they fine-tune the T5-base and T5-3B models for 10k steps, which corresponds to almost one epoch or approximately 530,000 query-passage pairs from the MS MARCO training set. They refer to the resulting models as monoT5-base-zero-shot and monoT5-3b-zero-shot. In the third run, they combine the answers from the two models. They apply their own answer selection method to select the final set of answers from the two models. Their ensemble model was ranked first in the COLIEE 2022 Task 2 competition.

3.4 Results

The F1-measure is used to assess performance in this task. The actual results of the submitted runs by all participants are shown on table 2, from which it can be seen that the NM team attained the best results. Among the three submissions from NM, two submissions were ranked first and second.

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

Team	Submission File	F1-score	Precision	Recall
NM	**monot5-ensemble.txt**	**0.6783**	0.6964	0.6610
NM	monot5-3b.txt	0.6757	0.7212	0.6356
JNLP	run2_bert_amr_remove_reduntdant_filter.txt	0.6694	0.6532	0.6864
JNLP	run3_bert_BM25.txt	0.6612	0.6452	0.6780
jljy	run2_task2.txt	0.6514	0.7100	0.6017
jljy	run3_task2.txt	0.6514	0.7100	0.6017
JNLP	run1_bert_amr_remove_reduntdant.txt	0.6452	0.6154	0.6780
jljy	run1_task2.txt	0.6330	0.6900	0.5847
NM	monot5-base.txt	0.6325	0.6379	0.6271
UA	res_score-0.95_max-1.txt	0.5446	0.6105	0.4915
UA	res_score-0.5_max-1.txt	0.5363	0.7869	0.4068
UA	res_score-0.95_max-5.txt	0.4121	0.3049	0.6356
nigam	bm25EF.txt	0.3204	0.1980	0.8390
UA	bm25BC.txt	0.2104	0.1300	0.5508

4 Task 3 - Statute Law Retrieval

4.1 Task Definition

Task 3 is a pre-processing step for legal textual entailment (Task 4), whose goal is to extract a subset of Japanese Civil Code Articles S_1, S_2,...,S_n from the entire Civil Code articles considered appropriate for answering the legal bar exam question Q such that
$Entails(S_1, S_2, ..., S_n, Q)$ or $Entails(S_1, S_2, ..., S_n, notQ)$.

Given a question Q and the all Civil Code Articles, the participants are required to retrieve the set of "$S_1, S_2, ..., S_n$" as the answer of this task.

4.2 Dataset

For Task 3, questions related to Japanese civil code articles were selected from the Japanese bar exam. However, since (updated in 2020), we use civil law articles that have official English translation (768 articles in total) as the target civil code.

The number of questions classified by the number of relevant articles is listed in Table 3.

Table 3. Number of questions classified by number of relevant articles

number of relevant article(s)	1	2	3	4	5	total
number of questions	94	11	2	1	1	109

4.3 Approaches

The following 5 teams submitted their results (15 runs in total). All teams had experience in submitting results in the previous competition and extend their previous approaches. Ordinal IR models such as BM25 [12], TF-IDF are still good models with better performance. Deep learning (DL) based approaches (using BERT [4] and other variants) are also effective for the task. There are several runs that combines outputs of these different approaches.

- **HUKB (3 runs)** [16] uses BM25 IR model with different document article databases (original article, rewritten articles using reference, and the judicial decision part of the articles). Final results are generated by using these results.
- **JNLP (3 runs)** [3] uses a deep learning (DL) based approach with identification of the use-case questions. Due to the different characteristics of the data for use-case question and others, they propose to make two models: one is for ordinal questions and the other is for use-cases.
- **LLNTU (3 runs)** previously used a BERT-based method, but there paper has no clear explanation about any adjustments for this year's task.
- **OvGU (3 runs)** [14] uses the scores of a TF-IDF model combined with a sentence-embedding based similarity score to produce answer rankings. They also use external knowledge (texts related to the articles) to calculate sentence-embedding.
- **UA (3 runs)** uses the TF-IDF model and BM25 model as an IR module.

4.4 Results

Table 4 shows the evaluation results of submitted runs. The official evaluation measures used in this task were macro average (average of evaluation measure values for each query over all queries) of F2 measure[7], precision, and recall.

$$precision = \frac{number\ of\ retrieved\ relevant\ articles}{number\ of\ returned\ articles} \tag{1}$$

$$recall = \frac{number\ of\ retrieved\ relevant\ articles}{number\ of\ relevant\ articles} \tag{2}$$

$$f2 = \frac{5 \times precision \times recall}{4 \times precision + recall} \tag{3}$$

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).

Table 4 shows the results of the evaluation of submitted results. Due to the limitation of the size of paper, the best performance run in terms of F2 are selected from each team runs.

Table 4. Evaluation results of submitted runs (Task 3) and the corresponding organizers' run

sid	return	retrieved	F2	Precision	Recall	MAP
HUKB2	136	101	0.820	0.818	0.841	0.843
OVGU_run3	161	96	0.779	0.778	0.805	0.836
JNLP.longformer	178	101	0.770	0.687	0.838	0.793
UA_TFIDF2	115	90	0.764	0.807	0.764	0.829
LLNTU0066cc	114	74	0.642	0.674	0.639	0.700

Figure 1,2,3 shows average of evaluation measure for all submission runs. The number of questions whose relevant article is 1 and F2 (average of all submission) is higher than 0.8 is 65.9% (62/94). This is better than that for the COLIEE 2021 29.2% (19/65). This may reflect the different characteristics of the dataset and participation of the well-experienced team, but this result shows that we have almost succeeded to develop a method for identifying easy questions. From Fig. 1, we confirmed that there are many easy questions for which almost all system can retrieve the relevant article. The easiest question is R03-2-A "The obligee may not exercise the right of the obligor, if the right is immune from attachment." whose relevant article contains sentence with same vocabularies. However, there are also many queries for which none of the system can retrieve the relevant articles. R03-07-E is an example of this question: "After A sold land

[7] Since task 3 is a preprocess for legal textual entailment (task 4), it is important to have all relevant articles in the retrieved results. So we put emphasis on recall in this evaluation

X owned by A to B, B resold land X to C, and each of them was registered as such. Later, the sales contract between A and B was voided on the grounds that A was an adult ward. If C did not know without negligence that A was an adult ward, A may not claim that the ownership of land X belongs to A," and relevant article is Article 121 "An act that has been rescinded is deemed void ab initio." This article uses specific legal terminology such as "rescinded" and "void ab initio." This is almost impossible to handle with an ordinal keyword-based IR system. It is also difficult for the DL-based approach, because it is not a simple semantic association matching.

For the question with multiple relevant articles, it is difficult to determine the significant difference from the viewpoint of overall evaluation. We still have problems to determine the whole set of relevant articles compared with single relevant article cases.

A new approach proposed in this year's competition is as follows:

- Using different document database (e.g., article text database and texts for judicial decision part database) for calculating the score to merge [16].
- Using external knowledge (text related to the articles) to enrich the sentence-embedding information of the article. [14]
- Classification of the query types; use-case queries or others. [3]

Those approaches are confirmed to be effective in this year's test data, and we expect that the combination of those approaches may improve the retrieval performance compared with this year's system.

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, 2020 and 2022, we returned to the pure textual entailment task to attract more participants, allowing more focused analyses. Participants can use any external data, however assuming that they do not use the test dataset and/or something which could directly contains the correct answers of the test dataset, because this task is intended to be a pure textual entailment task. Towards deeper analysis, we asked the participants to submit their outputs when using any fragment of the training dataset (H30-R02), in addition to the formal runs.

5.2 Dataset

Our training dataset and test dataset are the same as for Task 3. Questions related to Japanese civil law were selected from the Japanese bar exam. The

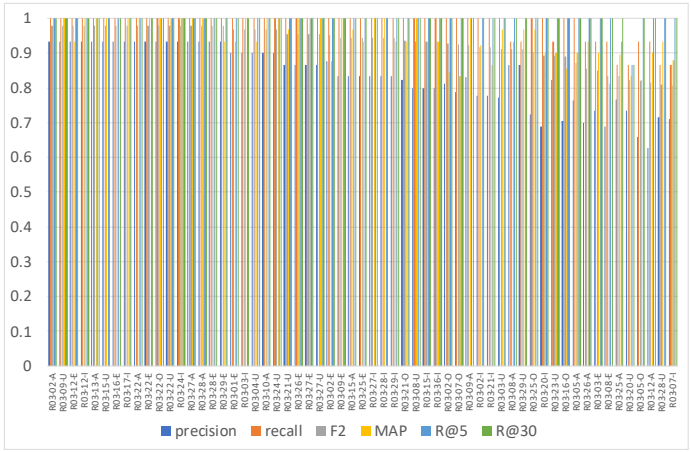


Fig. 1. Averages of precision, recall, F2, MAP, R_5, and R_30 for easy questions with a single relevant article

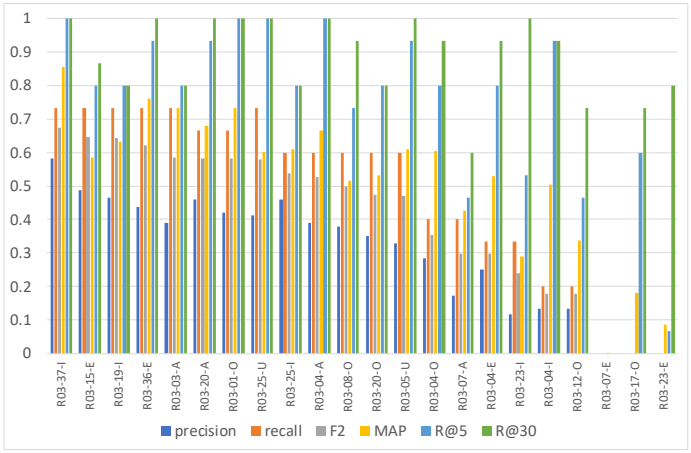


Fig. 2. Averages of precision, recall, F2, MAP, R_5, and R_30 for noneasy questions with a single relevant article

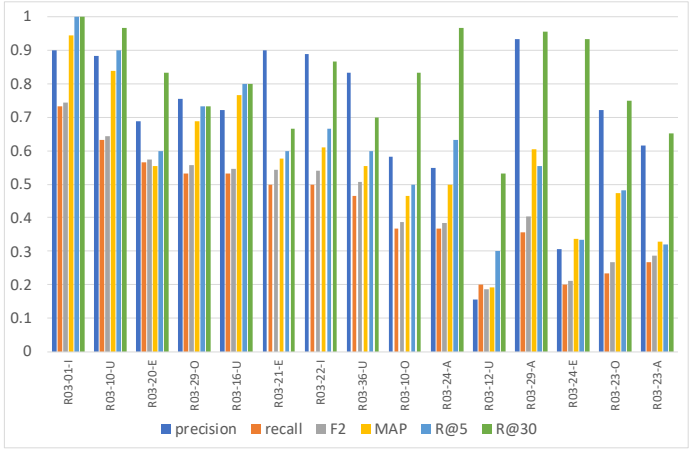


Fig. 3. Averages of precision, recall, F2, MAP, R_5, R_10, and R_30 for noneasy questions with a single relevant article

organizers provided a data set used for previous campaigns as training data (887 questions) and new questions selected from the 2022 bar exam as test data (109 questions).

5.3 Approaches

We describe approaches for each team as follows, shown as a header format of **Team Name (number of submitted runs)**.

- **HUKB (3 runs)** [16] proposed a method to select relevant part from the articles (**HUKB-2**) and a new data augmentation method (**HUKB-1**) in addition to their system in COLIEE 2021 (**HUKB-3**) which uses an ensemble of BERT with data augmentation, extracting judicial decision sentences, creating positive/negative data from articles.
- **JNLP (3 runs)** [3] compared ELECTRA, RoBERTa, and LegalBERT, which is pretrained using large legal English texts. They also compared impacts of negation data augmentation, and paragraph-level entailments.
- **KIS (3 runs)** [6] employed an ensemble of their rule-based method using predicate-argument structures which extends their previous work, and BERT-based methods. Their BERT-based methods commonly use data augmentation (**KIS2**), with data selection (**KIS1**), and with person name inference (**KIS3**). They also employed an ensemble of different trials of fine-tunings.
- **LLNTU (2 runs)** [7] restructured given data to a dataset of the disjunctive union strings from training queries and articles, and established a longest uncommon subsequence similarity comparison model, without stopwords (**LLNTUdiffSim**), and with stopwords (**LLNTUdeNgram**). One of their runs was retracted because they used their dataset via web crawling that could potentially include the correct answers of the test dataset.
- **OvGU (3 runs)** [14] employed an ensemble of graph neural networks (GNNs) as their previous work (**OvGU1**), concatenated with referring textbook nodes (**OvGU2** and averaging sentence embeddings (**OvGU3**). No submission for the past training datasets.
- **UA (3 runs)** [9] provides no description for Task 4.

5.4 Results

Table 5 shows evaluation results of Task 4, including the formal run results and training dataset. The evaluation results of the training dataset regard either of R02, R01, or H30 as test dataset, using corresponding former years' dataset (-R01, -H30, -H29) as training datasets; these configurations correspond to the formal runs of COLIEE 2021, 2020, and 2019, respectively.

6 Conclusion

We have summarized the systems and their performance as submitted to the COLIEE 2022 competition. For Task 1, UA submitted by the University of

Table 5. Evaluation results of submitted runs (Task 4). L: Dataset Language (J: Japanese, E: English), #: number of correct answers

Team	Submission ID	L	Formal Run		R02		R01		H30	
			#	Accuracy	#	Accuracy	#	Accuracy	#	Accuracy
N/A	Total	-	109	1.0000	81	1.0000	111	1.0000	70	1.0000
N/A	BaseLine	-	58	0.5320	43	0.5309	59	0.5315	36	0.5143
KIS	KIS2	J	74	0.6789	44	0.5432	71	0.6396	49	0.7000
HUKB	HUKB-1	J	73	0.6697	50	0.6173	73	0.6577	42	0.6000
KIS	KIS1	J	72	0.6606	48	0.5926	68	0.6126	46	0.6571
KIS	KIS3	J	72	0.6606	49	0.6049	68	0.6126	47	0.6714
HUKB	HUKB-2	J	69	0.6330	48	0.5926	74	0.6667	41	0.5857
HUKB	HUKB-3	J	69	0.6330	58	0.7160	72	0.6486	44	0.6286
LLNTU	LLNTUdeNgram	J	66	0.6055	47	0.5802	60	0.5405	33	0.4714
LLNTU	LLNTUdiffSim	J	63	0.5780	49	0.6049	56	0.5045	36	0.5143
OVGU	OVGU3	?	63	0.5780	-	-	-	-	-	-
UA	UA_e	?	59	0.5413	44	0.5432	69	0.6216	32	0.4571
UA	UA_r	?	59	0.5413	44	0.5432	69	0.6216	32	0.4571
UA	UA_structure	?	59	0.5413	44	0.5432	69	0.6216	32	0.4571
JNLP	JNLP1	J	58	0.5321	50	0.6173	59	0.5315	40	0.5714
JNLP	JNLP2	J	58	0.5321	49	0.6049	58	0.5225	40	0.5714
OVGU	OVGU2	?	58	0.5321	-	-	-	-	-	-
JNLP	JNLP3	J	56	0.5138	48	0.5926	65	0.5856	42	0.6000
OVGU	OVGU1	?	52	0.4771	-	-	-	-	-	-

Alberta team was the best performing team with an F1 score of 0.3715. In Task 2, the winning team combined T5-base and T5-3B models, and achieved an F1 score of 0.6783. For Task 3, the top ranked team is HUKB and achieved an F2 score of 0.8204. KIS was the Task 4 winner, with an Accuracy of 0.6789.

We intend to further improve the datasets quality in future editions of COLIEE so the tasks more accurately represent real-world problems.

Acknowledgements

This competition 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. Our work to create and run the COLIEE competition is also supported by our institutions: the National Institute of Informatics (NII), Shizuoka University and Hokkaido University in Japan, and the University of Alberta and the Alberta Machine Intelligence Institute in Canada. We also acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [DGECR-2022-00369, RGPIN-2022-0346]. This work was also supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Abolghasemi, A., Althammer, S., Hanbury, A., Verberne, S.: Dossier@coliee2022: Dense retrieval and neural re-ranking for legal case retrieval. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
2. Askari, A., Peikos, G., Pasi, G., Verberne, S.: Leibi@coliee 2022: Aggregating tuned lexical models with a cluster-driven bert-based model for case law retrieval. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
3. Bui, M.Q., Nguyen, C., Do, D.T., Le, N.K., Nguyen, D.H., Nguyen, T.T.T.: Using deep learning approaches for tackling legal's challenges (coliee 2022). In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
5. Fink, T., Recski, G., Kusa, W., Hanbury, A.: Statute-enhanced lexical retrieval of court cases for coliee 2022. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
6. Fujita, M., Onaga, T., Ueyama, A., Kano, Y.: Legal textual entailment using ensemble of rule-based and bert-based method with data augmentations including generation without excess or deficiency. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
7. Lin, M., Huang, S.C., Shao, H.L.: Rethinking attention: An attempting on revaluing attention weight with disjunctive union of longest uncommon subsequence for legal queries answering. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
8. Nigam, S.K., Goel, N.: nigam@coliee-22: Legal case retrieval and entailment using cascading of lexical and semantic-based models. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
9. Rabelo, J., Kim, M.Y., Goebel, R.: Semantic-based classification of relevant case law. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
10. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: COLIEE 2020: Methods for Legal Document Retrieval and Entailment, pp. 196–210 (06 2021). https://doi.org/10.1007/978-3-030-79942-7_13
11. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. In: The Review of Socionetwork Strategies. vol. 16, pp. 111–133 (04 2022). https://doi.org/10.1007/s12626-022-00105-z
12. Robertson, S.E., Walker, S.: Okapi/Keenbow at TREC-8. In: Proceedings of TREC-8. pp. 151–162 (2000)
13. Rosa, G.M., Bonifacio, L.H., Jeronymo, V., de Alencar Lotufo, R., Nogueira, R.: 3b parameters are worth more than in-domain training data: A case study in the legal case entailment task. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
14. Wehnert, S., Kutty, L., Luca, E.W.D.: Using textbook knowledge for statute retrieval and entailment classification. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
15. Wen, J., Zhong, Z., Bai, Y., Zhao, X., , Yang, M.: Siat@coliee-2022: Legal case retrieval with longformer-based contrastive learning. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)
16. Yoshioka, M., Suzuki, Y., Aoki, Y.: Hukb at the coliee 2022 statute law task. In: Sixteenth International Workshop on Juris-informatics (JURISIN) (2022)

Billions of Parameters Are Worth More Than In-domain Training Data: A case study in the Legal Case Entailment Task

Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo,
Roberto Lotufo, and Rodrigo Nogueira

NeuralMind, Brazil
University of Campinas (UNICAMP), Brazil

Abstract. Recent work has shown that language models scaled to billions of parameters, such as GPT-3, perform remarkably well in zero-shot and few-shot scenarios. In this work, we experiment with zero-shot models in the legal case entailment task of the COLIEE 2022 competition. Our experiments show that scaling the number of parameters in a language model improves the F1 score of our previous zero-shot result by more than 6 points, suggesting that stronger zero-shot capability may be a characteristic of larger models, at least for this task. Our 3B-parameter zero-shot model outperforms all models, including ensembles, in the COLIEE 2021 test set and also achieves the best performance of a single model in the COLIEE 2022 competition, second only to the ensemble composed of the 3B model itself and a smaller version of the same model. Despite the challenges posed by large language models, mainly due to latency constraints in real-time applications, we provide a demonstration of our zero-shot monoT5-3b model being used in production as a search engine, including for legal documents. The code for our submission and the demo of our system are available at https://github.com/neuralmind-ai/coliee and https://neuralsearchx.neuralmind.ai, respectively.

Keywords: Zero-shot learning · Legal NLP · Legal case entailment

1 Introduction

A common practice in machine learning competitions and leaderboards is to use training and test data from the same distribution. However, winning models often fail when applied to real-world applications, since real-world data is usually more diverse and may differ from the data seen during training [6, 7]. Thus, our goal is to develop models capable of generalizing to examples outside of the training data distribution.

In the legal domain, it is known that the low availability of annotated data is a serious problem [8, 23], making it very difficult to develop high-performance models for legal tasks. An alternative to address this problem is to transfer knowledge using models fine-tuned on related domains where annotated data is abundant. However, models tend to underperform in scenarios that involve

different data distributions (e.g., text domains) between training and inference [22, 2]. Developing models capable of overcoming this instability is an important challenge to be addressed, as in the long term it is desirable that our models are robust enough to take advantage of previously learned information to generalize well to new domains and tasks.

It has also been shown that the high number of parameters in larger language models, up to billions, results in greater generalization ability and consequently superior zero-shot performance compared to smaller models [3, 30, 27]. Rabelo et al. [17] were the first to apply zero-shot techniques to the legal case entailment task, and Rosa et al. [26] showed that zero-shot models can outperform fine-tuned ones on this task. In this work, we show that performance on the legal case entailment task can be further improved simply by scaling the language model, without the need for any adaptation to the legal domain or fine-tuning on the in-domain training dataset.

2 Related Work

Zero-shot and few-shot models have demonstrated competitive performance compared to fine-tuned ones in many textual tasks. For example, Pradeep et al. [16] used a Text-To-Text Transfer Transformer (T5) model fine-tuned only on a general domain ranking dataset to achieve the best or second-best performance in multiple domain-specific tasks such as Precision Medicine [24] and TREC-COVID [32]. Rosa et al. [26] showed for the legal case entailment task that language models without any fine-tuning on the target task can perform better than models fine-tuned on the task itself. They argued that, given limited labeled data for training and a held-out test set, such as in the COLIEE competition, a zero-shot model fine-tuned only on a large and diverse general domain dataset may be able to generalize better to unseen data than models fine-tuned on a small domain-specific dataset. In this work, we evaluate this zero-shot approach, but using a much larger model.

Larger language models have recently shown stronger zero-shot and few-shot capabilities than smaller models. For example, Brown et al. [3] presented GPT-3, a language model with 175 billion parameters. They evaluated the model learning ability in few-shot scenarios and showed that scaling up language models greatly improves performance, sometimes even achieving competitive results compared to models fine-tuned on hundreds of thousands of examples. Sanh et al. [27] developed a system to map any natural language task into human-readable prompted forms and converted several supervised datasets into that format. They fine-tuned a T5 model of 11 billion parameters on this new multitask dataset and found that the model achieves strong zero-shot performance across multiple held-out datasets, including outperforming models up to 16 times its size. Wei et al. [30] introduced *instruction tuning*, a new approach to improve the zero-shot learning ability of larger language models, in which task-specific instructions are given to models using natural language. This approach substan-

tially improved performance and outperformed the GPT-3 model on multiple datasets in zero-shot scenarios.

Zhong et al. [33] proposed *meta-tuning*, a method for optimizing zero-shot learning of large language models by fine-tuning on a collection of datasets in a question answering format. The model outperformed question answering models of the same size, including the previous state-of-the-art method. Furthermore, recent results suggest that we still did not reach the limits of this trend of increasing model size. To understand the impact of scaling in few-shot learning, Chowdhery et al. [5] introduced the Pathways Language Model (PaLM), a language model with 540 billion parameters that achieved state-of-the-art few-shot results in hundreds of language tasks. Furthermore, PaLM 540B outperformed fine-tuned state-of-the-art models in multiple tasks and also surpassed the average human performance on the BIG-bench benchmark.

Winata et al. [31] evaluated the few-shot learning ability of several GPT and T5 models in multilingual scenarios. They showed that given a few examples in English as context, language models can correctly predict test samples in languages other than English. Lin et al. [12] fine-tuned a multilingual language model with 7.5 billion parameters in a wide range of languages to study the model's zero-shot and few-shot learning capabilities across multiple tasks. The model achieved state-of-the-art results in more than 20 languages, outperforming GPT-3 on a number of tasks such as multilingual common sense reasoning and natural language inference.

However, these promising results were not limited to models with billions of parameters. Mishra et al. [13] fine-tuned a BART model with 140 million parameters using samples containing instructions written in natural language and evaluated the model few-shot ability on unseen tasks. The results suggested that fine-tuning on a collection of tasks improves performance on unseen tasks, even for smaller models.

3 Legal Case Entailment Task

The Competition on Legal Information Extraction/Entailment (COLIEE) [18, 20, 19, 10, 9] is an annual competition whose goal is to improve and evaluate the performance of automated systems in legal tasks.

Legal case entailment is the second task of COLIEE 2022, which consists of identifying a set of candidate paragraphs from previous judicial decisions that corroborate with or are relevant to a given new judicial decision. The dataset comprises case laws collected from the Federal Court of Canada. The training data contains 525 judicial decisions, their respective candidate paragraphs that perhaps may be relevant to the given decision, and a set of labels containing the candidate paragraphs that entail the judicial decision. The test set contains 100 judicial decisions and only their candidate paragraphs, since labels are only revealed after the competition. The dataset has an average of 35 candidate paragraphs per judicial decision, of which only one is relevant on average. In Table 1, we show the statistics of the COLIEE 2021 and COLIEE 2022 datasets.

Table 1. COLIEE dataset statistics.

	2021		2022	
	Train	**Test**	**Train**	**Test**
Examples (base cases)	425	100	525	100
Avg. # of candidates / example	35.80	35.24	35.69	32.78
Avg. positive candidates / example	1.17	1.17	1.14	-
Avg. of tokens in base cases	37.51	32.97	36.64	32.21
Avg. of tokens in candidates	103.14	100.83	102.71	104.61

The micro F1 score is the official metric in this task:

$$\textbf{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{1}$$

where *Precision* is the number of candidate paragraphs correctly retrieved for all judicial decisions divided by the number of paragraphs retrieved for all judicial decisions, and *Recall* is the number of candidate paragraphs correctly retrieved for all judicial decisions divided by the number of relevant candidate paragraphs for all judicial decisions.

4 Method

In this section, we describe monoT5, which is the model used throughout this work. MonoT5 is an adaptation of the T5 model [21] designed by Nogueira et al. [15]. The authors proposed to fine-tune the model on the MS MARCO dataset [1] to generate the tokens "true" or "false" depending on the relevance of a document to a query. During inference, monoT5 generates a score that measures this relevance by applying a softmax function to the logits of the tokens "true" and "false", and then considering the probability of the token "true" as the final score. The model is close to or at the state of the art on datasets such as TREC-COVID [28], TREC 2020 Precision Medicine [16] and Robust04 [29].

We extend our zero-shot approach developed for the COLIEE 2021 competition [26], but now we experiment with a much larger model, which has billions of parameters. We refer to this approach as zero-shot because the models were only fine-tuned on general domain text coming from MS MARCO and directly evaluated on COLIEE's test set, which is made up of legal texts. We use the T5-base and T5-3B models available at Hugging Face,[1] and already fine-tuned for 10k steps on MS MARCO, which corresponds to almost one epoch of its training set. We refer to the resulting models as monoT5-base-zero-shot and monoT5-3b-zero-shot.

At inference time, monoT5 uses the following input template:

[1] https://huggingface.co/castorini/monot5-base-msmarco-10k, https://huggingface.co/castorini/monot5-3B-msmarco-10k

$$\text{Query: } q \quad \text{Document: } d \quad \text{Relevant:} \tag{2}$$

where q represents a fragment of a judicial decision from a new legal case and d represents one of the candidate paragraphs of existing legal cases that may or may not be relevant to the given decision.

The model receives the input prompt and estimates a score s that quantifies the relevance of a candidate paragraph d to a fragment of a judicial decision q. That is,

$$s = P(\text{Relevant} = 1|d, q). \tag{3}$$

The final score for each input is the probability assigned by the model to the token "true". After computing all scores, we apply the method described in Section 4.1 to select the relevant candidate paragraphs.

4.1 Answer Selection

The models used in the experiments estimate a score for each pair of judicial decision and candidate paragraph. To select the final set of paragraphs for a given judicial decision, we apply three rules:

- Select paragraphs whose scores are above a threshold α;
- Select the top β paragraphs with respect to their scores;
- Select paragraphs whose scores are at least γ of the top score.

We use grid search to find the best values for α, β, γ on the training set of the COLIEE 2022 dataset. We sweep $\alpha = [0, 0.1, ..., 0.9]$, $\beta = [1, 2..., 10]$, and $\gamma = [0, 0.1, ..., 0.9, 0.95, 0.99, 0.995, ..., 0.9999]$.

Note that our hyperparameter search includes the possibility of not using the first rule or the third rule if $\alpha = 0$ or $\gamma = 0$ are chosen, respectively.

4.2 Ensemble

We use the following approach to combine the answers from two models. First, we concatenate the answers provided by each model. Second, we remove duplicates while preserving the highest score. We then apply the answer selection method explained in Section 4.1 to select the final set of answers. The goal is to keep only answers with a certain degree of confidence (i.e., score).

5 Results

We present our results in Table 2. The BM25 method scores above the median of submissions in the COLIEE 2020 and 2021 datasets (row 2 vs. 1a), suggesting that BM25 is a strong baseline and agrees with the results from other competitions [16, 25]. The median F1 score for 2022 submissions is considerably higher

Table 2. F1 scores on the test sets of the legal case entailment task of COLIEE 2020, 2021 and 2022. Our best single model for each year is in bold.

Description	Submission name	Params	2020	2021	2022
(1a) Median of submissions		-	0.5718	0.5860	0.6391
(1b) Best of 2020 [14]	JNLP.task2.BMWT	-	0.6753	-	-
(1c) 2nd best team of 2021 [11]	UA_reg_pp	-	-	0.6274	-
(1d) 2nd best team of 2022	run2_bert_amr	-	-	-	0.6694
(2) BM25 [26]		-	0.6046	0.6009	-
(3) DeBERTa [26]	DeBERTa	350M	**0.7094**	0.6339	-
(4) monoT5-base-zero-shot	monot5-base	220M	-	0.7192	0.6325
(5) monoT5-large [26]	monoT5	770M	0.6887	0.6610	-
(6) monoT5-large-zero-shot [26]		770M	0.6577	0.6872	-
(7) monoT5-3b-zero-shot	monot5-3b	3,000M	-	**0.7477**	**0.6757**
(8) Ensemble of (3) and (5) [26]	DebertaT5	1,120M	0.7217	0.6912	-
(9) Ensemble of (4) and (7)	monot5-ensemble	3,220M	-	0.7567	0.6783

than in previous years, approaching our best method of 2021, indicating that other teams are using pretrained transformer models for this task.

Results in the COLIEE 2021 test set show that scaling the model size from base to large leads to a decrease in performance. It is not clear why this happens, but it is not the first time that T5-base achieves better performance than T5-large on an entailment task [4]. However, scaling to monoT5-3B provides a significant performance gain, easily outperforming all single models and confirming the trend that much larger models bring better model quality.

For the COLIEE 2022 competition, we submitted three runs using different sizes of monoT5-zero-shot models: a monoT5-base (row 4), a monoT5-3B (row 7) and an ensemble between monoT5-base and monoT5-3B (row 9). Performance consistently increases with model size and two of our submissions (rows 7 and 9) score above the median of submissions and above the second best team in the competition. Furthermore, our ensemble method effectively combines the predictions of different monoT5 models, achieving the best performance among all submissions (row 9).

5.1 Ablation of the Answer Selection Method

Table 3. Ablation on the 2021 test set of the answer selection method.

Model	F1 Score	Precision	Recall	α, β, γ
monoT5-base-zero-shot (no rule)	0.7004	0.7600	0.6495	0, 1, 0
monoT5-base-zero-shot	0.7192	0.7387	0.7008	0, 4, 0.995
monoT5-3b-zero-shot (no rule)	0.7373	0.8000	0.6838	0, 1, 0
monoT5-3b-zero-shot	0.7477	0.7904	0.7094	20, 3, 0.999

In Table 3, we show the ablation result of the answer selection method proposed in Section 4.1. Our baseline answer selection method uses only the candidate paragraph with the highest score as the final answer set, i.e., $\alpha = \gamma = 0$ and $\beta = 1$. For all models, the proposed answer selection method gives improvements of at least one F1 point over the baseline.

6 Conclusion

In this work, we explored the zero-shot ability of a multi-billion parameter language model in the legal domain. We showed that, for the legal case entailment task, language models without any fine-tuning on the target dataset and target domain can outperform models fine-tuned on the task itself. Furthermore, our results support the hypothesis that scaling language models to billions of parameters improves zero-shot performance. This method has the potential to be extended to other legal tasks, such as legal information retrieval and legal question answering, especially in limited annotated data scenarios.

However, while large language models have achieved state-of-the-art performance in many tasks, one of their main disadvantages is inference speed, primarily due to their computationally intensive nature, which can make them expensive to deploy in real-world applications. Addressing this problem is an important challenge that requires the development of techniques that seek to optimize the latency of the models.

References

1. Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268v3 (2018)
2. Ben-David, E., Oved, N., Reichart, R.: Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. arXiv preprint arXiv:2102.12206 (2021)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
4. Carmo, D., Piau, M., Campiotti, I., Nogueira, R., Lotufo, R.: Ptt5: Pretraining and validating the t5 model on brazilian portuguese data (2020)
5. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J.,

Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)

6. Dash, T., Chitlangia, S., Ahuja, A., Srinivasan, A.: A review of some techniques for inclusion of domain-knowledge into deep neural networks. arXiv preprint arXiv:2107.10295 (2021)
7. Farahani, A., Voghoei, S., Rasheed, K., Arabnia, H.R.: A brief review of domain adaptation. arXiv preprint arXiv:2010.03978 (2020)
8. Hendrycks, D., Burns, C., Chen, A., Ball, S.: Cuad: An expert-annotated nlp dataset for legal contract review. arXiv preprint arXiv:2103.06268 (2021)
9. Kano, Y., Kim, M., Goebel, R., Satoh, K.: Overview of COLIEE 2017. In: COLIEE 2017 (EPiC Series in Computing, vol. 47). pp. 1–8 (2017)
10. 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: JSAI International Symposium on Artificial Intelligence. pp. 177–192 (2018)
11. Kim, M., Rabelo, J., Goebel, R.: Bm25 and transformer-based legal information extraction and entailment. Proceedings of the COLIEE Workshop in ICAIL (2021)
12. Lin, X.V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Koura, P.S., Chaudhary, V., O'Horo, B., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., Li, X.: Few-shot learning with multilingual language models. arXiv preprint arXiv:2112.10668 (2021)
13. Mishra, S., Khashabi, D., Baral, C., Hajishirzi, H.: Natural-instructions: Benchmarking generalization to new tasks from natural language instructions. arXiv preprint arXiv:1611.09268 (2021)
14. 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. arXiv preprint arXiv:2011.08071 (2020)
15. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings. pp. 708–718 (2020)
16. Pradeep, R., Ma, X., Zhang, X., Cui, H., Xu, R., Nogueira, R., Lin, J.: H2oloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. Corpus **5**(d3), d2 (2020)
17. Rabelo, J., Kim, M., Goebel, R.: Application of text entailment techniques in coliee 2020. International Workshop on Juris-informatics (JURISIN) associated with JSAI International Symposia on AI (JSAI-isAI) (2020)
18. Rabelo, J., Goebel, R., Kim, M.Y., Kano, Y., Yoshioka, M., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. Rev Socionetwork Strat (2022). https://doi.org/10.1007/s12626-022-00105-z (2022)
19. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: A summary of the COLIEE 2019 competition. In: JSAI International Symposium on Artificial Intelligence. pp. 34–49 (2019)

20. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: Coliee 2020: methods for legal document retrieval and entailment. In: JSAI International Symposium on Artificial Intelligence. pp. 196–210. Springer (2020)
21. 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
22. Ramponi, A., Plank, B.: Neural unsupervised domain adaptation in nlp—a survey. arXiv preprint arXiv:2006.00632 (2020)
23. Ratnayaka, G., de Silva, N., Perera, A.S., Pathirana, R.: Effective approach to develop a sentiment annotator for legal domain in a low resource setting. In Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, pages 252–260, Hanoi, Vietnam. Association for Computational Linguistics. (2020)
24. Roberts, K., Demner-Fushman, D., Voorhees, E., Hersh, W., Bedrick, S., Lazar, A.J., Pant, S.: Overview of the trec 2019 precision medicine track. The ... text REtrieval conference : TREC. Text REtrieval Conference **26** (2019)
25. Rosa, G.M., Rodrigues, R.C., , Nogueira, R., Lotufo, R.: Yes, bm25 is a strong baseline for legal case retrieval. Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (2021)
26. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: To tune or not to tune? zero-shot models for legal case entailment. ICAIL'21, Eighteenth International Conference on Artificial Intelligence and Law, June 21–25, 2021, São Paulo, Brazil (2021)
27. Sanh, V., Webson, A., Raffel, C., Bach, S.H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T.L., Raja, A., Dey, M., Bari, M.S., Xu, C., Thakker, U., Sharma, S.S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M.T.J., Wang, H., Manica, M., Shen, S., Yong, Z.X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J.A., Teehan, R., Bers, T., Biderman, S., Gao, L., Wolf, T., Rush, A.M.: Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207 (2021)
28. Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I., Wang, L.L.: Trec-covid: Constructing a pandemic information retrieval test collection. arXiv preprint arXiv:2005.04474 (2020)
29. Voorhees, E.M.: Overview of the trec 2004 robust track. Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, November 16-19, 2004 (2004)
30. Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=gEZrGCozdqR
31. Winata, G.I., Madotto, A., Lin, Z., Liu, R., Yosinski, J., Fung, P.: Language models are few-shot multilingual learners. arXiv preprint arXiv:2109.07684 (2021)
32. Zhang, E., Gupta, N., Nogueira, R., Cho, K., Lin, J.: Rapidly deploying a neural search engine for the covid-19 open research dataset. In: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020 (2020)
33. Zhong, R., Lee, K., Zheng Zhang, D.K.: Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. arXiv preprint arXiv:2104.04670 (2021)

Semantic-based Classification of Relevant Case Law

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. The challenge of information overload in the legal domain increases every day. COLIEE proposes four challenges which are intended to encourage the development of systems and methods to alleviate that pressure: a case law retrieval (Task 1) and entailment (Task 2), and a statute law retrieval (Task 3) and entailment (Task 4). In this paper, we describe our method for Task 1, which was ranked first among all participants in the COLIEE 2022 edition. Our approach relies on measuring the similarity between paragraphs of legal cases by generating feature vectors based on those similarities, and then using a classifier to determine if those cases should be noticed or not. We also apply simple pre- and post-processing heuristics to generate the final results.

Keywords: Legal textual retrieval · Semantic text representation · Document similarity · Binary classification · Imbalanced datasets.

1 Introduction

Every day, large volumes of legal data are produced by law firms, law courts, independent attorneys, legislators, and many other sources. In that scenario, management of legal information becomes manually intractable, and requires the development of tools which automatically or semi-automatically aid legal professionals to handle the information overload. In the COLIEE competition, we face four facets of that challenge: case law retrieval, case law entailment, statute law retrieval and statute law entailment. Here we provide the details of our approaches to those tasks, evaluate the results achieved and comment on future work to further improve our models.

For the case law retrieval (Task 1), our approach relies on a transformers-based model to create a multi-dimension numeric representation of each case paragraph, then we calculate the cosine distances between each paragraph from a query case and a candidate case. Next, we create a histogram from the resulting distances and train a binary classifier with those inputs. Besides that, we do some simple pre-processing (removal of French fragments) and post-processing (removal of candidates which have dates more recent than the query case, establish a maximum number of noticed cases per query case based on priors drawn

from the training dataset and apply a minimum confidence score for the classifier outputs) to generate the final results. This approach achieved an f1-score of 0.3715 in the official test dataset, the best among all competitors in Task 1 of the COLIEE 2022 edition.

The paper is organized as follows: section 2 presents a brief state-of-the-art analysis; section 3 describes our method in deeper detail; section 4 analysis the results; and section 5 brings some final remarks and proposes some future work.

2 Literature Review

Current methods for legal information retrieval rely on traditional IR approaches and, more recently, on transformers-based techniques. In COLIEE 2021, the main approaches revolved around those two main topics:

Li et al. [6] proposed a pipeline method based on statistics features and semantic understanding models, which enhances the retrieval method with both recall and semantic ranking.

Schilder et al. [10] applied a two-phase approach for task 1: first, they generate a candidate set which is optimized for recall. It. attempts to include all true noticed cases while removing some of the false candidates. The second step is a binary classifier which receives as input the pair $(query\ case, candidate\ case)$ and predicts whether they represent a true noticed relationship.

Rosa et at. [9] presented a vanilla application of BM25 to the case law retrieval problem. They do that by first indexing all base and candidate cases present in the given dataset. Before indexing, each document is split into segments of texts using a context window of 10 sentences with overlapping strides of 5 sentences (the 'candidate case segments'). BM25 is then used to retrieve candidate case segments for each base case segment. The relevance score for a $(base\ case, candidate\ case)$ pair is the maximum score among all their base case segment and candidate case segment pairs. The candidates are then ranked according to threshold-based heuristics.

Ma et al. [7] applied two methods for the case law retrieval task: the first one is a traditional language model for IR, which consists of an application of LMIR on a pre-processed version of the dataset. The TLIR team did not use the full case contents, but rather make use of the tags inserted in the text to indicate a fragment has been suppressed to identify the most relevant pieces. This specific approach ranked first place among all task 1 competitors, showing traditional IR methods achieve good results in the case law retrieval task. The second approach used by the TLIR team was a transformer based method, which splits a document into paragraphs and computes interactions between paragraphs using BERT. Compared with other neural models, BERT-PLI can take long text representations as an input without truncating them at some threshold.

Althammer et al. [1] combined retrieval methods with neural re-ranking methods using contextualized language models like BERT. Since the cases are typically long documents exceeding BERT's maximum input length, the authors adopt a two phase approach. The first phase combined lexical and dense retrieval methods on the paragraph-level of the cases. Then, they re-ranked the candidates by summarizing the cases and applying a fine-tuned BERT re-ranker on said summaries.

3 Our Method

3.1 Dataset Analysis

The training dataset is composed by 4,415 files, with 900 of those being query cases. There is a total of 4,206 noticed cases, an average of 4.67 cases noticed per query case. One case may potentially be noticed by more than one query case, but we found that to be rare: there are 3,126 cases that are noticed only once. Among cases that are noticed more than once, 293 are noticed twice, 66 three times, 20 are noticed four times, and 29 are notice five or more times. In the provided test dataset there were 300 query cases and a total of 1,563 files.

3.2 Details of the Approach

The approach we applied to the case law retrieval task in COLIEE 2022 relied on the usage of sentence-transformer model (see more details below) to generate a multidimensional numeric representation of text. This model is applied to each paragraph from the query case and every candidate case. Then, the distances between the 768-dimension vectors from the query paragraphs and the candidate paragraphs are measured using the cosine distance. A 10-bin histogram of those distances is generated and a Gradient Boosting [2] binary classification model is trained on those inputs.

Given the formulation of the problem, we had to make some choices to produce a reasonable training dataset: since the test set contains a total of approximately 1,500 query cases being 300 query cases, we assumed we should generate a training dataset with around 1,000 negative samples per query case. So we needed to down sample the negative samples in the training dataset to 1,000. At the same time, the positive class is far under represented (less than 5 samples per query case in average), so we over sampled those examples by simple replication.

Prior to that, we implemented some simple pre-processing steps:

Removal of French contents through a language identification model based on a naive Bayesian filter [8];

Splitting of input text into paragraphs based on simple pattern matching relying on the common format used in cases;

Extraction of dates mentioned in the cases through the application of a named entity recognition model [3]. These dates are used later to filter out candidate cases which mention dates more recent than the most recent date mentioned in a query case, under the assumption those candidates cannot be a true noticed case because they are likely more recent than the query case.

At inference time we:

Date filtering: apply the same date pre-processing steps mentioned above;

Histograms : generate histograms for every pair of query document and each candidate which does not contain dates more recent than the query document dates;

Apply model: run those histograms as inputs to our trained model.

Based on an analysis of the training dataset, we also apply some simple post-processing steps:

Number of noticed cases per query case: the average number of noticed cases per query case in the training dataset is 4.67, so we establish a range of 1 to 10 maximum noticed cases per query case;

Confidence score : we also establish a minimum confidence score for the classifier, disregarding outputs which are below that threshold;

Repeating noticed cases : if the same case is noticed across many different query cases, we also remove that noticed case from our final answer as it is observed in the training dataset that this is not a common situation.

We experimented with a range of parameters for each one of those post-processing criteria and selected the 3 combinations which produced the best output in a validation set containing 50 query cases.

We also explored two other approaches which we could not use in the final submissions due to limited access to hardware resources: usage of the FAISS library [4] for semantic similarity searches, and another one based on fine-tuning an ALBERT model [5] which showed promising results in preliminary evaluations. We intend to further explore these approaches in future editions of the COLIEE competition.

Sentence-Transformer Model The model used to produce 768-dimensional representations for the case paragraphs was the HuggingFace sentence-transformers/ all-mpnet-base-v2 model. That model was trained on very large sentence level datasets using a self-supervised contrastive learning objective using the pre-trained microsoft/mpnet-base model as the base model and fine-tuning it on

a 1B sentence pairs dataset. The authors use a contrastive learning objective: given a sentence from the pair, the model should predict which out of a set of randomly sampled other sentences, was actually paired with it in the dataset.

4 Results

The results on the official COLIEE evaluation set are shown in Table 1:

Team	File	F1 Score	Precision	Recall
UA	**pp_0.65_10_3.csv**	**0.3715**	**0.4111**	**0.3389**
UA	**pp_0.7_9_2.csv**	**0.3710**	**0.4967**	**0.2961**
siat	siatrun1.txt	0.3691	0.3005	0.4782
siat	siatrun3.txt	0.3680	0.3026	0.4695
UA	**pp_0.65_6.csv**	**0.3559**	**0.3630**	**0.3492**
siat	siatrun2.txt	0.2964	0.2522	0.3595
LeiBi	run_bm25.txt	0.2923	0.3000	0.2850
LeiBi	run_weighting.txt	0.2917	0.2687	0.3191
JNLP	run3.txt	0.2813	0.3211	0.2502
nigam	bm25P3M3.txt	0.2809	0.2587	0.3072
JNLP	run2.txt	0.2781	0.3144	0.2494
DSSR	DSSR_01.txt	0.2657	0.2447	0.2906
JNLP	run1.txt	0.2639	0.2446	0.2866
DSSR	DSSR_03.txt	0.2461	0.2267	0.2692
TUWBR	TUWBR_LM_law	0.2367	0.1895	0.3151
LeiBi	run_clustering.txt	0.2306	0.2367	0.2249
TUWBR	TUWBR_LM	0.2206	0.1683	0.3199
nigam	sbertP3M3RS.txt	0.1542	0.1420	0.1686
nigam	s2vecP3M3RS.txt	0.1484	0.1367	0.1623
DSSR	DSSR_02.txt	0.1317	0.1213	0.1441
LLNTU	2022.task1.LLNTUtfidCos	0.0000	0.0000	0.0000
LLNTU	2022.task1.LLNTUtanadaT	0.0000	0.0000	0.0000
LLNTU	2022.task1.LLNTU3q4clii	0.0000	0.0000	0.0000
Uottawa	Task1Run3_UottawaLegalBert.txt	0.0000	0.0000	0.0000
Uottawa	Task1Run1_UottawaMB25.txt	0.0000	0.0000	0.0000
Uottawa	Task1Run2_UottawaSentTrans.txt	0.0000	0.0000	0.0000

Table 1. Official results for the Case Law Retrieval task.

Our best result was achieved with the following post-processing parameters: minimum confidence score = 0.65, maximum noticed cases = 10, maximum number of repeated noticed cases = 3[4]. Our second best score had similar parameters (0.7, 9 and 2 respectively). In the third submission we only used the minimum score and maximum number of noticed cases (0.65 and 6, respectively). This

[4] We simply remove noticed cases which appear in more than the maximum allowed query cases. An obvious improvement is to keep just the highest scoring noticed cases

provided a more balanced trade-off between precision and recall, as opposed to the first two which had a high precision but a lower recall. This is an interesting characteristic for real world applications, as one could make an informed decision on how to tweak parameters depending on which metric is more important for their scenario.

5 Final Remarks

In this paper, we presented our approach for the Case Law Retrieval task in COLIEE 2022. Our team was ranked first among all competitors in that task (9 teams, which sent 26 submissions altogether). As future work, we intend to refine our post-processing approach to retain part of the cases identified as noticed when they are present in the output of many different query cases. That would be a simple extension of our current method. However, the main intended extension to this work is a more disruptive approach fully based on transformer-based models, which we had to abandon in this edition of the competition due to lack of hardware resources.

Acknowledgments

This research was supported by the Alberta Machine Intelligence Institute (AMII), the University of Alberta, and the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference numbers DGECR-2022-00369 and RGPIN-2022-0346].

References

1. Althammer, S., Askari, A., Verberne, S., Hanbury, A.: Dossier@coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In: Proceedings of the COLIEE Workshop in ICAIL (2021)
2. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
3. Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1373–1378. Association for Computational Linguistics, Lisbon, Portugal (September 2015)
4. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
5. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. CoRR **abs/1909.11942** (2019), http://arxiv.org/abs/1909.11942
6. Li, J., Zhao, X., Liu, J., Wen, J., Yang, M.: Siat@coliee-2021: Combining statistics recall and semantic ranking for legal case retrieval and entailment. In: Proceedings of the COLIEE Workshop in ICAIL (2021)

7. Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., Ma, S.: Retrieving legal cases from a large-scale candidate corpus. In: Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL) (2021)
8. Nakatani, S.: Language detection library for java (2010), https://github.com/shuyo/language-detection
9. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: Yes, bm25 is a strong baseline for legal case retrieval. In: Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL) (2021)
10. Schilder, F., Chinnappa, D., Madan, K., Harmouche, J., Vold, A., Bretz, H., Hudzina, J.: A pentapus grapples with legal reasoning. In: Proceedings of the COLIEE Workshop in ICAIL (2021)

HUKB at the COLIEE 2022 Statute Law Task

Masaharu Yoshioka[1,2], Youta Suzuki[2], and Yasuhiro Aoki[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
aoki.yasuhiro.k4@elms.hokudai.ac.jp

Abstract. HUKB group is participating in the Statute Law task (task 3 and 4) in COLIEE 2022. For the task 3, we propose a method that utilize three different IR systems. Our new proposed IR system utilize the similarity of descriptions about judicial decision between questions and articles. In addition to this new IR system, we also use ordinal keyword-based IR system (BM25) and the BERT-based IR system proposed in the COLIEE 2020. Due to the different characteristics of these systems, ensembled results has better recall without losing precision so much. Our system that ensembles the results of our new proposed IR and keyword-based IR system achieves best performance for the task 3. For the task 4, we extend our previous BERT-based entailment system (the best performance system of the COLIEE 2021) by using new data augmentation method and method to select relevant part from the articles. We discuss the characteristics of the system using the submitted results.

Keywords: Legal Information Retrieval · Query Analysis · Deep Neural Network · Keyword-based IR

1 Introduction

The competition on Legal Information Extraction/Entailment (COLIEE) [3, 2, 11, 7, 8, 5] serves as a forum to discuss issues related to legal information retrieval (IR) and entailment. There are two types of tasks in the COLIEE 2022. One is case law tasks (task 1 and 2), and the other is statute law task (task 3 and 4). This year, we participate the latter two tasks; statute law retrieval task (task 3) and legal textual entailment task (task4).

For the task 3, we proposed new statute law retrieval system that considers the similarity of judicial decision described in the questions and articles. Since this task is a preprocess for the task 4 to judge whether the question statement is true or not, existence of the judicial decision discussed in the question is important factor to find out relevant articles. For the task 4, we have extended our previous system for the COLIEE 2021 [6](the best performance system for the COLIEE 2021) by introducing new data augmentation method and method to select relevant part of articles from all articles.

In this paper, we introduce our methods for task 3 and 4 in detail and discuss the characteristics of the system using the evaluation results of the submitted runs.

2 Tools and Settings

In this section, we introduce tools and settings used for task 3 and 4. For the task 3, we use ordinal keyword-based IR system and BERT-based IR system. For the task 4, we use BERT-based system that classify the given relevant articles pair entails question or not. In this year, we use almost same setting for BERT used in task 3 and 4.

2.1 Keyword-based IR system

Since queries of statute law retrieval task (task 3) is extracted from the Japanese Bar-exam, the queries contain technical terms in the law domain. As a result, the keyword-based system that utilize such technical terms works well for the questions. However, there are following two problems in the keyword-based IR system.

(1) Missing judicial decision description
The most important point for discussing the entailment is description about judicial decision. However, previous keyword-based systems didn't pay any attention to this part, there are cases that the system returns articles that share many keywords without ones for judicial decision.
(2) Difficulty to handle reference to other articles
There are many articles that refer other articles in the Civil law. Most significant ones are "Mutatis Mutandis". It is difficult to find the pairs of such articles by using simple keyword-based IR system.

To tackle this problem, we propose a method to reconstruct civil law articles database by considering judicial decision and reference.

For the item (1), we select judicial decision parts from the articles and use this text for calculating the similarity between the judicial decision parts of the questions. For the articles that have multiple judicial decisions, we also make the sub-article database that split the articles for representing each judicial decision.

Followings are general methods to make such texts.

(A) Split the articles by using items. Most of the case when there are two or more sentences in the article, we split the text into sub-articles by considering the item, sentence, and existence of judicial decision. For the case of Fig. 1, underlined texts of the sentences are judicial decision parts. There are three sentences for the judicial decision, we split this article into three sub-articles.
(B) For making the article database, we use judicial decisions for all items as one for the article. For the article database, all text for the article is used for the text part of the article and all judicial parts are used for the judicial

decision part of the article. For the Fig. 1 case, text shown upper part of the figure except number of articles is used as text part and judicial parts (underlined text) are used for the judicial decision part.

(C) We also make sub-article database. For this data, one sub-article text has corresponding one judicial decision part (e.g., sub-article 299-(2)A uses judicial decision part of 299-(2)A in Fig. 1). However, for the text part, concatenation of previous text (including heading title of the article) is used for the text part, because important keywords are only used for the former part of the text.

For the item (2), we make new text by combining the information of original article and (an) refereed article(s).

Figure 2 shows an example of rewriting process for Mutatis Mutandis. In this case, referred article 299 is one for the 留置権 ("right of retention") and article 350 explains Mutatis Mutandis for 質件 ("right of pledges"). Therefore, combined articles replace 留置 ("retention") to 質 ("pledges").

We use Elasticsearch[3] with basic BM25 [9] settings with kuromoji_tokenizer[4] as an IR engine. Elasticsearch can use structured query that can use multiple indexes for calculating similarity. We use this function for using two indexes; one is for all text and the other is for judicial decision part.

For the article and sub-article database, we add title of the article (e.g., for the article 299 we use （留置権者による費用の償還請求） "(Demands for Reimbursement of Expenses by Holders of Rights of Retention)" and 350+299 we use title of article 350（留置権及び先取特権の規定の準用）"(Mutatis Mutandis Application of Provisions on Rights of Retention and Statutory Liens)") for the text of the articles. We also add judicial decision part text for the index of judicial decision.

At the time for making a query, we automatically extract a judicial decision part (detail of the method are explained below) from the question and calculate similarity score using two indexes. One is score calculated based on the similarity between the whole question text and article text. The other is score calculated based on the judicial decision part of the article and one of the questions extracted automatically. Those two scores are added for the final similarity score between the question and article.

For extracting the judicial decision parts, we extract judicial decision parts based on the method proposed in [12]. This method analyzes dependency structure of the question using CaboCha[4] and remove condition parts using clue keywords such as "～すれば"(if ...),"～としても"(in case of), "～場合、"(case), "～限る、"(limited to) . However previous method makes longer judicial decision part text, and it may not work well for the preliminary experiment. We select phrases from the root verb (or root adjective) by considering stop words such as "できる", "する", "ある" for verbs and "もの", "こと" "ため" for nouns. For each

[3] https://www.elastic.co/

[4] https://www.elastic.co/guide/en/elasticsearch/plugins/current/analysis-kuromoji-tokenizer.html

Japanese:
（留置権者による費用の償還請求）
第二百九十九条　1 留置権者は、留置物について必要費を支出したときは、所有者にその償還をさせることができる。
2　留置権者は、留置物について有益費を支出したときは、これによる価格の増加が現存する場合に限り、所有者の選択に従い、その支出した金額又は増価額を償還させることができる。ただし、裁判所は、所有者の請求により、その償還について相当の期限を許与することができる。

English:
(Demands for Reimbursement of Expenses by Holders of Rights of Retention)
Article 299 (1) If the holder of a right of retention incurs necessary expenses with respect to the thing retained, that holder may have the owner reimburse the same.
(2) If the holder of a right of retention incurs beneficial expenses with respect to the thing retained, to the extent that there is currently an increase in value as a result of the same, that holder may have the expenses incurred or the increase in value reimbursed at the owner's choice; provided, however, that the court may, at the request of the owner, grant a reasonable period of time for the reimbursement of the same.

Japanese:
（留置権者による費用の償還請求）
- 299-(1) 留置権者は、留置物について必要費を支出したときは、所有者にその償還をさせることができる。
- 299-(2)A 留置権者は、留置物について有益費を支出したときは、これによる価格の増加が現存する場合に限り、所有者の選択に従い、その支出した金額又は増価額を償還させることができる。
- 299-(2)B ただし、裁判所は、所有者の請求により、その償還について相当の期限を許与することができる。

English:
(Demands for Reimbursement of Expenses by Holders of Rights of Retention)
- 299-(1) If the holder of a right of retention incurs necessary expenses with respect to the thing retained, that holder may have the owner reimburse the same.
- 299-(2)A If the holder of a right of retention incurs beneficial expenses with respect to the thing retained, to the extent that there is currently an increase in value as a result of the same, that holder may have the expenses incurred or the increase in value reimbursed at the owner's choice;
- 299-2(B)provided, however, that the court may, at the request of the owner, grant a reasonable period of time for the reimbursement of the same.

Fig. 1. Make (sub)articles database with judicial decision

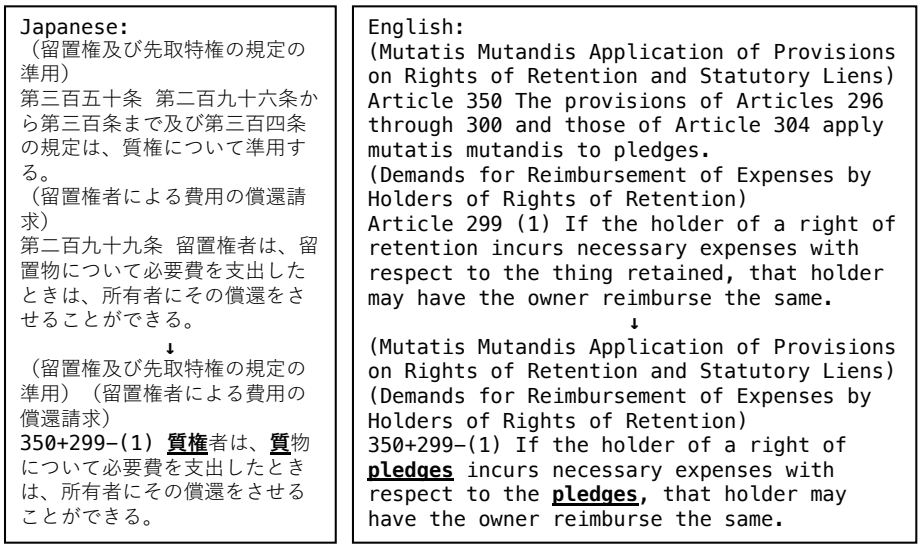


Fig. 2. Making combined articles using reference information

dependency branch of the root, two connected chunks are used for the judicial decision. If there is no dependency beach of the root (mostly caused by the stop word and/or removing condition parts), 2 connected chunks just before the root are used for the judicial decision parts.

Fig. 3 shows an example of this extraction process.

By using this method, we can emphasize the importance of the similarity of judicial decision parts.

However, based on the result of preliminary experiments, we found that there are several cases where the system fails to extract appropriate judicial decision parts. In such cases, emphasis on the judicial decision parts is harmful to find out relevant article. In addition, there are cases that question requires multiple articles with different characteristics; one is for judicial decision and the other is for analyzing the condition part. For such questions, emphasis on the judicial decision fails to find out the article for the condition part.

Therefore, we also make database without judicial decision. For the database, we use original article text for the text part of the article. We can use our new text for the text part, but preliminary experiments show IR results with original text database can find more complimentary answers (relevant answers that cannot be found by our proposed IR system with judicial decision information). This may be caused by the side-effect of rewriting the article text.

In addition to find another type of the complimentary answers, we also use BERT-based IR system introduced in the previous COLIEE [13]. Detail of the system will be discussed in the next section.

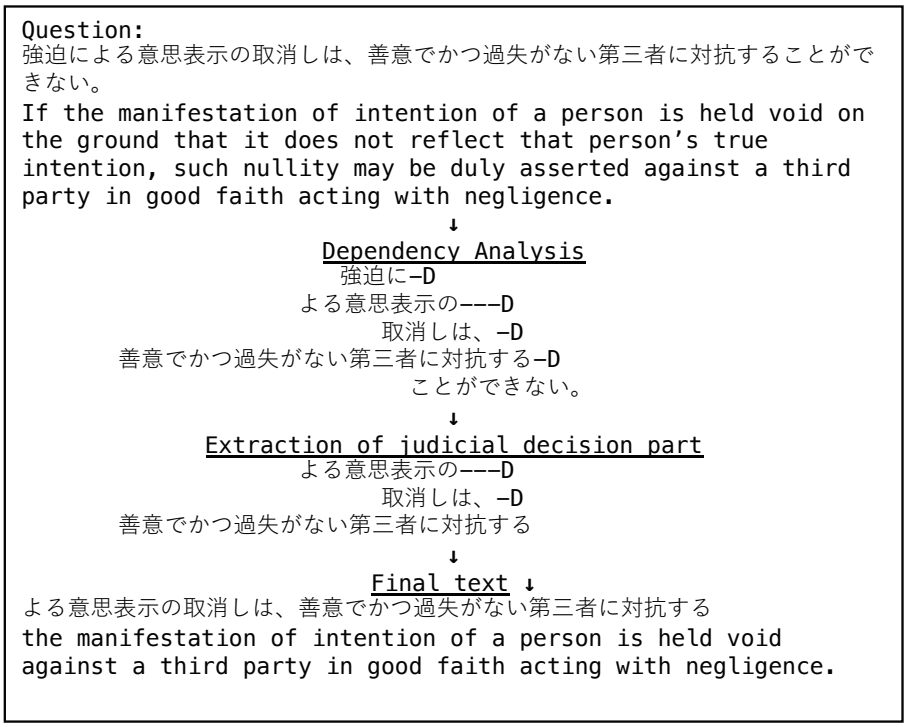


Fig. 3. Extraction of judicial decision parts from the question

For generating final retrieved results, it is necessary to decompose combined articles into original articles. For the post-process, we replace the result of combined articles into original ones. However, such results may contain redundant results. For such redundant cases, we select highest ranked one for the rank of the article. It is also true for the sub-article retrieved results.

2.2 BERT-based IR and entailment system

BERT [1] is as a deep neural network based natural language processing (NLP) tool that can be applicable to the variety of NLP tasks including IR and entailment. One of the characteristics of the BERT is usage of pretrained model trained by the general semantic understanding task. By using this model trained by large volume of texts, we can make a task specific model by using relatively small amount of data for training (fine-tuning) to have better performance. Recently BERT-based system is used for COLIEE and achieves best results for COLIEE 2021 task 4 [6]. For the task3, ordinal IR model such as BM25 and BERT-based model have similar performance with different characteristics.

We introduce BERT-based IR system and entailment system for COLIEE 2022.

BERT-based IR system For the IR task, BERT model is fine-tuned as a binary classification task that classify the pair of query and article are relevant or not [10]. Based on the trained model, the system can calculate a score for all documents for a given question. This score is used for ranking the documents.

This year's system uses BERT-Japanese [5] for the BERT-base model instead of the BERT model provided by Kyoto university[6].

For making the training data, we randomly select 50 negative examples from article dataset for each query and we also use (over-sample) relevant documents 5 times for making the balance of positive and negative examples.

For the target document database, we use both article dataset and sub-article database. All articles and sub-articles are sorted by the similarity score and remove redundant articles by using the same method for the keyword-based IR results.

We made three fine tuning models and return all top ranked documents using the given query.

BERT-based entailment system For the entailment task, BERT model is fine-tuned as a binary classification task that classify the pair of question and articles to check whether the articles entail the question or not. We use almost same settings for the previous COLIEE [6].

[5] https://github.com/cl-tohoku/bert-japanese with model https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

[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)`

One of the characteristics of [6] are data augmentation using article texts and ensembling results of BERT systems trained with different training sets.

The previous augmentation method decomposes all articles into smaller sentences for judicial decision. For example, “第七百八条 不法な原因のために給付をした者は、その給付したものの返還を請求することができない。ただし、不法な原因が受益者についてのみ存したときは、この限りでない。”(“Article 708 A person that has paid money or delivered thing for an obligation for an illegal cause may not demand the return of the money paid or thing delivered; provided, however, that this does not apply if the illegal cause existed solely in relation to the Beneficiary.”) has two sentences. Since it is difficult to understand the meaning of the second sentence by itself, we rewrite text to “不法な原因のために給付をした者は、その給付したものの返還を請求することができる。”(“If the illegal cause existed solely in relation to the Beneficiary, A person that has paid money or delivered thing for an obligation for an illegal cause may demand the return of the money paid or thing delivered”). After making sentences, a pair of the same sentence is a positive example and a pair of original sentences and flipped sentence generated by adding negation or removing negation is a negative example. We suppose this augmentation data is helpful to train the model for handling logical mismatch.

Utilization of this augmented data significantly improves the performance of the entailment system. However, based on the evaluation of different year' test data provided by COLIEE organizers, this BERT model has inconsistent performance for the different data set (The best performance system for year X does not mean good performance for other year). Therefore, [6] proposed an ensemble approach to use outputs of multiple BERT models trained by different training dataset.

This method utilizes three types of data: overall training data, validation data for ensemble and test data. Overall training data are randomly split into training and validation data for the BERT model training. For this year's task, we generate 20 models randomly by splitting overall data into training data (90%) and validation data (10%).

Validation data for ensemble is used for selecting appropriate ensemble model set as a combination of trained BERT models. Our ensemble models use the output of the entailment system that corresponds to a probability value for the “Yes” case. Original BERT system returns “Yes” if the value is larger than 0.5. We use average of this probability value for making ensemble results instead of majority voting, because we would like to emphasize the output with higher confidence larger value for “Yes” or smaller value for “No”.

Ensembled models are selected by using the entailment performance of the validation data for ensemble.

This year we have proposed following three methods to extend the previous one.

1. Construction of new augmentation data
   Due to the characteristics of rewriting method used in the previous COLIEE, The system does not have good numbers of examples that contains “ただし〜

この限りでない”(“provided”), because rewriting examples do not contains “ただし～この限りでない”(“provided”) for understanding the meaning of this phrase. Therefore to provide the examples for the phrase “ただし～この限りでない”(“provided”), we propose new method to make pairs of question and article. This method original article texts for the article parts. For example, “不法な原因のために給付をした者は、その給付したものの返還を請求することができる。”(“If the illegal cause existed solely in relation to the Beneficiary, A person that has paid money or delivered thing for an obligation for an illegal cause may demand the return of the money paid or thing delivered”) and original article texts “不法な原因のために給付をした者は、その給付したものの返還を請求することができない。ただし、不法な原因が受益者についてのみ存したときは、この限りでない。”(“A person that has paid money or delivered thing for an obligation for an illegal cause may not demand the return of the money paid or thing delivered; provided, however, that this does not apply if the illegal cause existed solely in relation to the Beneficiary.”) as a positive example.

2. Selection of useful sentences from articles for entailment.
   There are cases that the system uses (an) article(s) with multiple judicial decision. In such cases, the length of the article text becomes longer, and BERT cannot handle all text. Therefore, we propose a method to select a useful similar sentence calculated by the similarity score between the question and the sentences. This method also works well to include rewrote articles introduced in the last item. In such a case, it is not necessary to interpret the meaning of “ただし～この限りでない”(“provided”). For calculating the similarity between the sentence of the target article is calculated by using Elasticsearch with basic BM25 setting. The system selects one sentence with highest similarity score for each article.
3. Selection of the best ensemble model set using the loss function.
   Previous system selects the best performance sets by using the accuracy of the validation data for ensemble. Since this value is a discrete value, there are many tied sets and difficult to select the best one. To solve this problem, we use loss function to select the best one from the tied sets. The loss function is a simple function defined by equation 1 where g_i is a value for representing the correct label for question i ($g_i = 1, 0$ for “Yes” and “No” case, respectively) and $prob_i$ is a probability value calculated by the system for question i. We select the result with smallest $loss$ as the best one.

$$loss = \sum_{i=0}^{n} |g_i - prob_i| \tag{1}$$

3 Submitted runs and evaluation results

3.1 Task3

We submit three runs that combines output of three different IR systems; keyword-based IR system that uses reconstructed article database (New), keyword-based

IR system with original article text (Original) and BERT-based IR system (BERT). Ensemble method of the different models is very simple; just merging all candidates of the models.

We submit following three results by using these results.

HUKB1 Merge results of New, Original, and BERT
HUKB2 Merge results of New and Original
HUKB3 Use New results only

Table 1 shows evaluation results of our submitted runs and best runs for each team. HUKB2 achieves the best performance in terms of F2 measure. Total number of questions and relevant articles are 110 and 131 respectively.

Table 1. Evaluation results of the submitted runs (Task3)

ID	F2	Precision	Recall	Returns	Correct
HUKB2	0.8204	0.8180	0.8405	136	101
HUKB1	0.7908	0.6532	0.8901	223	109
OVGU_run3	0.7790	0.7781	0.8054	161	96
JNLP.longformer	0.7699	0.6865	0.8378	177	101
UA_TFIDF2	0.7638	0.8073	0.7641	115	90
HUKB3	0.7512	0.7890	0.7534	118	90
LLNTU0066cc	0.6416	0.6743	0.6391	113	74

Evaluation results of HUKB3 is not so good because this system returns multiple articles for the combined article cases. This system finds the appropriate results for R03-12-U that have "Mutatis Mutandis" articles. On the contrary there are cases that part of the combined article is relevant, but the other part is not relevant. It may be better to have a mechanism to check whether such combination is necessary or not.

Comparison between the evaluation results of HUKB2 and HUKB3 can be used for discussing the similarity and difference of New and Original database. Most of the case retrieved results are same and only 18 (136-118) articles are newly added as unique findings of original database. For this test set, 11 (101-90) articles out of 18 is correct one. There are two reasons why this system significantly improves the performance.

One is for the case for the side-effect of emphasis on judicial decision. There are cases that the judicial decision does not match the one in the query (mostly for the not entail case). For example, decision part of R03-05-O is similar for article 152, but the relevant article is article 149. However Original database can find this article by removing the side-effect of the emphasis.

The other is finding complimentary articles by using different method. For example, R03-20-E has two relevant articles: article 470 and article 442. New database finds article 442 that shares keywords for judicial decision. Original database found article 470, because it shares many keywords for explaining the situation.

For the HUKB1 the system achieves better recall, but the precision is not so good. BERT IR model can find the relevant article for the case with anonymized symbol (R03-25-A). However due to the inconsistent output of the different IR models, the number of returned documents is significantly increased (89 (223-136)). Even though the system found 8 (109-101) unique findings, degrading the value of precision causes the lower F2 values.

Based on this evaluation results, we found that there are three types of articles from the viewpoint of similarity between the queries and articles.

Simple case : Target question requires only one article that shares many terms. This type of question is easy to retrieve by any system.

Multiple case : Target question requires multiple articles. In such a case, the criteria for selecting these articles may be different. For example, one is the article that explains mutatis mutandis and the other is referred article. For the other case, one is the articles for judicial decision and the other is for analyzing the condition. Therefore, simple selection of multiple articles based on one similarity measure (e.g., BM25 score for the article database) may not be good enough to find out these documents.

Example case : It is necessary to understand the correspondence between the terms used for explaining the case with legal terminologies used in the article.

Our method shows good performance for the *simple* and *multiple cases* (HUKB2). We also found out BERT-based system may be effective for the *example case* (better recall by HUKB1), but due to the insufficient quality of our system, overall performance of HUKB1 is not good. For the next step, it is necessary to improve the performance of BERT-based system for the *example case*.

3.2 Task4

We submit three runs with different settings.

We submit following three results by using these results. Our baseline system is HUKB3 where BERT models are constructed using the method proposed in [6] and the best ensemble set is selected by using loss proposed in this paper. HUKB1 uses new augmented data and HUKB2 uses a method to select useful sentences for each article.

HUKB1 BERT model is trained with the new augmented data.

HUKB2 Select useful sentences by elastic search and use BERT model for HUKB3.

HUKB3 Baseline method

For making these training data, we split the data by considering the year. When the test data is questions for Year X (e.g., R03), questions for Year X-1 (e.g., R02) is used as an ensemble validation data set. Documents before Year X-2 (e.g., H18-R01) are randomly split for making training (90%) and validation data (10%) used for BERT fine-tuning.

Table 3 shows evaluation results of our submitted runs and best runs for each team. Our system performance is almost equivalent to the performance of the best team (KIS2). Among three different settings, HUKB1 achieves the best performance.

Table 2. Evaluation results of the submitted runs (Task4)

ID	Correct	Accuracy
KIS2	74	0.6789
HUKB1	73	0.6697
HUKB2	69	0.6330
HUKB3	69	0.6330
LLNTUdeNgram	66	0.6055
OVGU3	63	0.5780
UA_e	59	0.5413
JNLP1	58	0.5321

However, when we check the perforates of the other year dataset, our system is better in KIS works better for H30, but HUKB approach works well for R01 and R02 dataset. This means it is difficult to discusses the superiority of the method from these results.

For the comparison of HUKB1, HUKB2 and HUKB3, it is also difficult to say which method is the best. The most critical part for getting the better performance is selecting the best ensemble sets. Selection method using loss seems to be good (our system performs 2nd best group consistently), but there is no guarantee, the selected set works best for the test data. Based on the comparison of the performance of the three methods for the better loss (including best), it is difficult to say which method is the best one among them.

Table 3. Evaluation results of the submitted runs for other years(Task4)

ID	Accuracy (H30)	Accuracy (R01)	Accuracy (R02)	Macro Average (H30-R02)
LLNTUmojCraw	0.9857	1.000	1.000	0.9952
KIS2	0.7000	0.6789	0.5432	0.6407
HUKB1	0.600	0.6577	0.6173	0.625
HUKB2	0.600	0.6667	0.5926	0.6198
HUKB3	0.6286	0.6486	0.7160	0.6644
LLNTUdeNgram	0.4714	0.5405	0.5802	0.5307
UA_e	0.4571	0.6216	0.5432	0.5406
JNLP1	0.5714	0.5315	0.6173	0.5734

It is necessary to conduct detailed results to discuss the characteristics of these methods.

4 Summary

In this paper, we introduced our system for participating in Task 3 and 4 of COLIEE 2020. For the task 3, we propose to use new IR system that utilize the similarity of descriptions about judicial decision between questions and articles. This IR system can find relevant article focusing on the similarity of the decision. Merging the results of the IR system with ordinal keyword-based IR system achieves good recall without losing precision a lot. Merging the results of the BERT-based IR is good for improve recall but losing precision a lot. Our system using new IR system with ordinal keyword-based IR system achieves the best performance for the task 3. For the task4, we proposed different data augmentation method and preprocess method to select relevant part of articles. Our system generally works well, but it is difficult to say which settings is the best among them. Further analysis is necessary for understanding the characteristics of the proposed system.

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. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops). pp. 63–69 (2002)
5. Rabelo, J., Goebel, R., Kim, M.Y., Kano, Y., Yoshioka, M., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. The Review of Socionetwork Strategies (2022). https://doi.org/10.1007/s12626-022-00108-w
6. Rabelo, J., Goebel, R., Kim, M.Y., Kano, Y., Yoshioka, M., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. The Review of Socionetwork Strategies (2022). https://doi.org/10.1007/s12626-022-00108-w

7. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: A summary of the coliee 2019 competition. In: Sakamoto, M., Okazaki, N., Mineshima, K., Satoh, K. (eds.) New Frontiers in Artificial Intelligence. pp. 34–49. Springer International Publishing, Cham (2020)
8. Rabelo, J., Kim, M.Y., Goebel, R., Yoshioka, M., Kano, Y., Satoh, K.: Coliee 2020: Methods for legal document retrieval and entailment. In: New Frontiers in Artificial Intelligence. JSAI-isAI 2020. Lecture Notes in Computer Science. pp. 196–210. Springer International Publishing, Cham (2021)
9. Robertson, S.E., Walker, S.: Okapi/Keenbow at TREC-8. In: Proceedings of TREC-8. pp. 151–162 (2000)
10. 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
11. 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)
12. Yoshioka, M., Onodera, D.: A civil code article information retrieval system based on phrase alignment with article structure analysis and ensemble approach. 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. 9–22. EasyChair (2017)
13. Yoshioka, M., Suzuki, Y.: HUKB at COLIEE 2020 information retrieval task. In: The Proceedings of the 14th International Workshop on Juris-Informatics (JURISIN2020). pp. 195–208. The Japanese Society of Artificial Intelligence, (2020)

Less is Better: Constructing Legal Question Answering System by Weighing Longest Common Subsequence of Disjunctive Union Texts

Minae Lin[1*], Sieh-chuen Huang[2[0000-0003-3571-5236]**], Hsuan-lei Shao[3[0000-0002-7101-5272]**]

[1] National Taiwan University, Roosevelt Rd., Taipei City 106, Taiwan
LinMinae@gmail.com
[2] National Taiwan University, Roosevelt Rd., Taipei City 106, Taiwan
schhuang@ntu.edu.tw
[3] National Taiwan Normal University, Heping E. Rd., Taipei City 106, Taiwan
hlshao2@gmail.com

Abstract. This article is prepared for this year's *Competition on Legal Information Extraction/Entailment (COLIEE 2022)*, an international information processing and retrieval competition. The proposed method tackles on how to construct an answering system capable of responding Yes/No legal questions, ultimately recognizing entailment between legal queries from past Japanese bar exams and relevant articles of Japan Civil Code (both in Japanese). We first attempted to extract disjunctive union texts from each training query and relevant article(s) with corresponding 'Y/N' answers as their labels, eventually forming our reference database (training set). Then the same process was repeated on a sample of different queries and relevant articles yet without a 'Y/N' label as the input (testing set). Finally, when constructing our model, the similarity ratio between the test disjunctive union and the training disjunctive union by longest common subsequence was calculated as its basis. As a result, this model achieved an accuracy of 0.6055 in Task 4 (rank 3rd as a team, and 7th as a trial). This is an extremely simple and cost-saving model, yet shows a satisfactory performance.

Keywords: Textual Entailment, Disjunctive Union Texts, Longest Common Subsequence, Legal Analytics, COLIEE 2022, Lexical Features

1 Introduction

Competition on Legal Information Extraction/Entailment (COLIEE), known as a famous international information retrieval competition, had its ninth run in 2022 [1, 2, 3]. Four tasks are included in the 2022 competition: Tasks 1 and 2 are the case law competition, and tasks 3 and 4 are statute law competition. Task 1 is a legal case retrieval task, and it involves 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 is the legal case entailment task, which involves the identification of a paragraph from existing cases that entails the decision of a new case. As in previous COLIEE competitions, Task 3 is to pair a yes/no legal question Q to the most relevant statutes

(articles) from a database of Japanese Civil Code; Task 4 is to confirm entailment of a yes/no answer from the most relevant Civil Code articles provided [4].

Our team, *Legal Analytics Laboratory of National Taiwan University (LLNTU)*, participated in the Task 3 (the Statute Law Retrieval Task) and the Task 4 (the Legal Question Answering Task). In the Task 4, we constructed a dataset of disjunctive union texts from training queries and relevant articles. Next, the same process was applied to the unseen legal queries and its relevant articles. Finally, we compared the disjunctive union texts of the tested data to those in the training data, i.e., establishing a longest common subsequence similarity comparison model to find the most similar disjunctive union text in the training data. The 'Y/N' labels of the most resembling queries in the training data will be the recommended answer for the test queries. The accuracy of the model without stop words is 0.5780, while the accuracy of the model with punctuations as stop words is 0.6055 (rank 3 as a team).

2 Competition Data and Task Description

The COLIEE provided the "*Statute Law Competition Data Corpus*," in which legal questions were drawn from Japanese bar exams, and all Japanese Civil Code articles were also provided in both Japanese and English. The format of the COLIEE competition corpora has been derived from an NTCIR work [4], offering confirmed relationships between questions and the articles, as in the following example:

"R02-27-O"

```
<pair id="R02-27-O" label="N">
<t1>
Article 676 (3) A partner may not seek the division of
the partnership property before liquidation.
</t1>
<t2>
A partner may seek the division of the partnership
property before liquidation if the consent of the majority
of partners exists.
</t2>
</pair>
```

The <t2> part is a legal 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 "R02-27-O" is confirmed to be answerable from the Article 676 (3) (relevant to Task 3). The pair label "Y" in this example means the answer for this query is "Yes," which is entailed by the relevant article (relevant to Task 4). For both Task 3 and 4, the training data will remain the same. These training data were provided for two sub-tasks as follows:

1) Task 3: Legal Information Retrieval Task. Input is a bar exam 'Yes/No' question and output should be relevant Civil Code articles.

2) Task 4: Recognizing Entailment between Law Articles and Queries. Input are a bar exam question and its relevant article(s). And output should be 'Yes' or 'No' as the answer [4].

For Task 3, the evaluation measures include precision, recall and F2-measure (instead of the ordinary F1-measure). The COLIEE organizer places more emphasis on recall because the goal of information retrieval is to select articles to be used in the next entailment process [4], which is:

$$Precision = \frac{average\ of\ (the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query)}{the\ number\ of\ retrieved\ articles\ for\ each\ query}$$

$$Recall\ = \frac{average\ of\ (the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query)}{the\ number\ of\ relevant\ articles\ for\ each\ query}$$

$$F2\ measure\ = \frac{5 \times Precision \times Recall}{4 \times Precision + Recall}$$

The goal of Task 4 is to construct a 'Yes/No' question answering system for legal queries, interpreting entailment between queries and their relevant articles. How the proposed system performs will be evaluated using its 'Yes' or 'No' answers to previously unseen queries, after comparing input queries and provided Civil Code article(s). Training data consists of triplets of a query, relevant article(s), and a correct answer label in 'Y' or 'N'. Test data includes only queries and relevant articles, but no 'Y/N' label [4]. We will focus on Task 4 in the following parts of this article.

3 Data Preprocessing

Our research processed only the Japanese-language data corpus provided by the COLIEE organizer, not the English-language data corpus. At the beginning, we had 1,086 bar exam questions as <t2> from H18-R02, and 768 Civil Code articles as <t1>. Our design is to find the disjunctive union texts from training questions and relevant articles. That is to say, the machine firstly compares words of <t2> with those of <t1> and then picks up words appearing in either <t2> or <t1> as "disjunctive union texts", while discarding words existing in both <t2> and <t1>. Mathematically speaking, it is a process of finding the symmetric difference of <t2> and <t1> sets, as Fig. 1 shows.

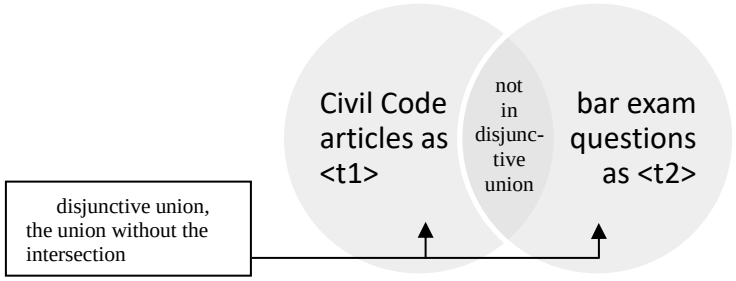


Fig. 1. Venn diagram of <t1>△<t2>

For example, as Table 1 indicates, in the training data <pair label="N" id="R02-27-O">, the query id "R02-27-O" (<t2>) is believed to be relevant with Article 676 (3) (<t1>). First, in order to reduce noises, the article number of the Civil Code, i.e., "Article 676 (3)" was excluded while constructing disjunctive union text (B). Next, the words appearing in both <t2> and <t1>, "*組合員は、清算前に組合財産の分割を求めることができ*"in Japanese and "*A partner may seek the division of the partnership property before liquidation.*" in English, will be abandoned, while words existing in either <t2> or <t1>, "*組合員の過半数の同意がある場合には，る。ない。*" in Japanese and "*if the consent of the majority of partners exists. not.*" in English will be recorded as "disjunctive union text." Please note only Japanese data is processed so the content's English translation may not be literally equivalent to its Japanese counterpart. Finally, we derived a new set of data (B) from combining "disjunctive union text" with the provided label 'N.' In the result dataset, there are a total of 1,086 (B) in number.

Table 1. Constructing disjunctive union text for <pair label="Y" id="R02-27-O">

Bar exam question for Training "R02-27-O" (<t2>)	Japan Civil Code article(s) (<t1>)	Disjunctive union text (B)	Label
組合員は，組合員の過半数の同意がある場合には，清算前に組合財産の分割を求めることができる。	~~*第六百七十六条　3*~~ *組合員は、清算前に組合財産の分割を求めることができない。*	*組合員の過半数の同意がある場合には，る。ない。*	N
A partner may seek the division of the partnership property before liquidation if the consent of the majority of partners exists.	~~*Article 676 (3)*~~ *A partner may not seek the division of the partnership property before liquidation.*	*if the consent of the majority of partners exists. not.*	N

After the above transformation, we acquired a dataset in which 1,086 disjunctive union texts and corresponding 'Y/N' labels are included, which will function as our database to work within the next step.

When handling the test data, we also preprocessed queries and relevant article(s) in the same manner as the approach described above. In other words, a disjunctive union text of the test data will also be produced, but without the corresponding 'Y/N' label. Next, our model searches the most similar entry in the dataset based on the similarity ratio calculated using longest common subsequence (LCS) between disjunctive union text of this test data and those in our database (constituted of training data) as the weights. Longest common subsequence (LCS), which represents one of the lexical features extractable from two or more sequences, can be obtained from counting the number of common subsequence appearing in all the sequences, but does not require those to be in consecutive positions.

For example, as Table 2 shows, in the test data <pair id="R03-28-U">, Article 677 was provided as the most relevant article but without a label of 'Yes/No' answer. The

disjunctive union text of this query (A) is "*組合員の持分の限度でる。ない。*"in Japanese and "*to the extent of the partner's interest. not*" in English.

Table 2. Answering Yes/No Question <pair id="R03-28-U">

Bar exam question for Testing "R03-28-U" (<t2>)	Japan Civil Code article(s) (<t1>)	Disjunctive union text (A)	Most similar datum ("R02-27-O")
組合員の債権者は、組合財産について、その組合員の持分の限度で権利を行使することができる。	~~第六百七十七条~~ *組合員の債権者は、組合財産についてその権利を行使することができない。*	*組合員の持分の限度でる。ない。*	["組合員の過半数の同意がある場合には，る。ない。", "N"]
A partner's creditor may exercise the rights of that creditor against the partnership property to the extent of the partner's interest.	~~*Article 677*~~ *A partner's creditor may not exercise the rights of that creditor against the partnership property.*	*to the extent of the partner's interest. not*	["*if the consent of the majority of partners exists. not.*", "N"]

Finally, the model found that the most similar disjunctive union text, which is the entry containing the highest ratio of LCS in our database, "R02-27-O" ["組合員の過半数の同意がある場合には，る。ない。","N"] in Japanese and ["*if the consent of the majority of partners exists. not.*", "N"] in English. Therefore, our system will output the response to this query as "N," brought by the most similar data text contained in "R02-27-O" as indicated in Table 1.

The degrees of similarity between querying disjunctive union text (i.e., (A) in the Table 2) and those texts in our database (for example, (B) in the Table 1) are calculated using the Sequence Matcher Similarity ratios (hereinafter referred to as SMS-ratios) of longest common subsequence intersecting both. Figure 2 visualizes the core concept when performing the LCS comparison.

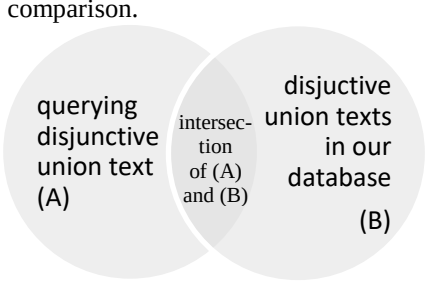


Fig. 2. Intersection of (A) and (B)

The following formula shows how to calculate SMS ratio of (A) and (B).

$$\text{Sequence Matcher Similaraty (SMS) ratio} = \frac{2 \times amount\ of\ A \cap B}{amount\ of\ A + amount\ of\ B}$$

4 The Result

To improve the accuracy, we attempted to adjust the disjunctive union texts detection mechanism to execute with or without stop words setup. In our case, punctuations are chosen to be stop words. We used the training data of H18-H29 to construct the database of disjunctive union texts. And the data of H30-R03 are utilized as the test set (self-testing sets). Table 3 shows both scenarios' performances in these four self-testing sets respectively.

Table 3. Accuracy in self-testing sets with two different constructions of the model

model	H30-accuracy	H30-performance	R01-accuracy	R01-performance	R02-accuracy	R02-performance	R03-accuracy	R03-performance
Without stop words (diffSim)	51.4%	36/70	50.5%	56/111	60.5%	49/81	57.8%	63/109
With stop words (deNgram)	47.1%	33/70	54.1%	60/111	58%	47/81	60.1%	66/109

In terms of performance, we achieve 0.5780 accuracy for the model without defining any stop words, and 0.6055 accuracy with the punctuations defined as stop words regarding the official test data (R03). Overall, it is ranked as 7th and 8th among 17 teams/trials. The performance of this non-deep-learning model shows the potential of employing a simple method on the task of answering legal questions.

5 Discussion and Conclusion

Our model is quite simple yet useful in interpreting most of the queries. The symmetric difference of a query and its relevant article(s) may be correlated with similar patterns extracted from the 'Yes/No' question triplets and hence those corresponding answers ('Y/N' label) associated with the specified pattern can be used to answer unseen queries. In concise, what this article proposed is an analogical approach relying on detecting resembling symmetric differences between analyzed queries and unprocessed ones presented in the input question. This approach can correctly identify responses to not only short queries but also long queries containing long texts. Table 4 demonstrates how our model could be applied to a lengthy question. By calculating the ratio of SMS (LCS) and identifying the nearest triplet, an answer ("N" in this cases) of acceptable accuracy should be obtained.

Table 4. Calculating SMS ratio to solve a 'Yes/No' question with long text <pair id="R03-02-I">

Bar exam question "R03-02-I" (<t2>)	Japan Civil Code article(s) (<t1>)	Disjunctive union text (A)	Most similar datum (similarity=0.654) ("H20-30-I")
表意者の意思表示がその真意ではないことを理由として無効とされた場合において、その無効は、善意であるが過失がある第三者に対抗することができる。	~~第九十三条~~ 意思表示は、表意者がその真意ではないことを知ってしたときであっても、そのためにその効力を妨げられない。ただし、相手方がその意思表示が表意者の真意ではないことを知り、又は知ることができたときは、その意思表示は、無効とする。 2 前項ただし書の規定による意思表示の無効は、善意の第三者に対抗することができない。	の理由としてとされた場合において、そであるが過失がある。は、知ってしたときであっても、そのためにその効力を妨げられない。ただし、相手方がその意思表示が表意者の真意ではないことを知り、又は知ることができたときは、その意思表示は、する。 2 前項ただし書の規定によるのない。	[“心裡留保の場合，相手方がをらなかったとし，知らないことについて重大な過失がなければ，有であ 意思表示は、がそではないことをってしたときであっ、そのためにその効力を妨げられない。ただし、相手方が表意者の真意を知り、又は知ることができたときは、無とす”, “N”]
If the manifestation of intention of a person is held void on the ground that it does not reflect that person’s true intention, such nullity may be duly asserted against a third party in good faith acting with negligence.	*~~Article 93~~ (1) The validity of a manifestation of intention is not impaired even if the person making it does so while knowing that it does not reflect that person's true intention;* *…* *(2) The nullity of a manifestation of intention under the provisions of the proviso to the preceding paragraph may not be duly asserted against a third party in good faith.*	*If the of a person is held void on the ground, such nullity may be duly asserted against The validity of a is not impaired even if the person making it does so while knowing…The nullity of a manifestation of intention under the provisions of the proviso to the preceding paragraph may not be*	[“*In the case of concealment of, assuming was unaware of party making the manifestation, and there was no gross negligence regarding that lack of knowledge,…knew or could have known that the manifestation was not person who made it, that is void.*”, “N”]

Bar exam question "R03-02-I" (<t2>)	Japan Civil Code article(s) (<t1>)	Disjunctive union text (A)	Second similar datum (similarity=0.562) ("H26-2-A")
表意者の意思表示がその真意ではないことを理由として無効とされた場合において、その無効は、善意であるが過失がある第三者に対抗することができる。	~~第九十三条~~ 意思表示は、表意者がその真意ではないことを知ってしたときであっても、そのためにその効力を妨げられない。ただし、相手方がその意思表示が表意者の真意ではないことを知り、又は知ることができたときは、その意思表示は、無効とする。 ２ 前項ただし書の規定による意思表示の無効は、善意の第三者に対抗することができない。	の理由としてとされた場合において、そであるが過失がある。は、知ってしたときであっても、そのためにその効力を妨げられない。ただし、相手方がその意思表示が表意者の真意ではないことを知り、又は知ることができたときは、その意思表示は、する。 ２ 前項ただし書の規定によるのない。	[“前二項規定よよし善か過失抗るな。相手方すにいてがを行った場合おて相手方そ事実を知ってたきはそ意思表示をりすとるが第三者が強迫を行った場合において相手方がその事実を知らかったときもその思表示を取り消ことができ。”, “Y”]
If the manifestation of intention of a person is held void on the ground that it does not reflect that person’s true intention, such nullity may be duly asserted against a third party in good faith acting with negligence.	~~*Article 93*~~ *(1) The validity of a manifestation of intention is not impaired even if the person making it does so while knowing that it does not reflect that person's true intention; …* *(2) The nullity of a manifestation of intention under the provisions of the proviso to the preceding paragraph may not be duly asserted against a third party in good faith.*	*If the of a person is held void on the ground, such nullity may be duly asserted against The validity of a is not impaired even if the person making it does so while knowing…The nullity of a manifestation of intention under the provisions of the proviso to the preceding paragraph may not be*	[“A manifestation of intention based on fraud or duress is voidable. If a first party In cases any person a second that is voidable only second or could have known that The rescission of induced by fraud under provisions preceding two paragraphs may be duly asserted against a third party in good faith acting without negligence.. the other such may be rescinded other such On the other hand, in cases any third party commits any duress inducing any person to make to other party, such manifestation other party did know such fact.”, “Y”]

Table 5 shows how to generate disjunctive union text of "H20-30-I" (B).

Table 5. Constructing disjunctive union text of <pair label="N" id="H20-30-I">

Bar exam questions for training "H20-30-I" (<t2>)	Japan Civil Code article(s) (<t1>)	Disjunctive union text (B)	Label
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。	~~第九十三条~~　意思表示は、表意者がその真意ではないことを知ってしたときであっても、そのためにその効力を妨げられない。ただし、相手方が表意者の真意を知り、又は知ることができたときは、その意思表示は、無効とする。	心裡留保の場合，相手方がをらなかったとし，知らないことについて重大な過失がなければ，有であ 意思表示は、がそではないことをってしたときであっ、そのためにその効力を妨げられない。ただし、相手方が表意者の真意を知り、又は知ることができたときは、無とす	N
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.	~~*Article 93*~~ ~~*(1)*~~ *The validity of a manifestation of intention is not impaired even if the person making it does so while knowing that it does not reflect that person's true intention; provided, however, that if the other party knew or could have known that the manifestation was not the true intention of the person who made it, that manifestation of intention is void.*	*In the case of concealment of, assuming was unaware of party making the manifestation, and there was no gross negligence regarding that lack of knowledge, that…knew or could have known that the manifestation was not person who made it, that is void.*	N

As Table 4 shows, the SMS similarity ratio of the most similar pair, (A) "R03-02-I" and (B) "H20-30-I," was 0.654, which was determined by LCS. Compared to that, the ratio of the second most similar pair, (A) "R03-02-I" and (B) "H26-2-A," was merely 0.562. The rest 1,084 entries of (B) in our database had lower ratios when compared with the above two. Therefore, (B) of "H20-30-I" was determined to be the most similar datum.

The disjunctive union text of the unseen query (A) may contain words similar to the instances of (B) in the database including but not limited to "意思表示 (manifestation of intention)", "表意者(person who made it)", and "真意 (true intention)." This approach naturally leads the model to pair the data with the same topic, "心裡留保 (concealment of intention)." However, the number of the Civil Code article, for example, "Article 93" in Table 4, was not included in the database while constructing disjunctive union text (B).

Therefore, in the case that a relevant article of the unseen query shares several words with a disjunctive union text in the database, our model may be misled. This happens in the "R03-08-E" (which is relevant to Article 183), in which our model incorrectly pairs it with the disjunctive union text of "H24-10-5" (which is more relevant to Article 184 and the label is "N"). We assume the reason is that Article 183 is very similar to Article 184 but our model fails to distinguish them due to the lack of information such as the number of the article.

To use disjunctive texts to understand semantics is not a brand-new idea [7, 8]. But it quickly reaches its limit as semantics become more complex. Also, the calculation of disjunctive union can incur intensive computation costs when processing lengthy and complicated text [9]. On the other hand, longest common subsequence (LCS) had been used by past participants [10, 11] as an approach to performing lexical matching in the previous COLIEE for Task 1 (legal case retrieval task). LCS method is also a traditional method and faces its limit [12, 13].

The idea of combining disjunctive union and LCS stems from the experience of one of the authors when preparing for the bar exam. The disjunctive union text, which is the non-overlapping parts of a query and its relevant article, demonstrates a pattern of the question maker's attempts to "camouflage" examinees. That is, it entails a designated set of rules to incorporate statutes with the exam questions. Hence, it is not the intersection of a query and its relevant article that is important, but the symmetric difference between those two matters the most. Our model compares the patterns of camouflage detected in previous exams with similar patterns found in unseen questions by measuring their longest common subsequence (LSC). This model deals merely with lexical features without semantics. Therefore, we cannot say that our model correctly answers the question. Perhaps it produces closely estimated answers by catching the question makers' biases. For example, in the case of human observation, when the word , "不動産 (real property)" in the statutes was replaced by "動産 (movables)" in the query, the label (answer) is usually "No" since they cannot be used interchangeably. On the contrary, when the word, "物 (thing)" in the statutes was replaced by "動産 (movables)" in the query, the label (answer) is usually "Yes" because the former is a broader concept that can be narrowed down to the latter.

Our experiment showed that a purely rule-based approach may still prove to be a useful solution, although this is limited to a very specific kind of corpuses. Overall,

when compared to other models applying complicated algorithms such as deep learning technique (which has been used by our LLNTU team in previous year, mainly based on the BERT [5, 6]), the model we proposed this time is simple enough to run without computational burden, which takes less than 1 second for preprocessing the training data and 8.14 seconds for processing the test data and doing the LCS comparison. Compared to this, the BERT-based model we introduced previously took several days to reach the outcome [14]. This is because the time complexity of preprocessing is $O(m^2)$ per pair with m as lengths of queries, and the time complexity of LCS detection is $O(n \cdot m^2)$ per query which n accounts for the number of entries in the training data. We recognize this simple model has its limits. To increase the model's performance, it may be necessary to do failure analysis more precisely and add additional rules when constructing the model. That will be our main goal next year.

Acknowledgment

Hsuan-lei Shao, "Knowledge Graph of China Studies: Knowledge Extraction, Graph Database, Knowledge Generation" (110-2628-H-003-002-MY4, *Ministry of Science and Technology, the MOST*), Taiwan.

Sieh-chuen Huang, "A Study on Property Management Regimes in Family and Succession Law in Taiwan" (110-2410-H-002-026-MY3, *Ministry of Science and Technology, the MOST*), Taiwan.

References

1. Rabelo, J., Kim, M. Y., Goebel, R., Yoshioka, M., Kano, Y., & Satoh, K. (2020). A Summary of the COLIEE 2019 Competition. In: Sakamoto, M., Okazaki, N., Mineshima, K., Satoh, K. (eds) New Frontiers in Artificial Intelligence. JSAI-isAI 2019. Lecture Notes in Computer Science, vol 12331. Springer, Cham. https://doi.org/10.1007/978-3-030-58790-1_3
2. Rabelo, J., Kim, M. Y., Goebel, R., Yoshioka, M., Kano, Y., & Satoh, K. (2020). COLIEE 2020: methods for legal document retrieval and entailment. https://sites.ualberta.ca/~rabelo/COLIEE2021/COLIEE_2020_summary.pdf
3. Rabelo, J., Goebel, R., Kim, M. Y., Kano, Y., Yoshioka, M., & Satoh, K. (2022). Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. The Review of Socionetwork Strategies 16, 111-133. https://doi.org/10.1007/s12626-022-00105-z
4. COLIEE organizer, COLIEE-2022 main webpage, https://sites.ualberta.ca/~rabelo/COLIEE2022/
5. Tohoku NLP Group. 2021. Pretrained Japanese BERT models (BERT-base_mecab-ipadic-char-4k_whole-word-mask). https://github.com/cl-tohoku/bert-japanese. (2022).
6. Alinear-corp, albert-japanese, https://github.com/alinear-corp/albert-japanese.
7. Kasper, R. T. (1986, July). A logical semantics for feature structures. In 24th Annual Meeting of the Association for Computational Linguistics (pp. 257-266).
8. Eiter, T., Gottlob, G., & Mannila, H. (1997). Disjunctive datalog. ACM Transactions on Database Systems (TODS), 22(3), 364-418.

9. Eiter, T., & Gottlob, G. (1995). On the computational cost of disjunctive logic programming: Propositional case. Annals of Mathematics and Artificial Intelligence, 15(3), 289-323.
10. Tran, V., Le Nguyen, M., Tojo, S., & Satoh, K. (2020). Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. Artificial Intelligence and Law, 28(4), 441-467.
11. Tran, V., Nguyen, M., & Satoh, K. (2020). Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. ICAIL '19: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, June 2019. https://doi.org/10.1145/3322640.3326740
12. Anan, Y., Hatano, K., Bannai, H., Takeda, M., & Satoh, K. (2012). Polyphonic Music Classification on Symbolic Data Using Dissimilarity Functions. In ISMIR (pp. 229-234).
13. Deken, J. G. (1979). Some limit results for longest common subsequences. Discrete Mathematics, 26(1), 17-31.
14. Shao, HL., Chen, YC., Huang, SC. (2021). BERT-Based Ensemble Model for Statute Law Retrieval and Legal Information Entailment. In: Okazaki, N., Yada, K., Satoh, K., Mineshima, K. (eds) New Frontiers in Artificial Intelligence. JSAI-isAI 2020. Lecture Notes in Computer Science, vol 12758. Springer, Cham. https://doi.org/10.1007/978-3-030-79942-7_15

SIAT@COLIEE-2022: Legal Case Retrieval with Longformer-based Contrastive Learning *

Jiabao Wen[1,2,3], Zeyi Zhong[1,3], Yuelin Bai[1,2,3], Xiaoyan Zhao[1,2,3], and Min Yang[1,3**]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] SIAT-DELI Artificial Intelligence and Law Lab
{jb.wen,zhongzy,yl.bai,xy.zhao,min.yang}@siat.ac.cn

Abstract. In the 2022 Competition on Legal Information Extraction and Entailment (COLIEE-2022), we participate in Task 1 (i.e., the legal case retrieval task), which aims at extracting supporting legal cases from the pre-constructed repository for a specific query case automatically. Pre-trained language models (e.g., BERT) have brought substantial improvement to the information retrieval systems. However, the length of the case law documents usually exceeds the input length of BERT, which limits transferring the advantages of language model to case law retrieval. In this paper, we introduce a longformer-based contrastive learning that advances the modeling of semantic retrieval features, making it easy to process sequences of thousands of tokens. In addition to neural-based methods, we also explore competitive traditional retrieval models in the legal case retrieval task. Our team won the second place among all teams participating in the COLIEE-2022 Task 1. Experimental results demonstrate that our proposed methods significantly outperform the other methods by exploring pre-trained language models and contrastive learning.

Keywords: Legal case retrieval · Long-document transformers · Contrastive learning.

1 Introduction

The law plays an important role in our daily lives and constitutes a primary information resource. Legal case retrieval, which aims at retrieving relevant supporting cases from a pre-constructed repository for a specific query case, has attracted considerable attention of judges and lawyers.

* This work was partially supported by the Natural Science Foundation of Guangdong Province of China (No. 2019A1515011705), Youth Innovation Promotion Association of CAS China (No. 2020357), Shenzhen Science and Technology Innovation Program (Grant No. KQTD20190929172835662), Shenzhen Basic Research Foundation (No. JCYJ20210324115614039).

** Corresponding author.

Conventional legal case retrieval methods usually employ keyword matching to search a large-scale repository for retrieving potentially supporting legal cases [3, 7]. However, the keyword-based retrieval methods in practice have several limitations. First, the list of keywords that may appear in the supporting cases should be determined in advance. The process of defining keyword list needs to be refined through repeated testing, which is time-consuming and labor-intensive. In addition, keyword search aims to exactly matching the terms specified in the documents, which may miss many potentially relevant cases. Subsequently, deep learning has been proposed to learn semantic representations of documents for accurate retrieval [9, 4]. The superior performance of deep learning based retrieval relies heavily on a large-scale labeled training data. Nevertheless, it is expensive to construct a sufficient labeled training corpus as the legal case annotation requires expert knowledge.

In parallel, legal case retrieval has been operationalized in the competition on Legal Information Extraction and Entailment (COLIEE) since 2014. The top performers at COLIEE achieve noticeable progress in legal case retrieval. In COLIEE-2019, the JNLP team [10], which was the winner of the COLIEE-2019 Task 1, proposed to take the advantage of both conventional lexical matching and deep neural networks for legal case retrieval. In COLIEE-2020, the THUIR team [8] proposed the exact string matching and semantic understanding by using word-entity duet framework and BERT-PLI model for legal case retrieval. In COLIEE-2021, the TLIR team [6] won the first place in Task 1, which compared both neural as well as traditional methods and demonstrated the competitive performance of traditional language models for legal case retrieval. The DoSSIER team [6] leveraged the dense retrieval and summarization-based re-ranking for handling long documents.

In this paper, we employ the long-document transformer (Longformer) [1] to effectively process long legal documents, which combines the local-windowed self-attention and the task motivated global attention. To further improve the generalization of the legal retrieval model, we also use contrastive learning to learn more robust document representations, where the semantically similar samples are pulled closely and the dissimilar samples are pushed apart. The experimental result is reported in the first run for Task 1. In addition, in the second run, we explore the effectiveness of the language model for information retrieval (LMIR) [5] in Task 1. Experimental results demonstrate that our neural method achieves good generalization performance for legal case retrieval, and our SIAT team won the second place among all teams participating in the COLIEE-2022 Task 1.

2 Task Description

2.1 COLIEE-2022 Task 1: Legal Case Retrieval

The goal of the COLIEE-2022 Task 1 (i.e., legal case retrieval) is to retrieve relevant supporting cases from a pre-constructed repository for each query case. Formally, given a query case q, the legal case retrieval aims to retrieve relevant cases (or supporting cases) from a corpus $D = \{d_1, d_2, \ldots, d_n\}$ with n samples.

Here, the "supporting case" (or "noticed case") is a legal term denoting the precedent in a pre-constructed repository, which may provide supporting views for the new query cases.

2.2 Dataset

To evaluate the effectiveness of the legal case retrieval models, an official experimental corpus for COLIEE-2022 Task 1 is given, which is related to a database of predominantly Federal Court of Canada case laws, provided by Compass Law [6]. The statistics of the corpus are reported in Table 1. In particular, the corpus contains 900/300 query cases for training/testing. The repository in the training phase contains 4415 cases, while there are 1563 cases in the repository for testing. The average number of noticed cases in the training set for each query case is 4.67. The teams participating in the COLIEE-2022 Task 1 are required to submit the retrieval cases of the testing query cases for the online evaluation purpose.

Table 1. The statistics of the dataset for the COLIEE-2022 Task 1.

Task 1	**Train**	**Test**
# Queries	900	300
# Candidate cases	4415	1563
# Noticed cases per query	4.67	-

2.3 Evaluation Measures

The official evaluation metrics for the COLIEE-2022 Task 1 include micro-average precision, recall, and F1-measure, which are formulated as follows:

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{1}$$

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

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{3}$$

where N_{TP} denotes the number of correctly predicted queries, $N_{TP} + N_{FP}$ is the number of retrieved legal cases for each query case, and $N_{TP} + N_{FN}$ is the number of noticed legal cases for each query case.

3 Methods

3.1 Run 1 & Run 3: Longformer with Contrastive Learning

The task of legal case retrieval faces a great challenge that both query and candidate cases involve extreme long texts. The applications of pre-trained language

models (PLMs) to ad-hoc retrieval have achieved impressive performance due to their ability in capturing rich language knowledge from large-scale corpora. Most of these methods are based on text matching function learning, which model inputs as query-passage pairs and capture the interactions between the query and candidate documents directly. Considering the extended length of legal cases and the efficiency of representation matching, we choose to apply the methods based on representation learning, utilizing the pre-trained language models to encode the query cases and the candidate cases separately. Then, their relevance is computed by applying a similarity function over the document representations.

On the other hand, the contrastive learning [2] produces superior and discriminative document representations. Contrastive learning can be combined with various PLMs and achieves competitive performance in many NLP tasks. Different from many PLMs which have limited ability to process long documents, Longformer [1] is proposed to effectively encode the long documents with thousands of tokens by combining both local and global attention mechanisms. In this paper, the results of Run 1 and Run 3 are mainly based on Longformer with contrastive learning (Longormer-CL), which achieves our best score in this competition.

Data Pre-Processing The experimental data is drawn from Federal Court of Canada case laws. Based on our observation, some of the original cases quote other cases, where the cited cases are replaced by placeholders such as "FRAGMANT_SUPPRESSED" in the released dataset. In this paper, we remove these placeholders to reduce unnecessary and interferential information. In addition, some of the case documents contain French text. Although most of these French words are from the translation of English text in the same document, there exist some documents that only contain French text. To keep overall information, we translate all of the French text into English through Google Translate. Then, we convert all English letters to lowercase and remove stopwords as well as punctuation.

Longformer with Contrastive Learning Contrastive learning aims to learn effective document representations by pulling semantically similar documents together and pushing apart the dissimilar documents. One critical problem that may greatly influence the retrieval performance is how to construct positive and negative samples for a given case. We directly take the cases that are labeled as similar to the query as positive samples and randomly choose the cases from the remaining documents as negative samples. Note that we also tried to replace the random negative sampling with hard negative sampling. The hard negative samples are not labeled as similar for each query case but retrieved by a primary re-ranker. However, the contrastive learning with hard negative sampling performs worse than the random negative sampling. Therefore, in this competition, we utilize random negatives in the training procedure.

The overall model structure is shown in Figure 1. The Longformer is employed to produce the query and candidate case representations separately by obtaining

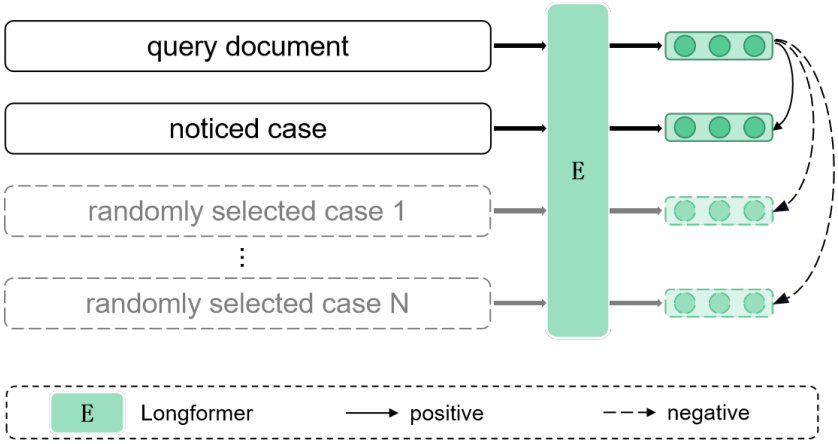


Fig. 1. The overall model structure.

the output hidden vector at the first token position. As a result, the output feature representation is denoted as $\mathbf{h} \in \mathbb{R}^{d_h}$ for each case where d_h is the hidden dimension. Then, we calculate the cosine similarity score of each two document representations for legal case retrieval. The contrastive loss function is defined as follows:

$$\mathcal{L}_{\text{CL}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{sim(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}(e^{sim(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{sim(\mathbf{h}_i,\mathbf{h}_j^-)/\tau})} \tag{4}$$

where τ is a temperature hyperparameter. $sim(\cdot)$ denotes the cosine similarity function. $\mathbf{h}_i$ denotes the representations of the query case x_i. $\mathbf{h}_i^+$ and $\mathbf{h}_i^-$ denote the representations of positive and negative samples for the query case x_i. M is the number of query cases in the training set. N is the number of negative samples in the current mini-batch. The parameters of the Longformer are optimized through back propagation.

Metrics-based Adaptive Retrieval Generally, we merge the recall and ranking stages into one phase according to the similarity scores between the query and candidate documents. In particular, we retrieve those cases with scores above a certain threshold and rank them accordingly. We notice that the query cases have different numbers of noticed cases, thus simply setting a fixed threshold is not optimal. To achieve an optimal threshold, we propose an adaptive retrieval strategy based on F1 score. The intuition behind this strategy is based on our assumption that the training and testing sets have same task distribution. As illustrated in Figure 2, we enumerate different threshold scores between 0.5 to 0.9 with a step of 0.01 and calculate corresponding F1 scores on the development dataset. Then, we take the value where the model gets highest F1 as our

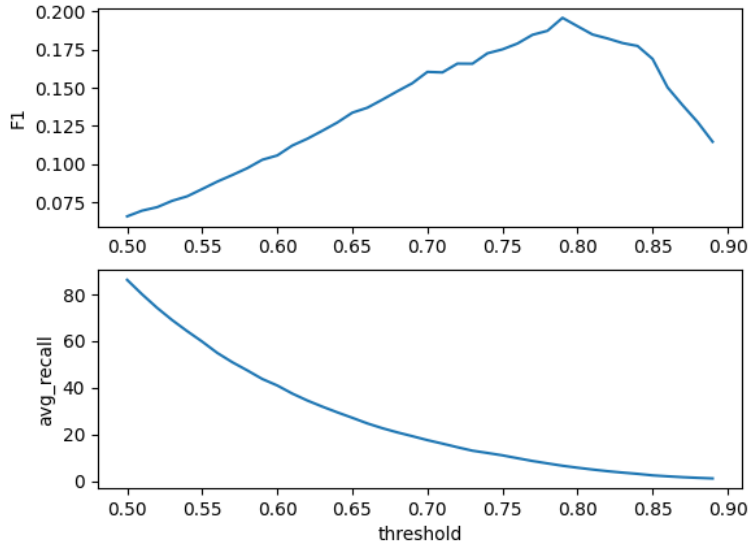


Fig. 2. The F1 and average recall of different threshold values on the valid set

threshold. Finally, in inference stage, we choose the threshold where the model retrieves the same average number of cases as on the development dataset.

Date-oriented Case Processing It is worth noticing that the supporting cases of a query case are usually time-stamped, which means that the noticed cases should appear before the query case itself. To this end, we extract the time information of each case and store them in chronological order in advance. Since there can be misleading time information such as previous judgement date, we merely extract the maximum date information.

Model Ensemble To further improve the robustness and accuracy of our legal case retrieval model, we apply k-fold cross validation and perform model ensembling. Specifically, we split the training data evenly into ten partitions, and take each one of them as valid set with the remaining as training data to train ten base models. After that we calculate their best F1 score. We define the weight for the i-th model as follows:

$$w_i = \frac{(F1)_i}{\sum_{j=1}^{10}(F1)_j} \tag{5}$$

where w_i is the weight of the i-th model. $(F1)_j$ denotes the F1 score on the i-th validation set. To aggregate the models, we calculate final similarity score

of each query-case pair (q, d) as the weighted sum of the results of individual models, which is defined as:

$$similarity(q, d) = \sum_{i=1}^{10} w_i * \cos\left(vec_q^i, vec_d^i\right) \tag{6}$$

where vec_q^i and vec_d^i denotes the query embedding and document embedding produced by the i-th model.

3.2 Run 2: Language Model for Information Retrieval

The TLIR team, which is the winner of COLIEE-2021 Task 1, utilized the language model for information retrieval (LMIR) [5] to get their best score. LMIR proves that traditional retrieval models have competitive results in legal case retrieval. Therefore, we adopt the LMIR as the retrieval model for our second run. LMIR is based on probabilistic language modeling. It is non-parametric and integrates document indexing as well as retrieval into one model. It significantly outperforms other traditional retrieval models such as TF-IDF.

LMIR estimates the probability of generating the query q and models the document d as a distribution over words. Then, the candidate documents will be ranked according to these generative probabilities. The likelihood ranking function of the query q is defined as:

$$Score(q, d) = \prod_{w \in q} P(w|d) \tag{7}$$

where w is the token in the query q and $P(w|d)$ represents the probability of token w given the document d. Then, the maximum likelihood estimates the probability of token w as follows:

$$P_{\text{mle}}(w|d) = \frac{c(w; d)}{\sum_{w \in d} c(w; d)} \tag{8}$$

where $c(w; d)$ is the frequency of token w in document d. However, there are several limitations with this simple estimation. It assigns zero probability to the words that are not observed in d and has high variance. Therefore, we choose to adapt Jelinik-Mercer Smoothing for estimating $P(w|D)$, which is defined as:

$$P_{\text{JMS}} = \lambda P_{\text{mle}}(w|D) + (1 - \lambda)P(w|C) \tag{9}$$

where $P_{\text{mle}}(w|d)$ is defined in Equation (8). $P(w|C)$ denotes the possibility of the token w over the whole training corpus C. λ is a adjustable parameter that controls the degree of smoothing.

We apply the same data pre-processing strategy as in Run 1 and Run 3. We tune the value of λ on the validation set to get the best performance. We also utilize the date-oriented case processing strategy described before to deal with the legal cases.

Table 2. Hyperparameter configurations of the experiments with Longformer.

Hyperparameter	Value
Batch size	8
Training epochs	20
Max sequence length	3072
Learning rate	2e-5

4 Experiments and Results

4.1 Experimental Settings

We conduct experiments on several models including unsupervised methods and Longformer based methods, therefore we apply different data split settings for them. For unsupervised methods, we evaluate results directly on the whole data set. While for Longformer based methods, we split the original training data into two partitions, with 10% of all query cases as the development set and the remaining data as the new training set. A key hyperparameter that needs to be determined is K, the number of retrieved cases per query. The other hyperparameters of our methods are described as follows:

- **LMIR** is adopted in our Run 2. We set hyperparameter λ to be 0.05. $P(w|C)$ plays an important role in determining the document relevance. We set K to be 6 in this experiment.
- **Longformer-CL** is used as the base encoder to produce document representations. We limit the max sequence length of the input document to 3072 so as to reduce the training consumption. The model is trained with an Adam oprimizer. The hyperparameter configurations are listed in Table 2. Longformer-CL is trained under a contrastive learning framework and produces results for our Run 1 and Run 3. We set K to be 6-7 in this experiment, and we adopt model ensemble for the final prediction on the test set.

4.2 Experimental Results

We submit three runs on the testing stage. Table 3 shows top-10 runs in COLIEE-2022 Task 1 (i.e., the legal case retrieval task) on the testing evaluation. The testing results are ranked according to the F1 score, and our results rank third (Run 1), fourth (Run 3) and sixth (Run 2) respectively. Among our submitted results ("SIAT"), Run 1, which leverages the Longformer with contrastive learning as the document encoder, achieves the best results and outperforms the LMIR-based method (our Run 2). The results suggest that the pre-trained language models and contrastive learning show strong capability in the legal case retrieval.

The top-2 runs are both from the team "UA". It is worth mentioning that the results of our runs 1&3 are only slightly lower than the team UA. We observe that our method achieves the highest recall score among all the submissions with 0.4782, which is 0.139 higher than the team UA.

Table 3. Top-10 results on COLIEE-2022 Task 1 test set.

Team ID	File	Precision	Recall	F1-Score	Rank
UA	pp_0.65_10_3	0.4111	0.3389	0.3715	1
UA	pp_0.7_9_2	0.4967	0.2961	0.3710	2
SIAT	siatrun1	0.3005	0.4782	**0.3691**	3
SIAT	siatrun3	0.3026	0.4695	0.3680	4
UA	pp_0.65_6	0.3630	0.3492	0.3559	5
SIAT	siatrun2	0.2522	0.3595	0.2964	6
LeiBi	run_bm25	0.3000	0.2850	0.2923	7
LeiBi	run_weighting	0.2687	0.3191	0.2917	8
JNLP	run3	0.3211	0.2502	0.2813	9
nigram	bm25P3M3	0.2587	0.3072	0.2809	10

5 Conclusion and Future Work

In this paper, we introduced our proposed methods for legal case retrieval task in COLIEE-2022. We employed the Longformer to effectively process long legal documents and leveraged contrastive learning to improve the generalization of the legal case retrieval. In addition to a neural-based method, we also explored the effectiveness of language model for information retrieval (LMIR). Experimental results demonstrated that our neural method achieved good generalization performance for legal case retrieval, and our team SIAT won the second place among all teams participating in the COLIEE-2022 Task 1.

In the future, we would like to explore powerful prompting techniques to exploit large-scale pre-trained language models. In addition, we also plan to pre-train the language models on a large amount of legal documents in a self-supervised manner.

References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
2. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
3. Jackson, P., Al-Kofahi, K., Tyrrell, A., Vachher, A.: Information extraction from case law and retrieval of prior cases. Artificial Intelligence **150**(1-2), 239–290 (2003)
4. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R.: Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(4), 694–707 (2016)
5. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: ACM SIGIR Forum. vol. 51, pp. 202–208. ACM New York, NY, USA (2017)
6. Rabelo, J., Goebel, R., Kim, M.Y., Kano, Y., Yoshioka, M., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. The Review of Socionetwork Strategies pp. 1–23 (2022)

7. Saravanan, M., Ravindran, B., Raman, S.: Improving legal information retrieval using an ontological framework. Artificial Intelligence and Law **17**(2), 101–124 (2009)
8. 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 (2020)
9. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management. pp. 101–110 (2014)
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)

JNLP team: Deep Learning Approaches for Tackling Legal Challenges in COLIEE 2022

Minh-Quan Bui, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen and Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, Minh Le Nguyen

Japan Advanced Institute of Science and Technology (JAIST) {quanbui, chau.nguyen, truongdo, lnkhang, ndhien, trangttn,phuongnm,nguyenml}@jaist.ac.jp

Abstract. Competition on Legal Information Extraction/Entailment (COLIEE) is an annual competition associated with the International Workshop in Juris-Informatics. The challenge for this competition is not only required the skills in processing long documents but also the ability to resolve ambiguity in the legal domain. For tackling these problems, we have applied several state-of-the-art methods and powerful pre-trained models. The results reflect the difficulty level and competitiveness in this competition.

Keywords: DL , Legal Text Processing, JNLP Team

1 Introduction

The competition is built around the goal of developing machine learning and other methods for the law domain. The applications are built on but this method will effectively support lawyers to reduce the time and effort of searching for necessary information for a certain trial. There are two types of data in COLIEE competition:

1. Case law: a database of predominantly Federal Court of Canada case laws, provided by Compass Law.
2. Statue law: answering yes/no questions from Japanese legal bar examinations.

and 4 tasks:

1. Task 1 (The Legal Case Retrieval Task): For a given case law and corpus, the machine learning approaches need to identify which cases related to or support the given case law in the corpus which contains a massive number of documents.
2. Task 2 (The Legal Case Entailment task): For the given decision and a set of candidate paragraphs, the same as task 1 competitor need to find a method to identify the relevant paragraph correctly.

3. Task 3 (The Statute Law Retrieval Task): In this task, the competitors' systems are required to retrieve an appropriate subset $(S_1, S_2, ..., S_n)$ of Japanese Civil Code Articles from the Civil Code texts dataset, used for answering a Japanese legal bar exam question Q.
4. Task 4 (The Legal Textual Entailment Data Corpus): The competitors' systems are required to determine textual entailment relationships between a given problem sentence and relevant article sentences. The systems must answer "yes" or "no" regarding the given problem sentences and given article sentences.

Based on our analysis, we have proposed some methods using deep learning (DL) approaches. Our proposals can be applied to practical applications or research for resolving problems in legal domain.

2 Related Works

2.1 Case Law

In previous years, we can see that lexical matching is the selection the teams always want to use. For examples:

- UBIRLED team extracted keywords by using Python Keyphrase Extraction Toolkit [1] and then applied K-NN algorithm combined with TF-IDF for ranking candidates for COLIEE task2 2018.
- iiest team used BM25 algorithm with configured score function for enhancing the ability to capture lexical information between candidate paragraphs and the base judgments in task 2 2020.
- UB team used learning to rank approach based on BM25 and TF-IDF algorithm in task 1 2020.

Because of the complexity of the legal domain, using only lexical matching can not achieve state-of-the-art results, especially in competitive competition such as COLIEE. Complex problems require even more complex and creative methods. For the exact purpose of demonstrating this statement, we can take a look at task the winner's method such as:

- JNLP teams used not only lexical features but also latent features obtained by a summary from several encoders. This approach significantly improves the performance and helps them achieve state-of-the-art research with nearly 20%(F score) away from the 2nd team in task 1 2018.
- Cyber team useful used universal sentence encoder, and SVM model on the output representation of query and candidate in TF-IDF space. This delicate combination makes them win the first prize in task 1 2020.
- Also in 2020, JNLP had a very creative approach that automatically generates silver data from training data provided by the organizer. With this approach, the amount of data was increased to hundreds of thousands. Its effectiveness is demonstrated by the fact that JNLP won first place in task2 2020.

[1] https://github.com/boudinfl/pke

2.2 Statue Law

For the statute law information retrieval task (Task 3), previous years, all the participants made use of pre-trained language models in their submissions. Team HUKB [19] built a BERT-based retrieval system along with Indri [16] with two different ways of forming the content of an article. JNLP [11] also built BERT-based retrieval system, they further employed a sliding window and a self-labeled technique to deal with the challenge of long articles. OvGU [18] utilized both BERT-based retrieval system (specifically, using sentence-BERT) and TF-IDF with the metadata-enriched articles and external data from the web. TR [14] employed Word Mover's Distance [9] and BERT based on the spaCy large language model. UA [8] utilized not only BERT but also BM25 and TF-IDF for different submissions.

For the statue law textual entailment task (Task 4), pretrained language models are also popular. HUKB [19] augmented the positive/negative samples with rules, then finetuned 10 BERT models before ensembling them in different ways. JNLP [11] ensembled various BERT-based classification model for article-statement pairs. KIS [7] extended their previous works of a classic NLP approach with the analysis on predicate-argument structure; they also designed an ensemble method. OvGU [18] ensembled an graph neural network approach and a LegalBERT-based [3] system. TR [14] made use of T5 [12] and GPT-3 [2], Electra [5], and distilled RoBERTa [10] in their approached. UA [8] employed BERT with semantic information.

In summary of the approaches for Task 3 and Task 4, language model-based systems are popular, along with the use of ensemble methods.

3 Approaches

3.1 Task1

In COLIEE 2022 Task 1, we observe that both the query and candidate cases are complex long texts. For example, the average number of words in each case is 4010. It is challenging to represent the whole case law well in a limited semantic space. Thus, we split the documents into paragraphs inspired by the success of BERT-PLI [15] and our previous work [11]. Then, in Runs 1 and 2, we calculate the similarities between cases by combining term-level matching and semantic relationships on the paragraph level. In Run 3, we propose an attention model to encode the whole query in the context of candidate paragraphs, then infer the relationship between cases.

Run 1 and Run 2: Combine lexical and semantic relationships Our previous work [11] has competitive performance in COLIEE 2021 Task 1. Thus, Runs 1 and 2 mainly use the lexical and semantic relationships between paragraphs. Given the query $q = \{q_1, q_2, ..., q_N\}$ and the candidate case
$k = \{k_1, k_2, ..., k_M\}$, where N and M denote the number of paragraphs in q and k, respectively, we use the BM25 model [13] to calculate the lexical mapping

score for each paragraph in q and k. To extract the semantic relationship between query paragraph and candidate paragraph, we leverage one of the largest pre-trained models for legal domain called LegalBERT [4]. The particular reason for us to use this model is the massive data used for training. The training data contains over 300,000 documents collected from different sources, and the training architecture is based on BERT[6], one of the state-of-the-art DL architecture.

We use the following formula to calculate the combined score:

$$combined\_score = score_{BM25} + \alpha * score_{LegalBERT} \tag{1}$$

For each paragraph of the query, we get the most relevant candidate paragraph and then calculate a similarity score between q and the candidate k as follows:

$$similarity\_score_{qk} = \frac{\sum_{i=1}^{N} \max_j(combined\_score_{q_i k_j})}{N} \tag{2}$$

After that, we obtain the top-5 candidates from the whole candidate pool. In Run 2, we additionally design a regulation to filter unreasonable candidates. Because a query only references cases judged before the query case itself, we extract dates from law cases using datefinder [2]. We define the regulation as follows:

$$Rank_1 = \{c | c \in Rank_0 \wedge max(dates(c)) \leq max(dates(q))\} \tag{3}$$

where $dates(d)$ is the set of dates that appeared in document d, $Rank_0$ is the top-5 candidates and $max(dates(d))$ is the date of the case d.

Run 3: The query q and one of the top 200 candidate documents k are represented by paragraphs, denoted as $q = \{q_1, q_2, ..., q_N\}$ and $k = \{k_1, k_2, ..., k_M\}$, where N and M denote the number of paragraphs in q and k, respectively. We use the pre-trained model for legal domain LegalBERT [3] to encode each paragraph. For each query paragraph q_i, we employ the attention mechanism to infer the importance of each candidate paragraph k_j to q_i. The attention weight of each candidate paragraph is measured by:

$$\alpha_{q_i k_j} = \frac{exp(\mathbf{e}_{k_j} \cdot \mathbf{e}_{q_i})}{\sum_{j'} exp(\mathbf{e}_{k_{j'}} \cdot \mathbf{e}_{q_i})} \tag{4}$$

where $\mathbf{e}_{q_i}$ and $\mathbf{e}_{k_j}$ are the encoding of the query paragraph q_i and candidate paragraph k_j, respectively.

We then get the representation of the query paragraph in the context of candidate paragraphs via attentive aggregation:

$$\mathbf{h}_{qki} = \sum_j \alpha_{q_i k_j} \mathbf{e}_{k_j} \tag{5}$$

[2] https://pypi.org/project/datefinder/

The attention mechanism is also employed to infer the importance of each paragraph position to the whole query case. The attention weight of each paragraph position is measured by:

$$\alpha_{qki} = \frac{exp(\mathbf{h}_{qki} \cdot \overline{\mathbf{e}_k})}{\sum_{i'} exp(\mathbf{h}_{qki'} \cdot \overline{\mathbf{e}_k})} \tag{6}$$

We can then get the case-level representation via attention aggregation:

$$\mathbf{d}_{qk} = \sum_i \alpha_{qki} \mathbf{h}_{qki} \tag{7}$$

Finally, the representation d_{qk} is passed through a linear layer followed by a softmax function to predict as follows:

$$\hat{\mathbf{y}_{qk}} = softmax(\mathbf{W}_p \cdot \mathbf{d}_{qk} + \mathbf{b}_p) \tag{8}$$

where R denotes the set of relevance labels, e.g., $R = \{0, 1\}$.
During the training procedure, we optimize the following cross-entropy loss:

$$L_{qk}(\hat{\mathbf{y}_{qk}}, \mathbf{y}_{qk}) = -\sum_{r=1}^{|R|} y_{qkr} log(\hat{y_{qk}}) \tag{9}$$

As for testing, we return top-5 candidates that are predicted as relevant cases corresponding to a given query case by our proposed model. We also apply the date filter similar to Run 2 and combine with the lexical score as two above runs.

3.2 Task2

As we mentioned above, case law often has a complex structure and the likelihood that lexical information can be resolved is very low. As we can see in the Fig 1, the candidate [4] and the given query also contain *"child"*, but this candidate is not a relevant paragraph of the given query. However, instead of *"child"* candidate [18] has *"daughter"* which refer to *"child"* in the corresponding query.

To help DL model learn the relationship such as *"daughter"* refer to *"child"*, we used transformer [17] the state-of-the-art DL model. In many previous research, domain adaptation is very important to enhance the performance of DL model for specific domains. For maximizing the performance of our method, we chose LegalBERT [4], one of the largest pre-trained models for legal domain, and task 2 COLIEE dataset for fine-tuning training data.

Although the performance of the transformer model is impressive, it still has the weakness of not working well with too long sentences ($>$ 512 words). For a given query Q and a corresponding candidate C. If the total length of Q and C more than 512 words, we will used spacy[3] to apply sentence segmentation on C and obtain a set of sentences $S_c = \{c_1, c_2, c_3, ..., c_N\}$. For fine-tuning phrase, we

[3] https://github.com/explosion/spaCy

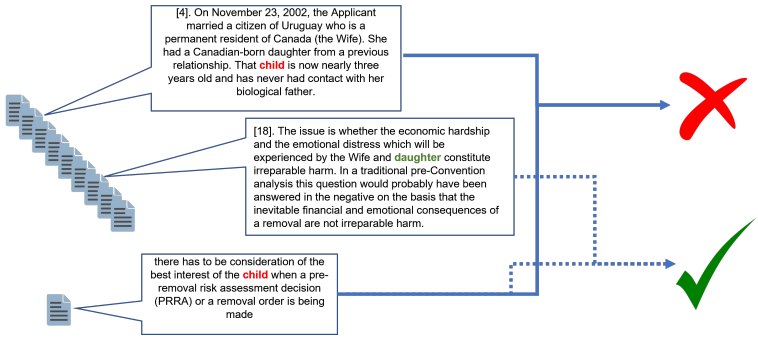


Fig. 1. Ambiguous when using lexical matching

fed the pair (Q - c_i) and corresponding label as a training sample for fine-tuning LegalBERT [3].

After fine-tuning process, we perform the combination between lexical matching and semantic matching as follows:

LegalBERT combine with BM25: If the total word is more than 512, we will apply sentence segmentation to obtain the total semantic matching score (TSMS) for the given query and candidate paragraph as:

$$TSMS = \frac{\sum_{i=1}^{N} O_i}{N} \tag{10}$$

where N is the total number of sentences after applying sentence segmentation, O_i is semantic score of the corresponding sentence. Then we used equation (1) to obtain the combination score of LegalBert [3] and BM25 algorithm. After having all the candidate scores for the given query, the highest score was assigned as the relevant paragraph. Furthermore, we identify other relevant paragraphs by setting a threshold. If exist a candidate score more than the threshold, we will assign those candidates as relevant paragraphs.

LegalBERT combine with BM25(AMR + spacy POS tagging): In our opinion, sentences always have a lot of meaningless words such as stop words, pronouns, etc. These words have no value in searching or calculating similarity. In this run, we use Abstract Meaning Representation (AMR) to capture the most important words in the query and corresponding candidate paragraph. After that, we performed BM25 algorithm on these important words. At last, we used the same approach as the first run to get the set of relevant paragraphs.

LegalBERT filtered by BM25(AMR + spacy POS tagging): In this run, we didn't combine the lexical and the semantic score together. We used LegalBert [3] and set up a threshold to get top N candidates, and AMR combine with BM25 as the second run to obtain M candidates. The relevant paragraphs are identified by $N \cap M$.

Example	Statement	Relevant Articles	Decision
1	Juristic act subject to a condition subsequent which is impossible is impossible shall be void.	**Article 133** (1) A juridical act subject to an impossible condition precedent is void. (2) A juridical act subject to an impossible condition subsequent is an unconditional juridical act.	No
2	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 owns land X, B needs to obtains the permission of a family court in order to sell Land X as an agent of A.	**Article 28** If an administrator needs to perform an act exceeding the authority provided for in Article 103, the administrator may perform that act after obtaining the permission of the family court. [...] **Article 103** An agent who has no specifically defined authority has the authority to perform the following acts only: (i) acts of preservation; [...]	Yes

Table 1. Some examples of training data for Task 3.

3.3 Task3

As mentioned in our previous work [11], there are two main challenges for Task 3. While all of the state-of-the-art models for this task in recent years employ DL techniques, the limitation of the number of maximum tokens a DL model can handle is a challenge for DL approaches. In other words, the first challenge is to address the long articles in the corpus. Of course, for this challenge can be overcome by lexical-based techniques, however the lexical-based techniques demonstrated humble performance compared to DL models. The second challenge is to address use-case statement (see example 2 in Table 1). Specifically, use-case statement are the statements which describe specific circumstances. Most of the use-case statements are written with characters indicating a person or a legal person, e.g., "The family court appointed **B** as the administrator of the property of absentee **A**".

Figure 2 shows the flow of our system. Inspired by the success of DL models for Task 3, we employ BERT [6] variants for the task of transfer learning. To address the long-article challenge, we utilize Longformer [1] model because it can handle long context. For the second challenge, which is addressing use-case statement, we propose to train a DL model to learn the specific relevancy patterns of only the pairs of use-case statements and their relevant articles. We call this model the use-case model. In our experiments, the use-case model demonstrated its distinguished learned patterns by predicting some retrieval

results that cannot be seen by other models which are trained with the full data (data including both non-use-case statements and use-case statements).

We suppose that it is because the use-case model can provide a different "view" on the use-case data. Furthermore, we also observed that different BERT variants have different "views" on the full data. Hence, we experimented with many BERT variants and ensemble their predictions for the final prediction. Briefly, we (1) employ transfer learning with many BERT variants on the full data, (2) filter use-case statements and finetune a BERT model on them, and (3) selectively ensemble the predictions of those models' predictions to construct the final prediction.

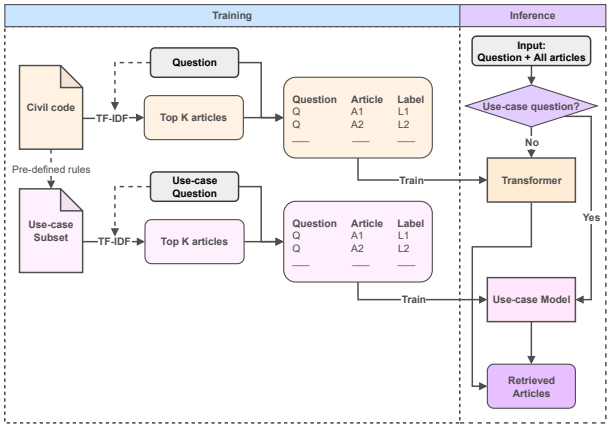


Fig. 2. Task 3 flow

Transfer learning with BERT variants. We finetune the pretrained models of many BERT variants (e.g., Longformer [1], LegalBERT [4], ELECTRA [5], RoBERTa [10]) on the provided training dataset with the task of sentence pair classification. Specifically, for a statement, we get positive training data by pairing it with each of its relevant article; for negative examples, we pair a statement with its non-relevant articles, where the non-relevant articles are selected based on their TF-IDF similarity with the statement (we select top K TF-IDF score).

Addressing use-case statements. First, we build the training data by filter the use-case statements. As mentioned above, most of the use-case statements are written with characters indicating a person or a legal person, we design a regular expression string to filter out the use-case statements. Second, we utilized a LegalBERT model which is finetuned on the full data as the initial model to be finetuned with the use-case data. The finetune task is the same as above.

Model ensembling. Finally, given a test data, we run the finetuned models to get predictions, then we ensemble their predictions with many settings and

choose the best setting of ensembling for the final prediction. While the Longformer model is supposed to be effective when dealing with long articles, the LegalBERT [3] model is supposed to better leverage the learned knowledge in its pretraining phase, and ELECTRA is supposed to provide a "more distinguished view" to the data due to the robustness of its discriminator.

3.4 Task 4. Statute Law Entailment and Question Answering

This task involves the identification of an entailment relationship between relevant articles $A_{i|1\leq i\leq n}$ (derived from Task 3's results) and a question Q. The models are required to determine whether the relevant articles entail "Q" or "$notQ$". Given a pair of legal bar exam question and article (Q, A_i), the models return a binary value for determining whether (A_i) entails (Q).

Through analysis, we find out that there is always a passage containing important information to answer the question. Based on this assumption, we employ TF-IDF to explore the most important passage from the given article. Furthermore, we use the same approach as Task 3 to address the use-case statements. Therefore, we also propose to train a DL model to learn the specific relevancy patterns of the pairs of use-case statements and their relevant paragraph. Similar to Task 3, we experimented with many BERT variants and ensemble their predictions for the final prediction because different BERT variants can learn different aspects of the data. Figure 3 shows the overall of our framework for Task 4.The key designs of our method is as follows.

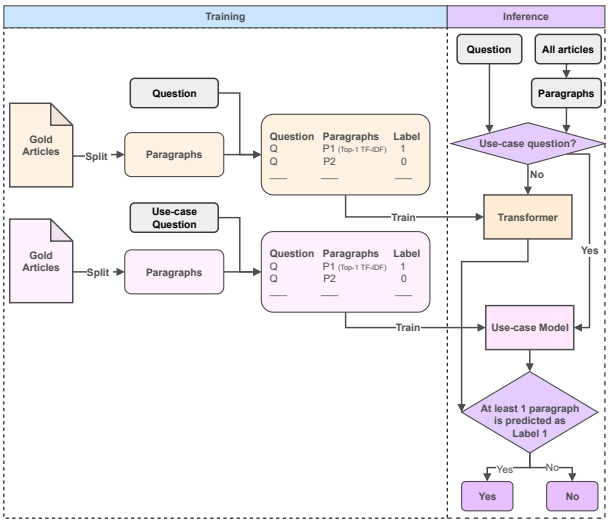


Fig. 3. Task 4 flow

Paragraph level entailment Instead of using the provided relevant articles, we split each article into paragraphs, each paragraph describe a separate subject and does not contains information of other paragraphs. The paragraphs in an article is denote by the prefix number such as (1), (2), etc., we use this rule to split the article into paragraphs.

Negation In previous competitions, data augmentation using negation proves to provide better result. Therefore, we employ negation at the paragraph level to generate more training data.

Transfer learning with BERT variants. We finetune the pretrained models of many BERT variants (e.g., Longformer [1], LegalBERT [4], ELECTRA [5], RoBERTa [10]) on the provided training dataset with the task of sentence pair classification. After splitting the relevant articles into paragraphs, we pair each paragraph with the question. The TF-IDF scores are calculated with the question as the query and the paragraphs as the corpus. The paragraph with the highest TF-IDF score has label "YES", other paragraphs have label "NO".

Use-case statements and model ensemble Similar to Task 3, we also address use-case statements and model ensemble. The approach to use-case statements is the same as in Task 3, we use a LegalBERT [3] model which is finetuned on the full data as the initial model and continue to finetune it with the use-case data. For model ensemble, we sum the prediction scores of all model for each label (YES and NO) and use this score for prediction.

4 Experiments and Results

4.1 Task 1.

Based on a comparison between the performances with different α values, we set the hyperparameter α to be 80 for Run 1 and α to be 30 for Run 2. Another key hyperparameter that needs to determined is K, the number of retrieved cases per query. According to the evaluation results on the training set, we set K to be 5

Table 2. Task 1 experimental results

Run	Precision	Recall	F1
Run 1	0.2446	**0.2866**	0.2639
Run 2	0.3144	0.2494	0.2781
Run 3	**0.3211**	0.2502	**0.2813**

Table 2 shows the performance on the test set. The result shows that the date filter in Runs 2 and 3 significantly improves the retrieval model's performance.

Employing the attention model to encode the whole query in the context of the candidate paragraphs and infer the candidate relevance (Run 3) achieves competitive performance, which shows that our proposed model is effective in legal case retrieval.

4.2 Task2

1. **LegalBERT combine with BM25:** In this run, we set $\alpha = 0.9$ for semantic score and $1 - \alpha$ for lexical score, $threshold = top1_{score} - 0.003$
2. **LegalBERT combine with BM25(AMR + spacy POS tagging:** For the second run, we parsed sentences into AMR graph and used spacy to filter the most important POS tagging such as: verb, noun, adjective, etc. After that, we performed BM25 algorithm on these sets of POS tagging. We also keep the α and threshold as the first run.
3. **LegalBERT filtered by BM25(AMR + spacy POS tagging):** As we have described in Section 3, we only set up *alpha* and threshold for semantic scores to get top N relevant paragraphs. For the lexical candidates, we chose top 2 candidate as relevant paragraphs. Finally, we performed intersection between semantic relevant paragraphs and lexical relevant paragraphs.

All the results of our methods are shown in Table 3

Table 3. Task 2 experimental results

Run	Precision	Recall	F1
Run 1	0.6612	0.6452	0.6780
Run 2	0.6452	0.6154	0.6780
Run 3	**0.6694**	**0.6532**	**0.6864**

4.3 Task 3.

Types of prediction We further assess the impact of the types of prediction on the performance of the model. We investigate two types of prediction. (1) Relax: If no article is classify as relevant to the question, we will retrieve one article with the highest confidence score. (2) Strict: If no article is classify as relevant, then no article is retrieved. The result suggests that the Relax prediction has better performance.

Metadata Metadata of the article includes the information about Part, Chapter, Section, Subsection, Division, and Annotation of the articles. These experiments try to assess whether the present of these metadata enhance the performance of the model. The result shows that the metadata significantly improves the model prediction.

From our observation from experiments, we decide to employ TF-IDF filter to 100 candidates, ratio of possitive and negative is 1:100, Relax prediction, and include metadata. We conduct experiments with these settings on various Transformer models. Table 4 shows the performance of these models on the development set.

Table 4. Task 3 result of chosen Transformer models

Method	Macro F2	Precision	Recall	Return	Retrieved
LEGAL-BERT$_{\text{Base}}$	70.93	0.68	**0.74**	104	**65**
ELECTRA$_{\text{Base}}$	**71.87**	**0.73**	0.73	93	64
RoBERTa$_{\text{Base}}$	69.42	0.70	0.71	94	63
Longformer$_{\text{Base}}$	69.09	**0.73**	0.69	86	62
ALBERT	68.72	0.70	0.69	86	60
DeBERTa	67.22	0.72	0.67	82	59

We also conduct experiments on the use-case question subset to verify the efficiency of the use-case model and report the result in Table 5. The use-case model was able to correctly retrieve some articles that the normal model cannot. Below is an an example from the test set where the Use-case model outperforms the normal model.

Example: If D owes C a debt (Y) of 300000 yen that is set off against the debt (X), and D demands payment of 600000 yen from A while C does not use a set-off for the debt (Y), A may refuse to pay 200000 yen of that debt.

Table 5. Task 3 result on Use-case question subset

Method	Macro F2	Precision	Recall	Return	Retrieved
LEGAL-BERT$_{\text{Base}}$	58.06	0.61	0.58	33	20
LEGAL-BERT$_{\text{Base}}$ Use-Case	**58.42**	0.58	0.60	36	20

Finally, we ensemble models, including Use-case models, and present the result in Table 6. The result indicates that the ensemble models have better result than single models, and Use-case models can further enhance recall and F2 score. We selected 3 methods from the result to submit our 3 runs.

Table 6. Task 3 Result of ensemble models including Use-case models. (*) Indicates that the models are used in the 3 submission runs.

Method	Macro F2	Precision	Recall	Return	Retrieved
ELECTRA + Longformer + RoBERTa + Use-case (*)	77.27	0.63	**0.86**	155	**80**
ELECTRA + Longformer + RoBERTa	76.58	0.65	0.83	139	77
RoBERTa + Longformer + Use-case (*)	**77.84**	0.67	0.85	139	78
RoBERTa + Longformer	75.52	**0.69**	0.80	119	73
ELECTRA + LEGAL-BERT + Longformer + RoBERTa (*)	76.21	0.63	0.85	154	78

4.4 Task 4.

According to our experiments, we draw the conclusion that data argumentation and important passage mining affect the performance of an information retrieval system.

Finally, we ensemble models after applying negative data argumentation and important passage mining. The results are in Table 7. We choose 3 most robust methods for the blind test.

Table 7. Task 4 Result of ensemble models. (*) Indicates that the models are used in the 3 submission runs.

Method	**Accuracy**
$\text{ELECTRA}_{\text{Base}}$ (*)	64.19
$\text{ELECTRA}_{\text{Base}}$ + LEGAL-BERT (*)	64.19
$\text{ELECTRA}_{\text{Base}}$ + LEGAL-BERT + Use-case (*)	61.72
$\text{ELECTRA}_{\text{Base}}$ + Longformer	62.96
$\text{ELECTRA}_{\text{Base}}$ + RoBERTa	55.55

5 Conclusion

In this paper, we have performed several DL approaches for tackling the challenges in this competition such as processing long documents, ambiguous sentence classification, shortage in training data, etc. The experiments and results of the blind test have convinced that our proposed methods can address a certain aspect of the legal deomain.

References

1. Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.
2. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
3. Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.
4. Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.
5. Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
7. Masaki Fujita, Naoki Kiyota, and Yoshinobu Kano. Predicate's argument resolver and entity abstraction for legal question answering: Kis teams at coliee 2021 shared task. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021.
8. Mi-Young Kim, Juliano Rabelo, and Randy Goebel. Bm25 and transformer- based legal information extraction and entailment. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021.
9. Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.
10. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
11. Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021. *arXiv preprint arXiv:2106.13405*, 2021.
12. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
13. Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.
14. Frank Schilder, Dhivya Chinnappa, Kanika Madan, Jinane Harmouche, Andrew Vold, Hiroko Bretz, and John Hudzina. A pentapus grapples with legal reasoning. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021.
15. 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 *IJCAI*, pages 3501–3507, 2020.
16. Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligent analysis*, volume 2, pages 2–6. Citeseer, 2005.
17. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
18. Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W De Luca. Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 285–294, 2021.
19. Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. Bert-based ensemble methods with data augmentation for legal textual entailment in coliee statute law task. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 278–284, 2021.

Legal Textual Entailment using Ensemble of Rule-based and BERT-based method with Data Augmentations including Generation without Excess or Deficiency

Masaki Fujita, Takaaki Onaga, Ayaka Ueyama and Yoshinobu Kano

Shizuoka University, Johoku, 3-5-1, Naka-ku, Hamamatsu-shi, Shizuoka, 432-8011 Japan

mfujita@kanolab.net, tonaga@kanolab.net,
aueyama@kanolab.net,kano@inf.shizuoka.ac.jp

Abstract.
We report our system architecture of COLIEE 2022 Task 4, which challenges to solve the textual entailment part of the Japanese legal bar examination problems. We successfully improved the correct answer ratio by an ensemble of a rule-based method and BERT-based method. Our proposed methods mainly consist of two parts: data augmentations of training dataset and an ensemble of the methods. Regarding training data augmentation, the civil law articles are segmented once and reconstructed again with all the combinations. Data expansion is then performed by replacing the data with negative forms and alphabetical symbols. Focusing on the characteristics that the rule-based method is high in its precision but low in its coverage, we employed a modular way in our ensemble. We integrated other proposed methods such as Sentence-BERT to select necessary data, person name inference to replace alphabetical anonymized symbols with the actual role name of the person. We confirmed that our suggested methods are effective by comparing with our baseline models, achieved 0.6789 correct answer ratio in accuracy on the formal run test dataset, which was the best score among the COLIEE 2022 Task 4 submissions.

Keywords: COLIEE, Question Answering, Legal Bar Exam, Legal information Extraction, BERT, Predicate Argument Structure Analysis.

1 Introduction

The COLIEE shared task series was held in association with the JURISIN (Jurisinformatics) workshops and ICAIL (International Conference on Artificial Intelligence and Law) workshops [1] [2] [3] [4] [5] [6] [7] [8]. COLIEE 2022 is the ninth shared task, which consists of four tasks. Task 1 and 2 are case law tasks using Canadian law. Task 3 and 4 are statute law tasks which use the Japanese legal bar examination. In Task 3, participants are given a problem text, then search for Japanese civil law articles which are necessary to solve the problem. In Task 4, participants are given a problem text and related article(s), then asked to answer Yes or No whether the content entails the given article(s) or not regarding the given article(s). We describe

our participation of COLIEE 2022 Task 4 in this paper. In Task 3, participants are given a problem, then search articles which are necessary to solve the problem. In Task 4, participants are given a problem and related article(s), then answer Yes or No whether the content entails the given article(s) or not regarding the given article(s). We challenged Task 4.

Unlike machine learning-based systems, where the reasoning process is a black box, rule-based systems are able to reason according to human thought, and thus are better in terms of explainable output, which is important when practically automating the trial process.

In Task 4 of the previous COLIEE 2021 shared task, we built a solver system that uses a rule-based solution method [9]. Other teams mainly used machine learning-based methods, such as teams using BERT [10] [11] [12], ensembles based on T5 [13], and graph neural networks [14]. We were the only team who employs a rule-based method.

Our previous rule-based system outputs different correct answers for each module within the rule-base system. Therefore, an ensemble by selecting modules is a straightforward extension of our previous modular system. While rule-based methods have advantages on their good precision scores for specific types of problems, their total performance decreases when we try increasing their recall scores in general. We implemented a machine learning-based system using BERT, in order to cover solving other types of problems in higher performance in which our rule-based system shows lower performance.

In addition to the ensemble of the rule-based and BERT-based parts above, we propose three data augmentation method for the training data of BERT. Firstly, we segment the text once and then reconstruct in various combinations. This process can create texts that are easy for BERT to infer, by containing no excess or deficiency of necessary data, and by reducing string lengths. Secondly, we augment the problem texts by converting into negative forms at the end of sentences. Thirdly, we replace person role names with alphabetical symbols. Fourthly, we replace referring article numbers by their referred actual article texts.

Finally, two new methods were added: data selection, which selects appropriate data from the combinatorial augmented data, and person name inference, which replaces alphabetical person name symbols with actual person role names.

These techniques resulted in a 67.89% correct answer ratio in accuracy on the COLIEE 2022 formal run's test data, which was the best score among the teams that submitted data. We compare results of our proposed methods with our baselines, discusses detailed analysis in the following sections.

2 Previous Works

Hoshino, et al. [15] suggested a rule-based solver system in COLIEE 2019, which syntactically parses sentences, and then sentences are segmented into clauses by their original definition, based on the parsing results. The parsing results are then used again to extract a clause set (subject, predicate, and object) for each clause. They built

a couple of solver modules: for example, their Precise match module compares the relevant civil law clauses with the problem clauses using all elements in the clause sets and answers Yes if they match, while their Loose match module answers Yes if one or more clause elements match. We used one of the modules of Hoshino et al.'s system, but we propose to perform ensemble with machine learning based systems, while Hoshino et al. employed only the rule-based system.

Yoshioka, et al. [10], the best team in Task 4 of COLIEE 2021, improved its performance mainly by augmenting the training data and by an ensemble of differently trained BERT [16] models. BERT is a frequently used deep learning model, which uses Transformer's encoder part. For their training data expansion, they used a list of civil law articles. They increased the training data size by segmenting the sentences, and by inserting negations that invert the sentence's logic. In the ensemble of BERT models, they selected models which showed better performances among the prepared BERT models. We reused their methods but also added new data augmentation methods to Yoshioka, et al.'s, data extension to the problem texts and data extension by replacing alphabetical symbols and article numbers.

3 System

3.1 System Overview: Rule-based part and BERT-based part

We only used resources of Japanese languages only including the official ones of COLIEE, while organizers provide both English and Japanese versions in Task 4.

Our solver system consists of a rule-based part and a BERT-based part. These parts are integrated by an ensemble, then output the final Yes/No answers.

In the previous work [9, 15], there were a couple of solver modules. We examined different ways of ensemble of these solver modules, but the modules other than the Precise Match module have not contributed to the scores. Therefore, we use the Precise Match module, which compares all of subjects, objects and predicates of the clause pairs, as our rule-based part.

The BERT-based part includes multiple BERT models with different fine-tunings, then integrated by an ensemble. This part is similar to the previous work by Yoshioka, et al. [10], but we extended their work as follows.

All of our BERT models use a pre-trained BERT model [16], which was trained by the Japanese Wikipedia articles. Our BERT models are fine-tuned as inputting the given problem text and the given relevant civil law articles, and outputting Yes/No binary answers. We set parameters as follow: the learning rate to 4e-5, the maximum string length to 512, the number of epochs to 6, and the batch size to 32. The difference of our BERT models is that they are fine-tuned by different datasets.

We explain our system process flow in the following subsections. Firstly, data augmentation is performed on the training and test data (3.2). Among our three formal run submissions (**KIS1**, **KIS2** and **KIS3**), **KIS2** is our suggested model. **KIS1** and **KIS3** selects which data to use of the augmented data (3.3), in addition to **KIS2**. **KIS3** further performs person name inference (3.4), which replaces anonymous per-

son names with inferred concrete person names to create new dataset. All of these three models finally perform an ensemble of the rule-based and BERT-based parts. then to output the final solution. Figure 1 illustrates the above process.

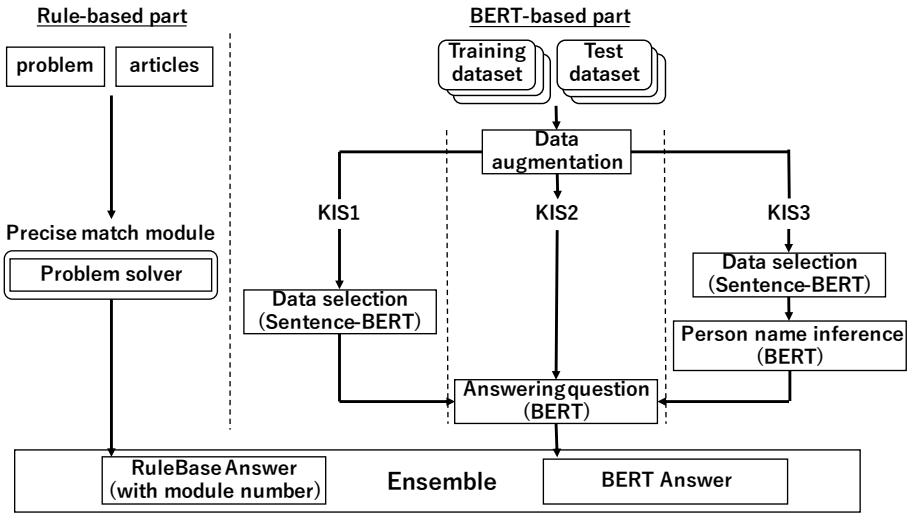


Figure 1. System Overview

3.2 Data Augmentation

We apply our data augmentation to the BERT-based part only. We apply our data augmentation method to both the civil law articles and the problem sentences, both training data and test data.

Data Augmentation by segmenting Civil Code Articles into Sentences. The relevant civil law articles, which are needed to answer a problem by textual entailment, may be segmented into multiple articles (Figure 2).

[Civil Code Article]
Article 679. In addition to the cases set forth in the preceding Article, a partner shall withdraw from the Partnership for any of the following things. (第六百七十九条　前条の場合のほか、組合員は、次に掲げる事由によって脱退する。)
1, death (一　死亡)
2, The decision to commence bankruptcy proceedings has been made. (二　破産手続開始の決定を受けたこと。)
3, The applicant has received a trial for commencement of guardianship. (三　後見開始の審判を受けたこと。)
4, Expulsion (四　除名)
Article 681 Calculations between a partner who has withdrawn and the other partners shall be made in accordance with the status of the assets of the Partnership at the time of withdrawal. (第六百八十一条　脱退した組合員と他の組合員との間の計算は、脱退の時における組合財産の状況に従ってしなければならない。)

2, The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of contribution made by the partner. (2, 脱退した組合員の持分は、その3 出資の種類を問わず、金銭で払い戻すことができる。)
3, Calculations may be made after the completion of any matter that has not yet been completed at the time of withdrawal.(3, 脱退の時にまだ完了していない事項については、その完了後に計算をすることができる。)
[Question text]
A partner may receive a refund of his/her interest even if he/she is expelled. (組合員は，除名された場合であっても，持分の払戻しを受けることができる。)

Figure 2. Examples of multiple article segmentations in Civil Law (H20-27-I)

This segmentation process shortens the input string length and augments the training data. In Figure 2, the example articles data is segmented into units of *articles* designated as Article 679 and Article 681, and in units of *terms* as well described as *1("一"), 2("二")...* in the same figure. Article 679 includes the statement *the following things*("次に掲げる") , which is followed by a description of what it specifies. For such a sentence, we substitute *the following things*("次に掲げる") by each of the contents specified by the description. Figure 3 shows the result of this process for the example in Figure 2, segmented into seven sentences.

1. In addition to the cases described in the preceding Article, a partner shall withdraw due to death.
(1，前条の場合のほか、組合員は、死亡によって脱退する。)
2. In addition to the cases described in the preceding Article, a partner shall withdraw from the Partnership upon a decision to commence bankruptcy proceedings.
(2，前条の場合のほか、組合員は、破産手続開始の決定を受けたことによって脱退する。)
3, In addition to the cases described in the preceding Article, a partner shall withdraw from the Partnership upon a decision of commencement of guardianship.
(3，前条の場合のほか、組合員は、後見開始の審判を受けたことによって脱退する。)
4, In addition to the cases described in the preceding article, a partner shall withdraw from the Partnership by expulsion.
(4，前条の場合のほか、組合員は、除名によって脱退する。)
5, Calculations between a partner who has withdrawn and the other partners shall be made in accordance with the status of the assets of the Partnership at the time of withdrawal.
(5，脱退した組合員と他の組合員との間の計算は、脱退の時における組合財産の状況に従ってしなければならない。)
The interest of a partner who withdraws may be refunded in cash, regardless of the type of investment.
(6，脱退した組合員の持分は、その出資の種類を問わず、金銭で払い戻すことができる。)
With respect to matters that have not yet been completed at the time of withdrawal, calculations may be made after the completion of such matters.
(7，脱退の時にまだ完了していない事項については、その完了後に計算をすることができる。)

Figure 3. Examples of segmented article texts

Reconstruction. We reconstruct sentences from segmented data, to create dataset that contains necessary data without excesses of deficiencies within all possible combinations, in shorter sentences then the original ones. For example, if a civil law article is segmented into seven sentences, 127 new patterns of sentences are generated. Figure 3 shows an example of such seven sentences but shows 1, 4, 6 only due to the space limitation.

1, In addition to the cases mentioned in the preceding article, a member shall withdraw from the association upon his/her death. (1，前条の場合のほか、組合員は、死亡によって脱退する。)

4, In addition to the cases described in the preceding article, a member shall be withdraw by expulsion from the Partnership. (4，前条の場合のほか、組合員は、除名によって脱退する。)

6, The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of investment. (6，脱退した組合員の持分は、その出資の種類を問わず、金銭で払い戻すことができる。)

1+4, In addition to the cases mentioned in the preceding article, a member shall withdraw from the association upon his/her death. In addition to the cases described in the preceding article, a member shall be withdraw by expulsion from the Partnership. (1+4，前条の場合のほか、組合員は、死亡によって脱退する。前条の場合のほか、組合員は、除名によって脱退する。)

1+6, In addition to the cases mentioned in the preceding article, a member shall withdraw from the association upon his/her death. The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of investment. (1+6，前条の場合のほか、組合員は、死亡によって脱退する。脱退した組合員の持分は、その出資の種類を問わず、金銭で払い戻すことができる。)

4+6, In addition to the cases described in the preceding article, a member shall be withdraw by expulsion from the Partnership. The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of investment. (4+6，前条の場合のほか、組合員は、除名によって脱退する。脱退した組合員の持分は、その出資の種類を問わず、金銭で払い戻すことができる。)

1+4+6, In addition to the cases mentioned in the preceding article, a member shall withdraw from the association upon his/her death. The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of investment. The interest of a partner who has withdrawn may be refunded in cash, regardless of the type of investment. (1+4+6，前条の場合のほか、組合員は、死亡によって脱退する。前条の場合のほか、組合員は、除名によって脱退する。脱退した組合員の持分は、その出資の種類を問わず、金銭で払い戻すことができる。)

Figure 4. Examples of combinations reconstructed from 1, 4, 6

In this example, the necessary information to solve the problem is 4 and 6, because 4 indicates that the expulsion is a withdrawal, and 6 indicates that the withdrawn member is entitled to a refund of his/her share. Therefore, in this example, the data constructed with 4+6 is an appropriate data set that contains the necessary information without excess or deficiency.

This automatic reconstruction will generate a large dataset, which could include data that are not suitable for the solution. We introduce Data Selection (3.3) as a solution to this issue.

Reference Substitution. Civil law articles sometimes refer to other articles by their article numbers as shown in Figure 5.

Article 572. Even where the seller has made a covenant not to assume liability for security in the case prescribed in the main text of **Article 560 paragraph 1** or **Article 565**, he/she may not be exempted from liability for facts that he/she knew but did not disclose and for rights that he/she established for a third party or assigned to a third party.
(第五百七十二条　売主は、**第五百六十二条第一項本文又**は**第五百六十五条**に規定する 場合における担保の責任を負わない旨の特約をしたときであっても、知りながら告げなかった事実及び自ら第三者のために設定し又は第三者に譲り渡した権利については、その責任を免れることができない。)

Figure 5. Examples of articles referring to other articles by article numbers

Because the referred "main text of Article 560 paragraph 1" and "Article 565" are not given as relevant articles, we need other information than the relevant articles. Since such reference patterns are complex, e.g., the "main text" excludes the proviso portion, or the section number may be specified after the article number, we manually replaced the references to other articles in all civil law articles. Because the relevant article texts are subset of the distributed civil law article, and there is no problem text which specifies article numbers, we do not need to modify the test dataset.

In some cases, the length of replaced sentence could get very long, e.g., when multiple article numbers are specified, or when the specified article refers to other in a nested way. We not only use the replaced data but also the data before the replacement in training and testing, thereby preventing cases in which a problem cannot be answered due to the long sentences.

Data Generation by Negative Form. We apply data expansion by performing a logical inversion at the end of a sentence. For this logical inversion, we manually list pairs of logically opposite character strings in advance, such as "can" and "cannot" or "will" and "will not". When this negative form expansion is applied, we invert the answer labels. This is similar to the previous work in section 2, but we apply this process to the problem text, in addition to the previous work's civil law text.

For questions where the answer is No, the negative form at the end of the sentence does not necessarily result in a Yes. We also tried a method that convert from Yes to No only.

AB Replacement. Person names are sometimes replaced by alphabetical characters in the problem texts, such as A and B, as a sort of anonymization. In order to deal with such problems, we create augmentation dataset from the training data, by replacing person names with alphabetic characters.

We applied this expansion when person's names appear more than once, because we aim to fine-tune the model to correctly identify person names when there are two or more choices. As same as the past problem texts, we replace the person's name which firstly appear with A, secondly appear with B, and so on. Since such problem

texts are written in the Japanese full-width symbols, we replace them with full-width alphabetic characters.

[Before replacement]
(Civil Code Article)
In the case of possession by an agent, the right of possession shall be extinguished by the principal's renunciation of the intention to allow the agent to take possession.
(代理人によって占有をする場合には、占有権は、本人が代理人に占有をさせる意思を放棄したことによって消滅する。)

(Problem text)
When possession is taken by an agent, the right of possession shall be extinguished by the principal's renunciation of the intention to allow the agent to take possession. (代理人によって占有をする場合には、占有権は、本人が代理人に占有をさせる意思を放棄したことによって消滅する。)

[After replacement]
(Civil Code Article)
same

(Problem text)
When possession is taken by an A, the right of possession shall be extinguished by the B's renunciation of the intention to allow the A to take possession. (Aによって占有をする場合には、占有権は、BがAに占有をさせる意思を放棄したことによって消滅する。)

Figure 6. An example applying alphabetical substitution to a problem using Article 204

Variation and Decision of Final Answers. When the above replacements and reconstructions are applied to a single problem, that single problem could be duplicated with different augmentation methods, resulting in potential multiple different answers. For each duplicated problem, we compare BERT's confidence values, then accepts the answer which has largest difference between the confidence scores of Yes and No.

Because different fine-tuning trail could result in different answers even when the same training dataset is used, we tried the fine-tuning five times, then accepted the majority vote results as the final output.

3.3 Data Selection

We calculate the similarity between the problem text and the generated civil law text for this selection, using pretrained Sentence-BERT [17]. We regard top five similar texts among texts after data segmentation, reconstruction, and replacement processes as the appropriate data set.

3.4 Person Name Inference

As described earlier, person names are sometimes replaced by alphabetical characters in the problem texts, such as A and B, as a sort of anonymization. We trained a BERT model to predict which alphabet represents which names, from semi-manually created 49 candidates: the person himself, agent, seller, buyer, etc. **KIS3** uses this prediction

result, replacing alphabets with prediction results as preprocessing. Contents of this section is applied to **KIS3** only.

If the problem directly indicates what the alphabet represents as appositional expression, we apply replacements by the following two rules, before applying the above BERT's predictions. Firstly, we replace all occurrences of "A" with appositions "XX person", when there are indications in the text, such as "the XX person (A)" (XX(人物名)者であるA), "A who is the XX…" (XX者A). Secondly, we replace expressions such as "between AB" ("AB間") with "between A and B" ("AとBとの間"), to be closer to expressions of the civil code. After these processes applied, our BERT-based system predicts replacements for the rest of the alphabetical characters.

We eliminated problems with alphabetical characters as person names in the training dataset, because such problems are converted into non-alphabetical symbols by applying the above process.

3.5 Ensemble of Rule-based part and BERT-based part

We built an ensemble of the rule-based part and the BERT-based part. As shown in Figure 1, the rule-based part (precise match) is applied first, because the rule-based part is better in its precision but lower in its coverage. If the rule-based part can output an answer, then that answer will be used as the system's final output; if the rule-based part cannot output its answer, then output of the BERT-based part will be used as the system's final answer.

4 Result and Experiments

4.1 Formal Run Results

Table 1 shows the COLIEE 2022's Task 4 formal run evaluation results for each team's submission. **KIS** is our team's name.

Table 1. COLIEE 2022 Task 4's formal run results for each participant's submission. "Correct" column shows the number of correct answers among 109 problems in total; "Accuracy" column shows the accuracy ratio.

Team	Submission ID	Correct	Accuracy
KIS	**KIS2**	**74**	**0.6789**
HUKB	HUKB1	73	0.6697
KIS	**KIS1**	**72**	**0.6606**
KIS	**KIS3**	**72**	**0.6606**
HUKB	HUKB2	69	0.633
HUKB	HUKB3	69	0.633
LLNTU	LLNTUdeNgram	66	0.6055
LLNTU	LLNTUdiffSim	63	0.578
OVGU	OVGU3	63	0.578
UA	UA_e	59	0.5413
UA	UA_r	59	0.5413
UA	UA_structure	59	0.5413
JNLP	JNLP1	58	0.5321
JNLP	JNLP2	58	0.5321
UVGU	OVGU2	58	0.5321
JNLP	JNLP3	56	0.5138
OVGU	OVGU1	52	0.4771

4.2 Experiments on Training Dataset

Table 2 shows the evaluation results using the H18-R02 training dataset, which includes 15 years' datasets, 887 problems in total. For each year of the training dataset, we regard that target year's dataset as a test dataset, and the rest of the years' datasets as a training dataset. For H30-R02 datasets, we follow the same setting with the previous formal runs i.e. regard the rest of the past datasets as the training dataset but not use "future" datasets of the target year. Results in Table 2 show total number of answered problems, the total number of correctly answered problems, and correct answer ratios, for each solver.

The first row in Table 2, **Individual**, shows individual solver's results; because we performed the fine-tuning five times as described earlier (3.2), count values in the *BERT-based* columns show summation of these five differently fine-tuned models. The second row, +**Majority Vote**, shows results of the majority vote integration of these five fine-tuned models. The third row, +**Ensemble**, shows results of an ensemble of the **Rule-based** part, which corresponds to the Precise Match module, and the BERT-based part, which corresponds to the majority vote integration.

Baseline column corresponds to our baseline model, which is same as the other BERT-based models but no data expansion is applied. Because the data expansion includes the sentence segmentation, inputs to the *Baseline* model sometimes exceed the maximum length of the BERT model. Our **Baseline** model does not output answers in such cases, thus the total number of the answered problems in Table 2 is different from other models.

KIS1_N is the implementation of the process of making the negative form at the end of the sentence only for the answers with Y.

KIS1_N removes negative form data augmentation from **KIS1** when the original problem's gold standard answer is *No*, because the answer may not be reversed to *Yes* by converting into negative forms when the original problem's gold standard answer is *No*. We did not submit **KIS1_N** as one of our formal runs.

Table 2. Results of training data execution.

	Rule-based	BERT-based				
		KIS1	KIS2	KIS3	KIS1_N	Baseline
Individual	545/887 (0.614)	**2688/4435 (0.606)**	2635/4435 (0.594)	2674/4435 (0.603)	**2681/4435 (0.605)**	2433/4135 (0.588)
+Majority vote	-	544/887 (0.613)	540/887 (0.609)	**553/887 (0.623)**	549/887 (0.619)	494/827 (0.597)
+Ensemble	-	558/887 (0.629)	557/887 (0.628)	559/887 (0.63)	**565/887 (0.637)**	505/827 (0.611)

Table 3 shows the coverage and accuracies of our **Rule-based** system: of the 887 questions in the training data, 142 questions were covered and answered by our **Rule-based** system, which corresponds to the **Rule-coverable Problems** row in Table 3; the rest of the training data is 745 questions, which corresponds to the **Others** row**.** The cell of **Rule-based/Others** shows the results when we applied our **Rule-based** system to the originally non-applicable problems, which is not used in our formal run sys-

tems. In other words, the summation of the cell values of **Rule-based/Rule-coverable Problems** and **KIS3/Others** in Table 3 is equals to the value in the **KIS3/+Ensemble** cell in Table 2.

Table 3. Coverage of our rule-based systems and accuracy scores for each system (**BERT-based KIS1-3** corresponds to the Majority vote version)

	Rule-based	BERT-based		
		KIS1	KIS2	KIS3
Rule-coverable Problems	**114/142 (0.803)**	100/142 (0.704)	97/142 (0.683)	108/142 (0.761)
Others	431/745 (0.579)	444/745 (0.596)	443/745 (0.595)	**445/745 (0.597)**

Regarding the person name inference issue (3.4), our system identified 173 of the 887 questions in the training data as such problems that require person name inference. Table 4 shows the results of the 173 questions, which corresponds to the **Problem Name Inference Problems** row, and the other 714 questions, which corresponds to the **Others** row.

Table 4. Results for questions that require person name inference and others

	Rule-based	BERT-based		
		KIS1	KIS2	KIS3
Person Name Inference Problems	105/173 (0.607)	**107/173 (0.618)**	99/173 (0.572)	101/173 (0.584)
Others	440/714 (0.616)	437/714 (0.612)	441/714 (0.618)	**452/714 (0.633)**

5 Discussion

Our **KIS2** submission achieved the 1st ranked score in Task 4 (Table 1). **KIS1** and **KIS3** were also able to solve the problem with a high percentage of correct answers, indicating that the ensemble of rule-based and BERT-based parts contributed to the improvement of the correct answer rate.

In Table 2, +**Majority Vote** with five fine-tunings shows greatly better scores than **Individual**, suggesting that even when training with the same model, settings, and training data, the answers often differ and unstable. These results imply that fine-tuning results could be unstable, thus an ensemble can be effective in general. More than five times of fine-tuning trial could be a future work.

Such variations of outputs make it difficult to analyze the effectiveness of the individual methods. Adjusting the training settings or further augmenting the training data could solve this issue.

Because the results of **KIS1-3** were always better than **Baseline** in Table 2, we confirmed that our proposed method is effective. **+Ensemble** is always better than corresponding **+Majority Vote**, thus our ensemble with the **Rule-based** part was also confirmed as effective.

Statistically speaking, **KIS3** is slightly better than **KIS1**, suggesting that the person name inference was sometimes effective. **Figure 7** shows an example of a person name inference failure. In this case, as inferred from the given civil law text, *agent("代理人")* to be in A and the *principal("本人")* to be in B, but the model inferred inversely.

[Before replacement] (Civil Code Article) 代理人が自己の占有物を以後本人のために占有する意思を表示したときは、本人は、これによって占有権を取得する。 When an agent has manifested his/her intention to take possession of his/her own possessions for the principal thereafter, the principal shall thereby acquire the right of possession. (Problem text) When A sells movable property X("甲") in his possession to B and expresses his intention to take possession of X("甲") for B thereafter, B shall acquire the right of possession of X("甲"). (Aが、自己が占有する動産甲をBに売却し、甲を以後Bのために占有する旨の意思を表示したときは、Bは、甲の占有権を取得する。)
[After replacement] (Civil Code Article) same (Problem text) When the principal sells movable property X("甲") that he/she possesses to the agent and indicates his/her intention to possess X("甲") for the agent thereafter, the agent shall acquire the right of possession of X("甲"). (本人が、自己が占有する動産甲を代理人に売却し、甲を以後代理人のために占有する旨の意思を表示したときは、代理人は、甲の占有権を取得する。)

Figure 7. Example of the inversed substitution of person names (R03-08-E)

However, **KIS3** had higher scores than **KIS1** and **KIS2** in Table 2, suggesting that even replacing alphabetic letters with wrong person names is effective. Further improvement of the person name inference will contribute to the final correct answer.

KIS1_N showed further accuracy improvement over other results. This suggests that negative form generation is more effective when the negative form conversion is limited to the problems which gold standard answers are *Yes*.

In Table 3, the **BERT-based** systems show lower scores than our **Rule-based** system for the questions answered by our **Rule-based** system (**Rule-coverable Problems**), but the **BERT-based** systems show higher scores than our **Rule-based** system for the questions that are *not* answered by our **Rule-based** system (**Others**) These results suggest that the ensemble achieved our goal, which aims to compensate our **Rule-based** system by machine learning methods, regarding the problems which our **Rule-based** system is not good at. In Table 4, **KIS3**, which made the person name inference separately, does not show a very high correct answer rate in **Person Name Inference Problems**. On the other hand, the correct answer rate for **Others** questions was improved, suggesting that exclusion of **Person Name Inference Problems** in the training data may improve the general accuracy regarding the **Others** questions.

6 Conclusion and Future works

We implemented BERT-based solvers and rule-based solver, integrated them by an ensemble. We improved the correct answer ratio through data expansions and ensembles, achieved 67.89% in accuracy on the COLIEE 2022 test data, placing first among the 18 formal runs of the participating teams.

Our original intention to develop the rule-based method was to make the process white-boxed, but the rule-based method is confirmed as effective through the ensemble. Since the rule-based method can explicitly select which problems to solve, it is possible to create rule-based modules specialized for problems that other types of methods, such as BERT, are not good at, thus further improvement in accuracy can be expected by creating rule-based modules specialized for ensembles.

The results of BERT's fine-tuning varied, making it difficult to determine whether there are consistent improvements, probably due to the randomness of the initial values. On the other hand, because the rule-based method always outputs the same result for the same data input, it was easier to judge whether the improvements are stable. In order to overcome this issue, we integrated five fine-tuning results of the BERT model, confirmed that it stabilizes the results and improves its overall accuracy.

We segmented and reconstructed the texts to contain necessary information without excess or deficiency, where Sentence-BERT was used to determine which data should be included. We also augmented the text data by replacing the article number with the actual article texts, transforming the data into negative forms, and replacing the data with alphabetical characters, resulting in better accuracy scores than our baseline which does not use these data augmentations.

Our future work includes employments of other machine learning methods than BERT, ensembles of such different machine learning methods or multiple models. Another future work is to improve the performance of our person name inference.

ACKNOWLEDGMENTS

This research was partially supported by MEXT Kakenhi 00271635, Japan.

References

1. "Competition on Legal Information Extraction/Entailment (COLIEE-14) Workshop on Juris-informatics (JURISIN) 2014", 2014, http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html.
2. M.-Y. Kim, R. Goebel, K. Satoh, "COLIEE-2015: Evaluation of Legal Question Answering" in Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN2015), 2015, pp.1-11.
3. M.-Y. Kim, R. Goebel, Y. Kano, K. Satoh, "COLIEE-2016: Evaluation of the Competition on Legal Information Extraction and Entailment" in Proceedings of the Tenth International Workshop on Juris-informatics (JURISIN2016), 2016, pp.1-13.
4. Y. Kano, M.-Y. Kim, R. Goebel, K. Satoh, "Overview of COLIEE 2017" in Proceedings of the Competition on Leal Information Retrieval and Entailment Workshop (COLIEE

2017) in association with the 16th International Conference on Artificial Intelligence and Law, 2017, vol.47, pp.1-8.
5. M. Yoshioka, Y. Kano, N. Kiyota, K. Satoh, “Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018" in Proceedings of the Twelfth International Workshop on Juris-informatics (JURISIN 2018), 2018, pp.1-12.
6. R. Goebel, Y. Kano, M.-Y. Kim, J. Rabelo, K. Satoh, M. Yoshioka, “COLIEE 2019 Overview” 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, pp.1-9.
7. J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, K. Satoh, “COLIEE 2020: Methods for Legal Document Retrieval and Entailment” in Proceedings of the Fourteenth International Workshop on Juris-informatic (JURISIN 2020), 2020, pp. 1-15.
8. J. Rabelo, R. Goebel, M.-Y. Kim, Y. Kano, M. Yoshioka, K. Satoh, “Summary of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021, pp.1-7.
9. M. Fujita, N. Kiyota, Y. Kano, “Predicate’s Argument Resolver and Entity Abstraction for Legal Question Answering: KIS teams at COLIEE 2021 shared task” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021, pp.15-24.
10. M. Yoshioka, Y. Aoki , Y. Suzuki, “BERT-based Ensemble Methods with Data Augmentation for Legal Textual Entailment in COLIEE Statute Law Task” in ICAIL '21 Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, 2021, pp.278-284.
11. H.-T. Nguyen, V. Tran, N.L. Minh, M.-P. Nguyen, T.-H.-Y. Vuong, M.Q. Bui, M.-C. Nguyen, B. Dang, K. Satoh, “ParaLaw Nets - Cross-lingual Sentence-level Pretraining for Legal Text Processing” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021, pp.54-59.
12. M.-Y. Kim, J. Rabelo, R. Goebel, “BM25 and Transformer-based Legal Information Extraction and Entailment” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021, pp.25-30.
13. F.Schilder, D. Chinnappa, K. Madan, J. Harmouche, A. Vold, H. Bretz, J. Hudzina, “A Pentapus Grapples with Legal Reasoning” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021, pp.60-68.
14. S. Wehnert, S. Dureja, L. Kutty, V. Sudhi, E.W.D. Luca, “Using Contextual Word Embeddings and Graph Embeddings for Legal Textual Entailment Classification” in Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment (COLIEE 2021), 2021 pp.69-77.
15. R. Hoshino, N. Kiyota, Y. Kano, “Question Answering System for Legal Bar Examination using Predicate Argument Structures focusing on Exceptions” in Proceedings of the Sixth International Competition on Legal Information Extraction/Entailment (COLIEE 2019), 2019.
16. J. Devlin, M-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2, 2019.
17. “ cl-tohoku/bert-japanese”, https://github.com/cl-tohoku/bert-japanese.
18. “【日本語モデル付き】2020年に自然言語処理をする人にお勧めしたい文ベクトルモデル (Japanese model attached: sentence vector models recommended to developers of natural language processing in 2020)”, https://qiita.com/sonoisa/items/1df94d0a98cd4f209051 (in Japanese)

nigam@COLIEE-22: Legal Case Retrieval and Entailment using Cascading of Lexical and Semantic-based models

Shubham Kumar Nigam*[1], and Navansh Goel*[2]

[1] Indian Institute of Technology, Kanpur, India
sknigam@iitk.ac.in
[2] Vellore Institute of Technology, Chennai, India
navansh.goel@gmail.com

Abstract. This paper describes our submission to the Competition on Legal Information Extraction/Entailment 2022 (COLIEE-2022) workshop on case law competition for tasks 1 and 2. Task 1 is a legal case retrieval task, which involves reading a new case and extracting supporting cases from the provided case law corpus to support the decision. Task 2 is the legal case entailment task, which involves the identification of a paragraph from existing cases that entails the decision in a relevant case. We employed the neural models Sentence-BERT and Sent2Vec for semantic understanding and the traditional retrieval model BM25 for exact matching in both tasks. As a result, our team ("nigam") ranked 5th among all the teams in Tasks 1 and 2. Experimental results indicate that the traditional retrieval model BM25 still outperforms neural network-based models.

Keywords: BM25 · Sentence-BERT · Sent2vec · Reduced-Space.

1 Introduction

Many countries, such as India, the United States, Canada, Australia, and Africa, follow the case law system. These case law systems are legal systems that give great weight to judicial precedent, and the volume of information produced in the legal sector nowadays is overwhelming. Due to this, it takes significant efforts of legal practitioners to search for relevant cases from the database and extract the entailment parts manually with the rapid growth of digitalized legal documents. Therefore, an efficient legal assistant system to alleviate the heavy document work is necessary for Legal-AI.

The Competition on Legal Information Extraction and Entailment (COLIEE) workshop has been organized for several years to assist the legal research community. They provide the legal texts and specific problems in the legal domain such as question-answering, case law retrieval, case law entailment, statute law

* These authors contributed equally to this work

retrieval, and statute law entailment to help the research community develop Legal-AI techniques.

This paper introduces our approaches to completing two case law tasks, Task 1 (the retrieval task) and Task 2 (the entailment task). We employed both traditional retrieval models like BM25[6] for exact matching and semantic understanding like Sentence-BERT[5], Sent2Vec[4] in both task, and their combination in Task 1. In combination, we first select top-K candidates according to the BM25 rankings and afterward get the representation features of each sentence for the document we used pre-trained Sentence-BERT and Sent2Vec via comparing paragraph-level interaction. Furthermore, we used cosine similarity with the max-pooling strategy to get the final document score between query and noticed cases. Since case law references can be as small as one or two paragraphs, the max-pooling strategy gives the best possible score for a query-candidate case pair over alternative strategies like average-pooling. As a result, our team ranked 5th among all the teams in both tasks. Experimental results suggest that the exact matching using the traditional retrieval model BM25 still outperforms representation features generated by neural network-based models. We released the codes for all subtasks via GitHub[3].

2 The Task

2.1 Task 1 - Legal Case Retrieval

The main objective of legal case law retrieval is to obtain relevant supporting documents for a given query case Q. The task involves retrieving noticed cases $N = \{N1, N2, ..., N_m\} \mid N \subseteq S$ given that $S = \{S1, S2, S3, ..., S_n\}$ belonging to a given case law corpus of all possible supporting cases. The corpus predominantly consists of case law documents from the Federal Court of Canada database provided by Compass Law.

The query case documents contain suppressed reference markers instead of actual references to precedent case laws, compelling the participants to design models that can understand the text around the references and return valid noticed cases. A supporting case $S_k : S_k \in S$, is a noticed case for query $Q_j : Q_j \in Q$, *iff* Q_j contains a reference to S_k.

2.2 Task 2 - Legal Case Entailment

Task 2 comprises identifying a specific paragraph from a given supporting case that entails the decision for the query case. Thus, given a query case Q, a supporting case R and paragraphs $P = \{P1, P2, P3, ..., Pn\}$ such that $P \subseteq R$, the task requires one to correctly identify $P_i \in P$ such that it entails the decision for query case Q.

[3] https://github.com/ShubhamKumarNigam/COLIEE-22

3 Data Corpus

The data corpus for both Task 1 and Task 2 belongs to a database of case law documents from the Federal Court of Canada provided by Compass Law. Table 1 presents the dataset statistics for both the tasks. For Task 1, 898 query cases were given against 3531 candidate cases as a part of the training data corpus. The test dataset had 300 query cases against 1263 candidate cases. For Task 2, 525 query cases were provided for training against 18,740 paragraphs. There were 100 query cases against 3278 candidate paragraphs as part of the testing dataset. The organizers provided the candidate paragraphs for a given query case separately. On average, there are 35.627 candidate paragraphs for each query case in the training dataset and 32.455 candidate paragraphs for each query case in the testing dataset.

On further analysis, we found that the average number of relevant candidate cases for Task 1 was 4.684 cases for the training dataset and 4.21 relevant cases for the test dataset. As a part of our methodology, we predict the top 5 possible noticed cases for each query case based on these numbers. The average number of relevant paragraphs for Task 2 was a little over 1 paragraph, 1.14 paragraphs for training and 1.18 paragraphs for testing. These numbers show that most documents for Task 2 had their decisions entailed inside a single paragraph. The table also shows the average length of each document. The average number of words is higher in the testing query dataset than in the training query dataset for Task 1.

On the contrary, the average number of words per candidate document decreases from the training data to the testing data. The opposite trend is observed for Task 2, where the average number of words per query document decreases from training data to testing data. In contrast, the average number of words per candidate paragraph increases from training to testing.

Table 1. Statistics of the training and test data for Task1 and Task2

	Task 1		**Task 2**	
	Train	**Test**	**Train**	**Test**
# of queries	898	300	525	100
# of candidate cases/paragraphs	3,531	1,263	18,740	3,278
avg # of candidate paragraphs per query	-	-	35.627	32.455
avg # of relevant candidates/paragraphs	4.684	4.210	1.14	1.18
avg query length (words)	28,866.73	32,461.27	24,966.55	22,326.95
avg candidate length (words)	30,113.09	29,756.87	636.08	643.69

3.1 Preprocessing

Since the given documents are very long and represent the entire document singly is not efficient. Also, it is often noted that case judgments cite multiple cases and write multiple sentences about them. So we segment the document into

meaningful sentences using the spacy library[4] and retrieve relevant contents using RegEx, which contains the citation marker "FRAGMENT_SUPPRESSED," say i, k previous $i-k$ and k following sentences $i+k$ which can say context around the citation. These sentences feed the models to make a corpus or generate distributed representations.

3.2 Evaluation Metrics

The evaluation metrics of Tasks 1 and 2 are precision, recall, and F-measure. All the metrics are micro-average, which means the evaluation measure is calculated using the results of all queries. Definition of these measures are as follows:

$$Precision = \frac{\#\text{ correctly retrieved cases(paragraphs) for all queries}}{\#\text{ retrieved cases(paragraphs) for all queries}} \tag{1}$$

$$Recall = \frac{\#\text{ correctly retrieved cases(paragraphs) for all queries}}{\#\text{ relevant cases(paragraphs) for all queries}} \tag{2}$$

$$F - measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

4 Our Methods

4.1 BM25

In previous COILEE submissions, many participants such as Kim et al. [2], Ma et al. [3], Rosa et al. [7], Shao et al. [8] and Shao et al. [9] in IJCAI-20 concluded that traditional bag-of-words IR models have competitive performances in legal case retrieval tasks. So our first preference is to try the exact matching-based BM25 model [6], which is a probabilistic relevance model based on a bag-of-words approach. The score of a document D given a query Q which contains the words $q_1, ..., q_n$ is given by:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \tag{4}$$

where $f(q_i, D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters.

$$\text{IDF}(q_i) = ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1) \tag{5}$$

[4] https://spacy.io/usage/linguistic-features#sbd

where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i[5].

We tried BM25 using Rank-BM25 library[6], where the algorithms are taken from the paper Trotman et al. [10]. Although we are not getting good results, we tried to implement Okapi BM25 with sklearn's TfidfVectorizer[7] which converts a collection of raw documents to a matrix of TF-IDF features. TfidfVectorizer has valuable features; some important feature descriptions are:

- **stop_words:** Either we can pass string 'english' or a list containing stop words, all of which will be removed from the resulting tokens.
- **ngram_range:** The lower and upper boundary of the range of n-values for different n-grams to be extracted. For example, an ngram_range of (1, 1) means only unigrams, and (1, 2) means unigrams and bigrams.
- **max_df:** While building the vocabulary, ignore terms with a document frequency strictly higher than the given threshold (corpus-specific stop words). The range is [0.0, 1.0].
- **min_df:** While building the vocabulary, ignore terms with a document frequency strictly lower than the given threshold. This value is also called a cut-off in the literature. The range is [0.0, 1.0].
- **norm:** Each output row will have a unit norm, either:
  - **'l2':** The sum of squares of vector elements is 1. The cosine similarity between two vectors is their dot product when the l2 norm has been applied.
  - **'l1':** Sum of absolute values of vector elements is 1.

These features are beneficial for BM25 to extract better features, and even we can restrain these features. Our results depict that after employing TfidfVectorizer, BM25 performed much better.

4.2 Sent2Vec

Sent2Vec is an Unsupervised Learning of Sentence Embeddings or distributed representations of sentences using Compositional N-Gram Features. It can be thought of as an extension of word2vec (CBOW) to sentences. Sentence embedding is the average of the source word embeddings of its constituent words. We generate features from the pre-trained model "sent2vec_wiki_bigrams" from Sent2vec[8] library and paper Pagliardini et al. [4], which is trained on the English Wikipedia dataset and embedding dimensions are 700. Now to find the most similar sentences between query case and candidate case in a dataset, we compared using cosine-similarity[9] using max-pool strategy. Finally, we used the max-pooling strategy between query and noticed cases to get the most similar document.

[5] https://en.wikipedia.org/wiki/Okapi_BM25
[6] https://github.com/dorianbrown/rank_bm25
[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[8] https://github.com/epfml/sent2vec
[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

4.3 Sentence-BERT

Since we need sentence/paragraph representation, instead of using any transformer and generating sentence embedding using a pooling strategy, it will be more suitable to directly use a model to derive semantically meaningful representations for sentences. We use the Sentence-BERT[10] framework based on the paper [5], which provides an easy method to compute dense vector representations for sentences and paragraphs. This model is based on transformer networks like BERT/RoBERTa, but this modification of the pre-trained BERT network uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. We generate dense vectors from the pre-trained model "all-mpnet-base-v2" from Sentence-BERT[11] library, which is trained on a large and diverse dataset of over 1 billion training pairs, max sequence length 384, mean-pooling, and generates 768 dimensions. To find the most similar paragraphs between query and candidate cases in a dataset, we compared using cosine-similarity using the max-pool strategy. Finally, we used the max-pooling strategy between query and noticed cases to get the most similar document.

4.4 Reduced-Space

As illustrated in Figure 1, we first represent the query corpus in paragraphs $Q = (Q_{p1}, Q_{p2}, ..., Q_{pN})$ by extracting the sentences which contain the "FRAGMENT_SUPPRESSED" marker say i, k previous $i - k$ and k following sentences $i + k$ and citation corpus by segmentation of logical paragraphs $C = (C_{p1}, C_{p2}, ..., C_{pM})$. In total, we obtain $N * M$ pairs of paragraphs. Then, we used a lexical model to calculate the similarity between a given query and the entire case law corpus on paragraph-wise segmented documents. We limit the searching space from 3531 to 100 candidate cases by picking the top 100 cases with the highest BM25 scores for a given query. After reducing the searching space to 100 documents, we want to extract the semantic relationship between the query and candidate paragraphs pairs, e.g., $(Q_{pi}; C_{pj})$. To obtain this relationship score, we use the pre-trained model Sentence-BERT and Sent2vec. We used the max-pooling strategy to find the most similar paragraphs and documents in the same approach as above.

Reason for Max-pooling:
The relevant candidate documents carry essential information related to the query case, but the amount of information is often disputable. There are many candidate cases wherein only one or two paragraphs' worth of relevant information is present in the context of the query case. For such documents, taking alternative pooling strategies such as average pooling can result in a decreased similarity score. With a decrease in the score, the model can find it difficult to identify relevant candidate cases from cases containing general information related to case

[10] https://www.sbert.net/
[11] https://www.sbert.net/docs/pretrained_models.html

laws. We consider a 2D max-pooling strategy that allows the model to extract the most similar documents for a given query case to solve this issue.

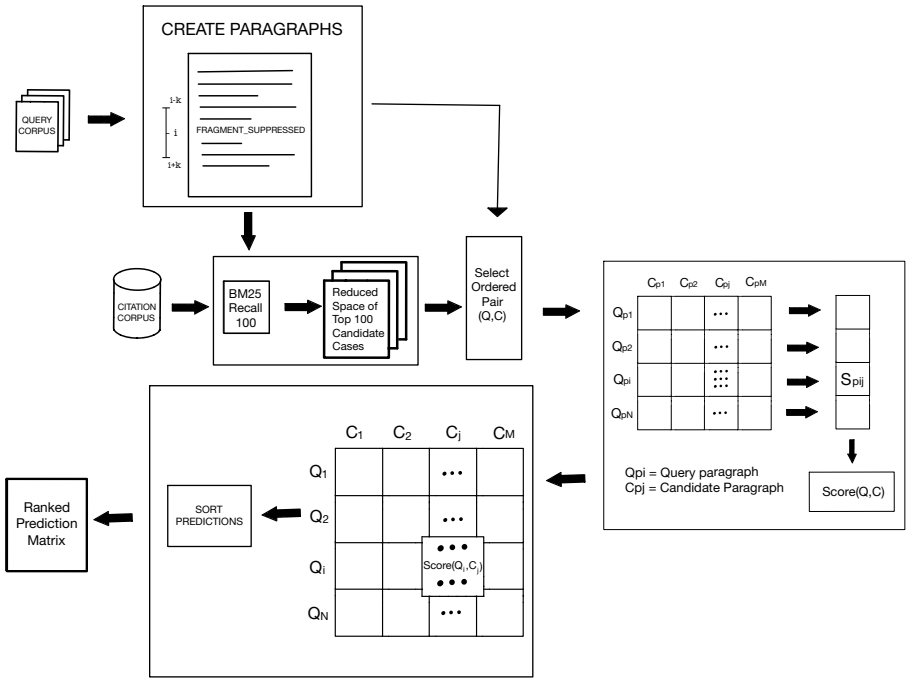


Fig. 1. An illustration of Reduced-Space Architecture.

5 Results and Analysis

5.1 Task-1

We submitted three official runs in table 2, run-1 is BM25, in a cascaded manner reduced space after using BM25 Recall@100, run-2 is pre-trained Sentence-BERT, and run-3 is pre-trained Sent2vec. All these official runs have done experiments on the paragraph level comparison, and every query case predicts a top-5 citation for both training and test dataset because the average number of relevant cases is approximately 5 in training data. We get the top score in using the traditional model BM25 for exact matching; on the other hand, in semantic understanding models, Sentence-BERT gets second, and Sent2vec gets third. Our team ("nigam") ranked 5th among all the teams in Tasks 1, and the results indicate that the traditional retrieval model BM25 still outperforms neural network-based models.

Apart from that, we tried several experiments without reduced space means the search space was the entire citation corpus in Sent2vec and Sentence-BERT.

Similar behavior can be seen in table 3 for the training dataset, where BM25 performs better than Sentence-BERT and Sent2vec in reduced space. Along with experiments on the paragraph level comparison, we also tried document level comparison, but they did not give good results. We even included results without reduced space, and it can easily be observed that the reduced space results are slightly improved. Training and testing for every query case predict top-5 case law because the average number of relevant cases is approximately 5. We removed stop-words and experimented with a range of values in max_df, min_df, b, $k1$, and $ngram\_range$ parameters while keeping other hyperparameters with their default values. We achieved the best F1 Score under this setting, i.e. $\{max\_df = 0.90, min\_df = 1, b = 0.99, k1 = 1.6, ngram\_range = (2, 6)\}$. The default values worked the best for other hyperparameters, and experimenting with them did not add value to the result.

Table 2. Results on the test set of Task 1

Team	F1 Score	Precision	Recall
UA	0.3715	0.4111	0.3389
UA	0.371	0.4967	0.2961
siat	0.3691	0.3005	0.4782
siat	0.368	0.3026	0.4695
UA	0.3559	0.363	0.3492
siat	0.2964	0.2522	0.3595
LeiBi	0.2923	0.3	0.285
LeiBi	0.2917	0.2687	0.3191
JNLP	0.2813	0.3211	0.2502
nigam	**0.2809**	**0.2587**	**0.3072**
JNLP	0.2781	0.3144	0.2494
DSSR	0.2657	0.2447	0.2906
JNLP	0.2639	0.2446	0.2866
DSSR	0.2461	0.2267	0.2692
TUWBR	0.2367	0.1895	0.3151
LeiBi	0.2306	0.2367	0.2249
TUWBR	0.2206	0.1683	0.3199
nigam	**0.1542**	**0.142**	**0.1686**
nigam	**0.1484**	**0.1367**	**0.1623**
DSSR	0.1317	0.1213	0.1441

Table 3. Task 1 results in the training set. Every query case predicts top-5 case law because the average number of relevant cases is approximately 5. We remove stop-words, max_df=0.90, min_df=1, b=0.99, k1=1.6, ngram_range=(2,6) in BM25. Moreover, other parameters are set to default for all experiments.

Model	F1 Score	Precision	Recall
BM25	**0.1957**	**0.1895**	**0.2023**
Sent2vec	0.0812	0.0786	0.0839
SBERT	0.0844	0.0817	0.0873
BM25 + Sent2vec (Reduced Space)	0.1080	0.0961	0.1234
BM25 + SBERT (Reduced Space)	0.1017	0.0984	0.1051

5.2 Task-2

We submitted two official runs in table 4; both runs are based on the BM25 model. Run-1 uses only entailed fragment text as a query, and run-2 uses the filtered base case such that search entailed fragment text into the base case then input to the model as searched results along with previous and following sentences to capture the other relevant information. Every query case predicts one case law paragraph because the average number of relevant paragraphs is approximately 1.14 in the training dataset. Our team ("nigam") ranked 5th among all the teams in Tasks 2, and the results indicate that using only fragment text as the query will give better scores. Although, we got the highest Recall score among all participants.

Similar behavior can be seen in table 5 for the training dataset, where BM25 performs better when only fragment text as the query and outputs only a single paragraph. Apart from that, we also tried two paragraphs prediction for every query; indeed, that gives a better Recall value, but on the other hand, precision values drop significantly. We also tried experiments where along with fragment text, we retrieved previous and following sentences from the base case after matching fragment text to capture the other relevant information. For example, $[-i, +j]$ means i previous and j following sentences from the matched fragment sentences. However, the results show that results drop as we include the other information. We kept stop-words and experimented with a range of values in max_df, min_df, b, $k1$, and $ngram\_range$ parameters while keeping other hyper-parameters with their default values. We achieved the best F1 Score under this setting, i.e. $\{max\_df = 0.65, min\_df = 1, b = 0.7, k1 = 1.6, ngram\_range = (1, 1)\}$. The default values worked the best for other hyperparameters, and experimenting with them did not add value to the result. One thing that can be noticed here is that uni-gram works better because, in Task-2, the length of entailed fragment text is small compared to the query case in Task-1.

Table 4. Results on the test set of Task 2

Team	F1 Score	Precision	Recall
NM	0.6783	0.6964	0.661
NM	0.6757	0.7212	0.6356
JNLP	0.6694	0.6532	0.6864
JNLP	0.6612	0.6452	0.678
jljy	0.6514	0.71	0.6017
jljy	0.6514	0.71	0.6017
JNLP	0.6452	0.6154	0.678
jljy	0.633	0.69	0.5847
NM	0.6325	0.6379	0.6271
UA	0.5446	0.6105	0.4915
UA	0.5363	0.7869	0.4068
UA	0.4121	0.3049	0.6356
nigam	**0.3204**	**0.198**	**0.839**
nigam	**0.2104**	**0.13**	**0.5508**

Table 5. Task 2 results in the training set. Every query case predicts either 1 or 2 case law paragraphs because the average number of relevant paragraphs is approximately 1.14. We kept stop-words, max_df=0.65, min_df=1, b=0.7, k1=1.6, ngram_range=(1,1). Moreover, other parameters are set to default for all experiments in BM25. Query cases are input such that either only entailed fragment text or search that text into the base case with previous and following sentences.

Query case Input	# Prediction per query	F1 Score	Precision	Recall
consider only Entailed Fragment text	**1**	**0.6228**	**0.6667**	**0.5843**
	2	0.5106	0.4010	0.7028
using base case [-1,+1]	1	0.4359	0.4667	0.4090
	2	0.4172	0.3276	0.5743
using base case [-1,+2]	1	0.4110	0.4400	0.3856
	2	0.3857	0.3029	0.5309
using base case [-1,+3]	1	0.3790	0.4057	0.3556
	2	0.3663	0.2876	0.5042
using base case [-2,+3]	1	0.3594	0.3848	0.3372
	2	0.3469	0.2724	0.4775
using base case [-3,+3]	1	0.3488	0.3733	0.3272
	2	0.3457	0.2714	0.4758

6 Conclusion

We participated at COLIEE 2022 in Task 1 and Task 2, which allowed exploring information retrieval challenges in the legal case law retrieval and legal entailment. Our main objective was to combine traditional lexical retrieval models BM25 with dense passage retrieval models Sentence-BERT and Sent2vec for re-ranking the results. We show that the paragraph-level retrieval in the first stage outperforms the document-level retrieval. Also, overall ranking after combining lexical and semantic models improves the ranking of dense passage retrieval models alone for task-1. Although usually, neural network-based models perform best in the NLP tasks, especially transformers. Nevertheless, our results and previous COLIEE participants show that these pre-trained models do not work well in a specialized domain; traditional lexical retrieval models still perform better.

Furthermore, in task-2, using only fragment text as the query will give better scores, whereas accommodating other information from the base case only impairs the accuracy. Since we got the highest Recall score among all participants, we should also try the reduced space approach. In the future, we plan to investigate dense retrieval models domain-specific contextualized language models like LegalBERT [1] and graph neural networks [11].

References

1. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP

2020. pp. 2898–2904. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261, https://aclanthology.org/2020.findings-emnlp.261
2. Kim, M.Y., Rabelo, J., Okeke, K., Goebel, R.: Legal information retrieval and entailment based on bm25, transformer and semantic thesaurus methods. The Review of Socionetwork Strategies pp. 1–18 (2022)
3. Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., Ma, S.: Retrieving legal cases from a large-scale candidate corpus (2021)
4. Pagliardini, M., Gupta, P., Jaggi, M.: Unsupervised learning of sentence embeddings using compositional n-gram features. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 528–540. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1049, https://aclanthology.org/N18-1049
5. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410, https://aclanthology.org/D19-1410
6. 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)
7. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: Yes, bm25 is a strong baseline for legal case retrieval. arXiv preprint arXiv:2105.05686 (2021)
8. 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. arXiv preprint arXiv:2012.13102 (2020)
9. 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. pp. 3501–3507 (2020)
10. Trotman, A., Puurula, A., Burgess, B.: Improvements to bm25 and language models examined. In: Proceedings of the 2014 Australasian Document Computing Symposium. pp. 58–65 (2014)
11. Yang, J., Ma, W., Zhang, M., Zhou, X., Liu, Y., Ma, S.: Legalgnn: Legal information enhanced graph neural network for recommendation. ACM Transactions on Information Systems (TOIS) **40**(2), 1–29 (2021)

DoSSIER@COLIEE2022: Dense retrieval and neural re-ranking for legal case retrieval

Amin Abolghasemi[1] ⋆, Sophia Althammer[2] ⋆, Allan Hanbury[2], and Suzan Verberne[1]

[1] Leiden University, The Netherlands
m.a.abolghasemi,s.verberne@liacs.leidenuniv.nl
[2] Institute for Information Systems Engineering, TU Wien, Vienna, Austria
name.surname@tuwien.ac.at

Abstract. We present our approaches for the case law retrieval task (Task 1) in the Competition on Legal Information Extraction/Entailment (COLIEE) 2022. We investigate lexical and dense first stage retrieval methods aiming for a high recall in the initial retrieval. Furthermore we compare shallow neural re-ranking with the MTFT-BERT model to the BERT-PLI model and investigate which part of the text of a legal case should be taken into account for re-ranking. We find that while the MTFT-BERT model reaches the highest precision for legal case retrieval on the COLIEE 2021 test set, it is less effective than optimized BM25 on the COLIEE 2022 test set. In fact, our results show that BM25 shows a consistent high effectiveness across different test collections in comparison to the neural re-ranking models.

Keywords: Dense retrieval · BERT-based ranking · Case law retrieval.

1 Introduction

In case law systems, legal cases are a primary source for decision making for a new case and therefore it is crucial for legal practitioners to find the precedent cases that entail the decision of a new case [26]. As the amount of digitized legal information is exponentially growing [27], it costs legal professionals increasingly more effort to retrieve the cases which are relevant to their case. Legal professionals require a search system that finds all relevant cases [2] and simultaneously has a high precision, as legal practitioners will only examine the top retrieved results [24].

We address these challenges of legal case retrieval in Task 1 of the Competition on Legal Information Extraction/Entailment (COLIEE) 2022 by combining high-recall first stage retrieval solutions with precision-oriented neural re-ranking models. In the web domain, neural retrieval models which are based on the contextualization with the pre-trained language model BERT [9] have demonstrated large effectiveness gains for first stage retrieval [11, 13, 15] as well as for neural re-ranking [12, 20, 21]. It is our goal to investigate these benefits of neural retrieval and ranking models for the task of legal case retrieval.

However, the task of legal case retrieval holds challenges for neural retrieval models, different to the web domain: i) the relevance of a legal case is not only determined

⋆ Both authors contributed equally to this research.

on the document-level but also on the paragraph-level [27], ii) legal practitioners require a high-recall and high-precision solution [24], and iii) legal documents tend to be long [25, 27]. For example in the COLIEE 2022 test set the query documents and the documents in the collection have more than $4,000$ words. In order to attain a high recall, and precision solution, we approach the legal case retrieval task with a first stage retrieval step, with the goal of a high recall and a second neural re-ranking step in order to improve the precision at high ranks. For the first stage retrieval we compare lexical matching retrieval models (BM25[22]) on the validation set with the dense retrieval model PARM [5], which was proposed for the document-to-document retrieval task of legal case retrieval. PARM proposes paragraph-level retrieval for the first stage retrieval and thereby overcomes the limited input length for the contextualization with BERT. It reaches high recall results especially at higher ranks. We optimize BM25 with respect to the input parameters in order to improve the retrieval scoring for first stage retrieval. We investigate:

RQ1 Which first stage retrieval method is most effective in terms of recall for case law retrieval?

We compare the lexical and dense retrieval models on the COLIEE 2021 test set and find that the optimized BM25 leads to the highest recall values ranging from the cut-offs 5 to 100. We find that the paragraph-level retrieval of PARM leads to performance gains compared to the BM25 retrieval on the whole document, however the optimization of BM25 on the validation set shows the highest recall for the first stage retrieval.

As second stage we compare different neural re-rankers that all use the contextualization of BERT [9]. We compare the optimized BM25 ranking with a cross-encoder BERT and with the recently proposed MTFT-BERT model[1], which is trained on multi-task optimization for re-ranking of legal case retrieval results and reaches current state-of-the-art results on the COLIEE 2021 Task 1. Due to the limited input length of BERT, the MTFT-BERT model does not take the whole text of the query and candidate document for relevance estimation into account. Therefore we also investigate the re-ranking with the BERT-PLI model, which re-ranks the query and candidate document on a paragraph-level and aggregates the paragraph-interactions with a recurrent neural network. Furthermore we analyze the performance of MTFT-BERT with re-ranking on generated summaries of the legal cases, which also tackles the problem of limiting the input length of the legal cases[6]. We investigate:

RQ2 Do neural re-ranking models improve upon a lexical ranking model for legal case retrieval?

We compare the lexical ranking of optimized BM25 with the neural re-ranking of MTFT-BERT and BERT-PLI on the test set of COLIEE 2021 Task 1. As for the final ranking the recall and precision are both crucial measurements of retrieval quality, we compare the re-ranking models in terms of F1-score. We find that the neural re-ranking with the MTFT-BERT model, the BERT-PLI model and the summary-based BERT re-ranking does not improve the overall retrieval effectiveness.

As the input length for text is limited by the BERT architecture and the long case law documents exceed this input length, we investigate which text of the document the

neural re-ranking model should take into account for the highest re-ranking effectiveness in terms of precision and recall:

RQ3 What effect does the content of the document taken into account for re-ranking have on the retrieval quality?

With the three different BERT-based re-ranking strategies we compare re-ranking with the contextualization based on the first paragraph truncated at the input length of BERT for MTFT-BERT, re-ranking based on the summaries of the legal cases for MTFT-BERT and the whole document contextualized on the paragraph-level for BERT-PLI. Our experimental results show that the MTFT-BERT model with only the first paragraph leads to the highest retrieval quality for COLIEE 2021 and is on par with BERT-PLI for COLIEE 2022.

As the performance of the re-ranking models also depends on the re-ranking depth, we furthermore investigate:

RQ4 What effect does the re-ranking depth for different re-ranking models have on retrieval quality?

We investigate the retrieval effectiveness in terms of F1-score for the different neural re-ranking models and find that MTFT-BERT model achieves the highest retrieval quality at a re-ranking depth of 15, whereas BERT-PLI reaches the highest performance at re-ranking at 50. To summarize, our contributions are the following:

- We compare lexical and dense retrieval models for the first stage retrieval of legal case retrieval aiming for a high recall
- We investigate different neural re-ranking models and find that BM25 shows a consistent high effectiveness across different test collections in comparison to the neural re-ranking models
- We investigate which content of the document to take into account for effective and efficient neural re-ranking and find that the first paragraph contains sufficient information for an effective neural re-ranking model

2 Task overview

We introduce the retrieval task and give an overview of the datasets used, the evaluation metrics, as well as the related work.

2.1 Task description

Legal Case Retrieval is the task of retrieving relevant cases which could support the decision of a query case. These relevant cases are all supporting (or noticed) cases. Legal case retrieval is crucial for legal practitioners as it is time-consuming to manually find the relevant cases due to the large number of cases in the corpus.

Table 1. Data Statistics for the Task 1 of COLIEE 2021 and COLIEE 2022

	Train	**Test**
COLIEE 2021 #Query Cases	650	250
COLIEE 2021 #Candidate Cases	4415	4415
COLIEE 2021 #Average Relevant Cases	5.1	3.6
COLIEE 2022 #Query Cases	-	300
COLIEE 2022 #Candidate Cases	-	1563

2.2 Datasets

In COLIEE 2022 Task 1, the corpus consists of Federal Court of Canada case laws. Table 1 shows the statistics of both COLIEE 2021 and COLIEE 2022 data for Task 1. Since the test and the train data from COLIEE 2021 Task 1 is used as training data for COLIEE 2022, we also include the statistics of this data as well.

2.3 Evaluation Metrics

The evaluation metrics for legal case retrieval task in COLIEE are micro-average Precision, Recall, and F1-score which are defined as follows:

$$Precision = \frac{\#TP}{\#TP + \#FP} \tag{1}$$

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

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

2.4 Related Work

In COLIEE 2021, word-based ranking models showed competitive results in the legal case retrieval task.

NeuralMind [23] achieved the second place in the legal case retrieval task of COLIEE 2021 employing segmentation of the query and candidate documents and use the maximum relevance score of BM25 of all segmentation pairs of the query and document.

The JNLP team [19] applied an interpolation between the relevance score from two models: a lexical model and a semantic model. As their lexical model, they use BM25 to retrieve top candidate documents, and for the semantic model, they use transformers and fine-tune them on two tasks: a) predicting if a sentence supports another sentence, b) if a paragraph supports another paragraph. Finally, they use an aggregation of the relevance scores provided by these two models to provide the final relevance score.

The SIAT team [16] applied a two-stage retrieval pipeline. As the first-stage model, they apply BM25 to retrieve top candidate documents, followed by the second stage

in which they use BERT-based ranking to select the final supporting cases. For both query and candidate documents, they segment the documents into paragraphs. Then, each paragraph pair, including one paragraph from the query and one from a candidate document, is used as the input to a cross-encoder BERT, and the output vector of the $[CLS]$ token is extracted to build the matrix of the interaction vectors. Next, they use a max-pooling operation followed by an LSTM [10] to predict the final relevance score.

TLIR [18], which is the winner of the COLIEE 2021 legal case retrieval task, uses Language Modeling-based retrieval with Jelinek-Mercer smoothing in their winning run. It should be noted that they use paragraphs with placeholders such as "FRAGMENT_SUPPRESSED" and "REFERENCE_SUPPRESSED", which are indications that show the paragraph is directly relevant to a notice case. TLIR also applied BERT-PLI [25] with some modifications. As BERT-PLI was shown to be effective for legal case retrieval [25], we were interested in the ranking quality of BERT-PLI with a very shallow re-ranking depth on top of an effective first-stage retrieval.

The DoSSIER team [3] which achieved the third place in COLIEE 2021 Task 1, used a transformer-based multi-stage pipeline. To do so, they applied dense retrieval with BERT (and also BM25) as their first stage retrieval model, which was then followed by neural re-ranking with BERT. However, due to the maximum input length of BERT, they applied case summarization to fit the documents to the fine-tuned BERT re-ranker.

3 Methodology

We introduce the models that we employ for first stage retrieval and re-ranking of the retrieved results.

3.1 First stage retrieval

For first stage retrieval we employ three different models, namely BM25 [22], BM25 with optimized parameters, and the Paragraph Aggregration Retrieval Model PARM [5].

BM25 BM25 is a term-based retrieval model which assigns the relevance score between the query and the documents as the following:[3]

$$s(q,d) = \sum_{i}^{n} IDF(q_i) \frac{f(q_i, D) * (k_1 + 1)}{f(q_i, D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})} \tag{4}$$

Here, $s(q,d)$ indicates the relevance score, $IDF(q_i)$ indicates the inverse document frequency of ith query term, $f(q_i, D)$ stands for the frequency of ith query term in the document D, and $|D|$ shows the length of the document D. k_1 and b are parameters of the model for which we also use optimized values in addition to the default values. Accordingly, $\textbf{BM25}_{opt}$ refers to a BM25 with optimized k_1, and b parameters.

[3] We use the implementation by Elasticsearch; nevertheless, it should be noted that there are various BM25 versions [14]

PARM The Paragraph Aggregation Retrieval Model (PARM) [5] is proposed for high-recall retrieval for document-to-document retrieval tasks and performs retrieval on the paragraph-level: for each query paragraph, relevant documents are retrieved based on their paragraphs. This paragraph-level retrieval can be done with lexical retrieval using BM25 [22] or a dense passage retrieval model (DPR) [15, 11], which uses a BERT-based [9] encoder for the query and document paragraphs. With the paragraph-level retrieval PARM liberates DPR models from their limited input length and furthermore increases the interpretability of the retrieval results. After paragraph-level retrieval the relevant results per query paragraph are aggregated into one ranked list for the whole query document. For the aggregation of dense retrieval results a vector-based aggregation with reciprocal rank fusion (VRRF) weighting is proposed, for the aggregation of the lexical retrieval results the reciprocal rank fusion (RRF) [8] is employed.

3.2 Re-ranking models

As re-ranking models we evaluate the re-ranking models BERT [6], MTFT-BERT [1] and BERT-PLI [25], which all rely on the contextualization with BERT [9].

BERT We apply BERT as a cross-encoder in our BERT-based ranking. The cross-encoder BERT ranker proposed by Nogueira and Cho[20], uses the concatenation of query and document as its input and predicts the relevance score using a fully-connected (FC) layer on top of the final hidden state of the BERT [CLS] token. Therefore, with W_{FC} as the weights of the fully-connected layer, the relevance score between query q and the document d is:

$$s(q,d) \; = \; BERT([CLS]\; q\; [SEP]\; d\; [SEP])_{[CLS]} * \; W_{FC} \tag{5}$$

MTFT-BERT MTFT-BERT has the same architecture as a cross-encoder BERT re-ranker. The difference between BERT and MTFT-BERT re-rankers is in their fine-tuning procedure where MTFT-BERT uses a multi-task optimization. To be specific, the fine-tuning of this model incorporates representation learning on query and document as an additional objective besides a pairwise ranking objective [1]. We employ the MTFT-BERT using the first passage of the query and candidate documents or using a summary of the query and the document, generated by a Transformer-model trained on generating the summaries of the legla documents. In Figure 1, the architecture of the MTFT-BERT model is visualized using either using the first passage or the summaries for re-ranking.

BERT-PLI Shao et al. [25] propose the BERT-PLI framework which models the Paragraph-Level Interactions with the language model BERT for re-ranking of case law retrieval documents. We visualize the workflow of BERT-PLI in Figure 1. Due to the limited input length of BERT, the interaction between the query and candidate document is modelled per contextualization of each pair of query and candidate paragraphs using the BERT model. This leads to an interaction matrix with the entries of the interaction

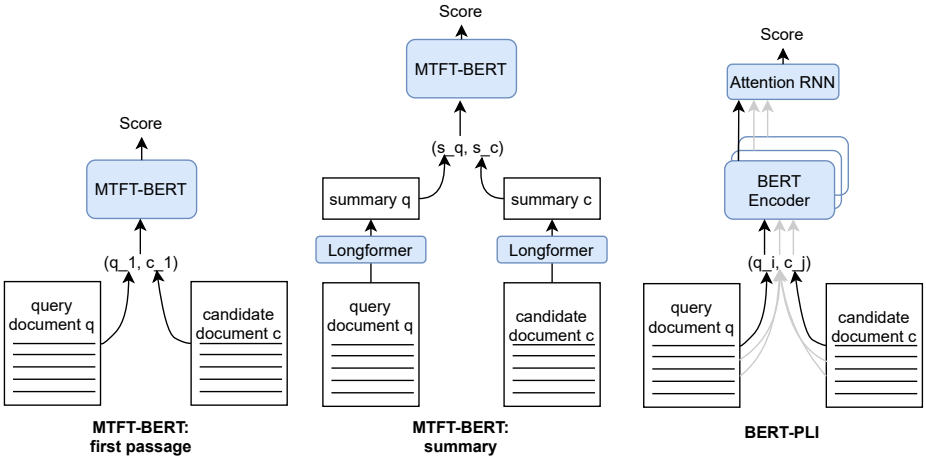


Fig. 1. Neural re-ranking architectures for case law retrieval.

for each query and candidate paragraph pair. This interaction matrix is then used as an input to an attention-based Recurrent Neural Network (RNN) or an attention-based Long-Short Term Network (LSTM), which is trained on predicting the relevance between the query and candidate document. The BERT-PLI model has the advantage that it takes the text of the whole query and the whole candidate into account for contextualization.

4 Experiments

We report on the experiments that we conducted using the different retrieval models.

4.1 First-stage Retrieval

BM25 For the BM25 implementations, we use Elasticsearch[4] to index and retrieve the documents. Following the effectiveness of term selection using Kullback-Leibler divergence for Informativeness (KLI) in case law retrieval [6, 17, 28], we apply query generation with KLI and to this aim, we use the top-10% of a query document terms scored with KLI as the query for BM25.

PARM For PARM we do retrieval with BM25 and RRF aggregation as well as retrieval with a DPR model and VRRF aggregation as proposed by Althammer et al. [5]. For BM25 we use the Elasticsearch[5] implementation with default values, for dense retrieval we use the published pretrained DPR models.[6] We leverage the DPR model trained on

[4] https://github.com/elastic/elasticsearch
[5] https://www.elastic.co/de/elasticsearch
[6] https://github.com/sophiaalthammer/parm

the paragraph-level and the document-level case law retrieval training collections from COLIEE 2021 from Task 1 and Task 2, which is denoted in [5] as "LegalBERT doc". The documents are split into paragraphs according to their structure into an introductory part, the summary and the single claims as one paragraph respectively.

4.2 Re-ranking

BERT and MTFT-BERT For BERT and MTFT-BERT, we follow the work of Abolghasemi et al. [1] and use the published code[7]. Therefore, as the initial checkpoint for the encoder of both BERT and MTFT-BERT re-rankers, we use LegalBERT [7], which is a domain-specific BERT model pre-trained on the legal domain. Due to the maximum input length limitation in transformer-based language models, we use BERT and MTFT-BERT in two different setups as follows: i) using the first passage of the documents and ii) using the summary of the cases as the documents. While using the first passage as a representative of the documents is naive, prior work shows that by using a *shallow re-ranking depth*, the BERT re-ranker can improve on the results of initial rankers which do the retrieval based on the whole documents [1]. However, the experimental settings of COLIEE 2021 and COLIEE 2022 are different as the collection of the COLIEE 2022 test set differs from the collection in COLIEE 2021, which we use for training and validation of the models.

BERT-PLI For the BERT-PLI implementation we use the reproduced work of Althammer et al. [4]. They show that training the contextualization model BERT on paragraph-level modeling before inferencing for re-ranking does not bring performance gains in the re-ranking, therefore we use the BERT-base-uncased model[8] for contextualization of the query and candidate paragraph pairs. The documents are split into paragraphs according to their structure, generating an introductory paragraph, the summary and the single claims as one paragraph each.

As the BERT-PLI model includes an attention-based recurrent neural network for predicting the relevance between query and candidate document, we train an attention-based Gated-Recurrent Network (AttenGRU) as well as an attention-based LSTM model on the training collection of COLIEE 2021 Task 1. We train the recurrent neural networks on the top-15 of the optimized BM25 first stage retrieval results for the training collection of COLIEE 2021 Task 1. For training set we augment the top-15 retrieval results with the relevant documents which are not included in the top-15 results. Furthermore we investigate training the attention-based RNNs with a balanced training set with all positive documents to the query case and with the same amount of negative documents, sampled randomly from the top-15 first stage retrieval results of optimized BM25. We also use the published AttenGRU and AttenLSTM[9] models for inference, which are trained on the COLIEE 2019 Task 1. Due to the computational complexity of BERT-PLI, we evaluate the different BERT-PLI models on the top-50 of our validation set (COLIEE 2021 Task 1 test). We find that the BERT-PLI model with the AttenLSTM

[7] https://github.com/aminvenv/mtft
[8] https://github.com/google-research/bert
[9] https://github.com/sophiaalthammer/bert-pli

trained on the COLIEE 2019 Task 1 data leads to the highest F1-score. Furthermore we evaluate the F1-score on the validation set for different re-ranking depths and cut-off values and see that a re-ranking depth of 50 and the cut-off of 5 leads the highest F1-score.

5 Results

We evaluate our retrieval results both on the COLIEE 2021 Task 1 as well as the COLIEE 2022 task1 test sets.

5.1 RQ1: First Stage Retrieval Effectiveness

We compare our baseline of BM25 to the optimized BM25 as well as to paragraph-level retrieval with PARM with lexical retrieval (BM25) or dense retrieval (DPR). In Figure 2 the first stage retrieval recall at different cut-offs from 5 to 100 is visualized for BM25, optimized BM25 (BM25 opt) and PARM with BM25 retrieval and RRF aggregation (BM25 PARM-RRF) as well as PARM with DPR retrieval with VRRF aggregation (DPR PARM-VRRF). The recall is here measured on the COLIEE 2021 Task 1 test set. Overall we can see that optimized BM25 reaches the highest recall at all cut-offs, furthermore we see that paragraph-level retrieval with PARM improves the recall at cut-off 40 for lexical retrieval and at cut-off 90 for dense retrieval.

We also denote the exact retrieval recall results of the different approaches in Figure 3 for the COLIEE 2021 Test set as well as for the COLIEE 2022 Test set.

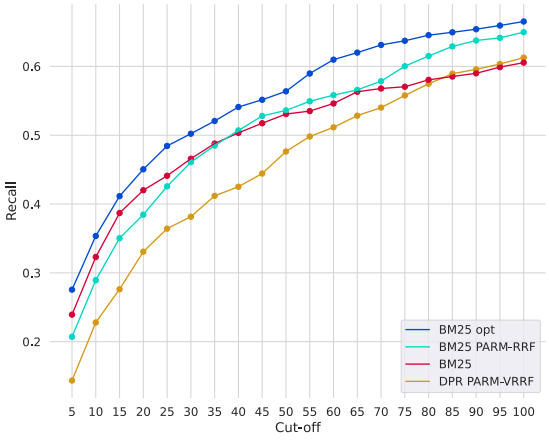


Fig. 2. First stage retrieval recall at different cut-offs for BM25, optimized BM25 and BM25 and DPR with PARM on the COLIEE 2021 Task 1 test set.

	COLIEE21 Test		
	R@15	R@50	R@100
BM25	.3869	.5307	.6054
DPR PARM	.2761†	.4763†	.6127†
BM25 PARM	.3504†	.5360	.6496†
BM25 opt	**.4114**†	**.5637**†	**.6651**†

Fig. 3. First stage retrieval effectiveness in terms of Recall (R) with different cutoff (15/50/100). Statistical significance to BM25 baseline denoted with † with paired t-test; p ¡ 0.05, Bonferroni correction with $n = 3$.

	COLIEE21 Test			COLIEE22 Test		
	P	R	F1	P	R	F1
Previous winner						
COLIEE 2021 winner	.1533	.2556	.1917	.	.	.
COLIEE 2022 winner	.	.	.	.4111	.3389	.3715
Our runs						
BM25 opt	.1700	.2536	.2035	.2447	.3679	.2939
BERT	.1440	.2463	.1817	.1940	.2984	.2351
BERT$_{sum}$	.1096	.1915	.1394	.1960	.2989	.2368
BERT-PLI	.1536	.2133	.1786	.2267	.3338	.2700
MTFT-BERT$_{sum}$	.1152	.2086	.1486	.1980	.3016	.2391
MTFT-BERT	**.1744**	**.2999**	**.2205**	.2247	.3349	.2689

Table 2. Re-ranking effectiveness results on COLIEE 2021 Task 1 test and COLIEE 2022 Task 1 test for Precision (P), Recall (R) and F1-Score (F1). Our runs are at a cutoff at 5.

We choose the cut-offs of 15 and 50 as the MTFT-BERT model uses the top-15 for re-ranking and the BERT-PLI model the top-50. Here we also see that the optimized BM25 model achieves the highest recall at cut-offs below or equal to 100.

5.2 RQ2: Re-ranking Effectiveness

As we can see in the Figure 2, in contrast to the COLIEE 2021, BM25$_{opt}$ shows the highest performance compared to the re-ranked results. Also, BERT-PLI and MTFT-BERT ranking quality are in par with each other. While we can see that MTFT-BERT shows improvement over BERT which is similar to the results from COLIEE 2021, it cannot show superior performance over BM25$_{opt}$. It is noteworthy that the experimental setup in COLIEE 2021 and COLIEE 2022 are different in the sense that the evaluation in COLIEE 2021 is collection-specific as the candidate documents are seen during training which is not the same as in COLIEE 2022. It should be noted that the result in Figure 2 are achieved using pytrec evaluation toolkit [10] which shows different results in comparison to the method used by COLIEE 2022 competition organizers.

5.3 RQ3: Textual Content for Re-ranking

We explore the effect of using first-passage, summaries, and the whole document content on the re-ranking effectiveness. We can see that in both BERT and MTFT-BERT ranker, using the first passage shows higher ranking quality in comparison to the summary. In addition, we can see that while BERT-PLI which takes the whole document into account is more effective than BERT and MTFT-BERT acting on the summaries, it is not more effective than BERT and MTFT-BERT based on the first passage. However, it should be noted that in this latter comparison we cannot derive a definite conclusion as the re-ranking models have different architectures.

[10] https://github.com/cvangysel/pytrec_eval

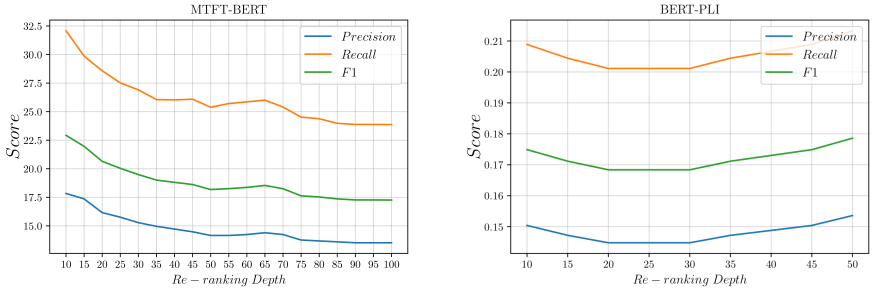


Fig. 4. Effectiveness at different re-ranking depths for MTFT-BERT on the left and BERT-PLI on the right on the COLIEE 2021 Task 1 test set.

5.4 RQ4: Effect of Re-ranking Depth

We investigate the effect of the re-ranking depth on the effectiveness for the MTFT-BERT based on the first passage and also the BERT-PLI model. In Figure 4 the Precision, Recall and F1-Score for different re-ranking depths are visualized for both models. We see that for MTFT-BERT the performance decreases with higher re-ranking depth, therefore we conclude that re-ranking at shallower depths is the more effective for the MTFT-BERT model. The BERT-PLI model reaches a high F1-score at the shallow re-ranking depths of 10, but reaches the highest F1-score at 50. We only re-rank the top-50 results due to the computational complexity of the BERT-PLI model and leave explorations of deeper re-ranking depth to future work.

We submit for the test set of COLIEE 2022 the optimized BM25 run with a cut-off at 5 as first run (DSSR_01). As second run we submit the $BM25_{opt}$ run re-ranked with MTFT-BERT based on the first passage (DSSR_02) with a re-ranking depth of 15 and a cut-off at 5. It should be noted that this run was broken as we had a technical issue during its evaluation and therefore our evaluation results differ from the leaderboard. As third run (DSSR_03) the top-50 of optimized BM25, re-ranked with BERT-PLI with the AttenLSTM trained on COLIEE 2019 Task 1 data and with a cut-off at 5 and re-ranking depth at 50.

6 Conclusion

This paper documents our participation in the COLIEE 2022 legal case retrieval task. As the experimental setting for the COLIEE 2021 and COLIEE 2022 are different in terms of the candidate documents to be re-ranked during the training and validation, we were interested to evaluate the performance of the models which previously show ranking improvements for legal case retrieval. We investigate dense and lexical first stage retrieval methods and find that lexical retrieval with the optimized BM25 leads to the highest first stage retrieval recall. Furthermore, we investigate if the shallow re-ranking with a cross-encoder BERT and a MTFT-BERT re-ranker still shows a higher ranking quality in comparison to other models including BERT-PLI and the optimized BM25. Our experiments demonstrate that MTFT-BERT improves over BERT ranker on the COLIEE 2021 test set. Besides, we conclude that while in COLIEE 2021 using

a shallow re-ranking with MTFT-BERT showed superior ranking quality, we cannot see the same effect in the COLIEE 2022 which could be due to the aforementioned difference in the experimental setup.

7 Acknowledgments

This work is funded by the DoSSIER project under European Union's Horizon 2020 research and innovation program, Marie Skłodowska-Curie grant agreement No. 860721.

References

1. Abolghasemi, A., Verberne, S., Azzopardi, L.: Improving bert-based query-by-document retrieval with multi-task optimization. In: Advances in Information Retrieval, 44rd European Conference on IR Research, ECIR 2022 (2022)
2. Al-Kofahi, K., Tyrrell, A., Vachher, A., Jackson, P.: A machine learning approach to prior case retrieval. pp. 88–93. Association for Computing Machinery (2001)
3. Althammer, S., Askari, A., Verberne, S., Hanbury, A.: Dossier@coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In: Proceedings of the Eigth International Competition on Legal Information Extraction/Entailment (COLIEE 2021). 18th International Conference on Artificial Intelligence and Law (ICAIL) (2021)
4. Althammer, S., Hofstätter, S., Hanbury, A.: Cross-domain retrieval in the legal and patent domains: a reproducibility study. In: Advances in Information Retrieval, 43rd European Conference on IR Research, ECIR 2021 (2021)
5. Althammer, S., Hofstätter, S., Sertkan, M., Verberne, S., Hanbury, A.: Paragraph aggregation retrieval model (parm) for dense document-to-document retrieval. In: Advances in Information Retrieval, 44rd European Conference on IR Research, ECIR 2022 (2022)
6. Askari, A., Verberne, S.: Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In: Alonso, O., Marchesin, S., Najork, M., Silvello, G. (eds.) Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems, Padova, Italy, September 15-18, 2021. CEUR Workshop Proceedings, vol. 2950, pp. 162–170. CEUR-WS.org (2021), http://ceur-ws.org/Vol-2950/paper-02.pdf
7. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261, https://www.aclweb.org/anthology/2020.findings-emnlp.261
8. Cormack, G.V., Clarke, C.L.A., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 758–759. SIGIR '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1571941.1572114, https://doi.org/10.1145/1571941.1572114
9. 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

10. 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
11. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In: Proc. of SIGIR (2021)
12. Hofstätter, S., Lipani, A., Althammer, S., Zlabinger, M., Hanbury, A.: Mitigating the position bias of transformer models in passage re-ranking. In: Hiemstra, D., Moens, M.F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) Advances in Information Retrieval. pp. 238–253. Springer International Publishing, Cham (2021)
13. Hofstätter, S., Althammer, S., Schröder, M., Sertkan, M., Hanbury, A.: Improving efficient neural ranking models with cross-architecture knowledge distillation (2021)
14. Kamphuis, C., de Vries, A.P., Boytsov, L., Lin, J.: Which BM25 do you mean? A large-scale reproducibility study of scoring variants. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) Advances in Information Retrieval. pp. 28–34. Springer International Publishing, Cham (2020)
15. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550, https://www.aclweb.org/anthology/2020.emnlp-main.550
16. Li, J., Zhao, X., Liu1, J., Wen1, J., Yang, M.: Siat@coliee-2021: Combining statistics recall and semantic ranking for legal case retrieval and entailment. In: Proceedings of the Eigth International Competition on Legal Information Extraction/Entailment (COLIEE 2021). 18th International Conference on Artificial Intelligence and Law (ICAIL) (2021)
17. Locke, D., Zuccon, G., Scells, H.: Automatic query generation from legal texts for case law retrieval. Information Retrieval Technology: 13th Asia Information Retrieval Societies Conference, AIRS 2017, Proceedings (Lecture Notes in Computer Science, Volume 10648) pp. 181–193 (2017)
18. Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., Ma, S.: Retrieving legal cases from a large-scale candidate corpus (2021)
19. Nguyen, H.T., Nguyen, P.M., Vuong, T.H.Y., Bui, Q.M., Nguyen, C.M., Dang, B.T., Tran, V., Nguyen, M.L., Satoh, K.: Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021. arXiv preprint arXiv:2106.13405 (2021)
20. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
21. Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-stage document ranking with bert (2019)
22. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (Apr 2009). https://doi.org/10.1561/1500000019, https://doi.org/10.1561/1500000019
23. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: Yes, bm25 is a strong baseline for legal case retrieval. arXiv preprint arXiv:2105.05686 (2021)
24. Russell-Rose, T., Chamberlain, J., Azzopardi, L.: Information retrieval in the workplace: A comparison of professional search practices. Information Processing and Management **54**, 1042–1057 (11 2018). https://doi.org/10.1016/j.ipm.2018.07.003
25. 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: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 3501–3507. International Joint Conferences on Artificial Intelligence Organization (7 2020). https://doi.org/10.24963/ijcai.2020/484, https://doi.org/10.24963/ijcai.2020/484, main track
26. Turtle, H.: Text retrieval in the legal world. Artificial Intelligence and Law, Volume 3 pp. 5–54 (1995)

27. Van Opijnen, M., Santos, C.: On the concept of relevance in legal information retrieval. Artif. Intell. Law **25**(1), 65–87 (Mar 2017). https://doi.org/10.1007/s10506-017-9195-8, https://doi.org/10.1007/s10506-017-9195-8
28. Verberne, S., Sappelli, M., Hiemstra, D., Kraaij, W.: Evaluation and analysis of term scoring methods for term extraction. In: Information Retrieval Journal (2016)

LeiBi@COLIEE 2022: Aggregating Tuned Lexical Models with a Cluster-driven BERT-based Model for Case Law Retrieval

Arian Askari*[1], Georgios Peikos*[2], Gabriella Pasi[2], and Suzan Verberne[1]

[1] Leiden Institute of Advanced Computer Science, Leiden University
{a.askari,s.verberne}@liacs.leidenuniv.nl
[2] Department of Informatics, Systems and Communication, University of Milano-Bicocca {g.peikos,gabriella.pasi}@unimib.it

Abstract. This paper summarizes our approaches submitted to the case law retrieval task in the Competition on Legal Information Extraction/Entailment (COLIEE) 2022. Our methodology consists of four steps; in detail, given a legal case as a query, we reformulate it by extracting various meaningful sentences or n-grams. Then, we utilize the pre-processed query case to retrieve an initial set of possible relevant legal cases, which we further re-rank. Lastly, we aggregate the relevance scores obtained by the first stage and the re-ranking models to improve retrieval effectiveness. In each step of our methodology, we explore various well-known and novel methods. In particular, to reformulate the query cases aiming to make them shorter, we extract unigrams using three different statistical methods: KLI, PLM, IDF-r, as well as models that leverage embeddings (e.g., KeyBERT). Moreover, we investigate if automatic summarization using Longformer-Encoder-Decoder (LED) can produce an effective query representation for this retrieval task. Furthermore, we propose a novel re-ranking cluster-driven approach, which leverages Sentence-BERT [21] models that are pre-tuned on large amounts of data for embedding sentences from query and candidate documents. Finally, we employ a linear aggregation method to combine the relevance scores obtained by traditional IR models and neural-based models, aiming to incorporate the semantic understanding of neural models and the statistically measured topical relevance. We show that aggregating these relevance scores can improve the overall retrieval effectiveness.

1 Introduction

COLIEE is a workshop held every year since 2014 as a series of evaluation competitions related to case law. It divides four tasks into two groups: retrieval and entailment. For our participation, we have focused on Task 1, i.e., legal case law retrieval.

* Both authors contributed equally to this research.

Finding supporting precedents for a new case is critical for lawyers to fulfil their responsibilities to the court in countries with common law systems. However, due to the large number of digital legal records — in 2021, the number of filings in the United States district courts for total cases and criminal defendants was $526,477$[3] — it requires time for legal professionals to scan specific cases and retrieve the relevant sections manually. According to studies, attorneys spend about 15 hours per week looking for case law [14].

To cover this vast amount of information requests, the use of information retrieval technologies tailored to the legal field is necessary. Lawyers expect their search algorithms to identify all relevant cases. At the same time, in practice, they would often only analyze up to 50 retrieved results, necessitating a precision-oriented retrieval approach [9]. To fit the requirements of this search task, we leverage traditional lexical based IR models, which we optimize by tuning their parameters. By doing so, we manage to increase their precision, as we showed in prior work [4]. To further enhance the retrieval precision at high ranks, we experiment with several neural-based re-ranking models that measure the semantic similarity between the query and the top-50 retrieved documents.

Specifically, in this work, we address the following questions regarding the case law retrieval task:
1. What is the retrieval effectiveness that can be obtained when various query reformulation methods are used along with traditional IR models for legal case retrieval?
2. Can the retrieval effectiveness be improved when a re-ranking approach based on Sentence-BERT models that have pre-tuned on billions of data is employed on top of a traditional IR model?
3. Can the aggregation of the relevance scores obtained by a traditional IR model and a neural re-ranker lead to greater retrieval precision?

We first employ various collection dependent and independent query reformulation approaches. Then, using the obtained query formats, we extensively evaluate the retrieval effectiveness of BM25, a Divergence from Randomness model, and a statistical language model. Moreover, we tune the models' associated parameters to show the significant impact of tuning these models in the studied retrieval task. In addition, we investigate the effectiveness of Vanilla BERT as a neural-based re-ranker. Finally, we aggregate the two relevance scores of best lexical model and the proposed cluster-driven re-ranker to examine whether this can further improve retrieval effectiveness.

2 Task description

For a legal professional, the case law process consists of reading a new unseen case Q_d and then, given a collection of previous cases, choosing supporting cases $S1, S2, ..., Sn$ (that are called 'noticed cases') to strengthen the decision for Q_d.

[3] https://www.uscourts.gov/statistics-reports

Challenges. Case law retrieval is a form of Query-by-document (QBD) retrieval, a task in which the user enters a text document – instead of a few keywords — as a query, and the Information Retrieval system finds relevant documents from a text corpus [26,25]. Transformer-based architectures [23], such as BERT-based ranking models [19,1] have yielded improvements in many IR tasks. However, the time and memory complexity of the self-attention mechanism in these architectures is $O(L^2)$ over a sequence of length L [6,27]. That causes challenges in QBD tasks where we have long queries and documents. For instance, the average length of queries and documents in the legal case law retrieval task of COLIEE 2022 is more than 4000 words. Moreover, variants of Transformers that aim to cover long sequences such as Longformer [6] have not shown high effectiveness which could be due to the their sparsified attention mechanism [22] or the limited number of training instances for professional search tasks [4]. In this paper, we propose a cluster-driven BERT-based methodology that can consider the whole length of the query, while simultaneously overcoming the complexity imposed by length of text. Our novel re-ranking approach further improves the system's effectiveness using a weighting mechanism that combines the BM25 score obtained from the initial retrieval with the scores from the neural models.

3 Methodology

Our methodology consists of four steps, each to tackle case law retrieval: (1) Query Case Reformulation: to make queries shorter and in keyword format for lexical models as previous studies show it improves the retrieval effectiveness [4,16] (2) Lexical Ranker: for retrieving first stage results (3) Neural Re-ranker: for re-ranking top-k candidate documents considering semantic besides of exact keyword matching (4) Relevance score aggregation: to combine {statistical (2) and semantic (3)}-based models together. Question 1, 2 (as well as challenge mentioned in Section 2), and 3 that are mentioned in introduction addressed by step (1,2), (2), and (4) respectively.

3.1 Query Case Reformulation

In the literature, several approaches for reformulating a verbose query into either a keyword like representation or a summary have been proposed [4]. We experiment with two different approaches to create shorter query cases: (1) term extraction and (2) abstractive summarization. We do not experiment with named entity recognition and noun phrase detection methods as they have shown lower effectiveness in comparison to KLI previously on COLIEE in previous work [4].

Term extraction. We experiment with three lexical-based similar to Locke et al. [16] and one neural-based approaches for term extraction: (1) Kullback-Leibler divergence (KLI), (2) parsimonious language model (PLM) [12], (3) IDF-r [13], (4) KeyBERT [11].

KLI. We used Kullback-Leibler divergence for Informativeness (KLI) [24]. The KLI score was computed for each term t in a query case document,Q_d similar to [4]:

$$KLI(t) = P(t|Q_d) \times \log \frac{P(t|Q_d)}{P(t|C)} \tag{1}$$

where $P(t|D)$ is the probability of t in the query document D and $P(t|C)$ is the probability of t in a background language model. We use all candidate documents as the background collection to compute $P(t|C)$.

IDF-r. The IDF-r method selects the $\lceil \frac{|Q_d|}{r} \rceil$ terms in Q_d with the highest Inverse Document Frequency (IDF-r) score where r refers to the proportion of selected terms.

PLM. For the PLM method, we used our collection, $P(t|C)$, as the background language model and the information object, Q_d, as the foreground language model. The expectation maximization algorithm was used to estimate probabilities, with the following steps:

$$E - step : e_t = tf(t, Q_d).\frac{\lambda.P(t|Q_d)}{(1-\lambda).P(t|C) + \lambda P(t|Q_d)} \tag{2}$$

$$M - step : P(t|Q_d) = \frac{e_t}{\sum_{t' \in Q_d} e_{t'}} \tag{3}$$

Where $\lambda \in [0, 1]$ is a smoothing parameter that controls the influence of statistics from the collection (C) over the statistics from the information object (D).

KeyBERT. The aforementioned methods rely on document and collection statistics to extract informative terms. Recent advances allow the employment of word embeddings for term extraction. These methods can capture the semantic relationship between a document's terms and extract those that better represent its content. For our experiments, we have employed KeyBERT [11], which is a term extraction technique that leverages BERT-based pre-trained models.

The main idea behind KeyBERT is that those terms that have a vector representation similar to the document's vector representation, can be considered the document's most representative terms. However, legal documents can be lengthy. Moreover, a document's topic may shift from one paragraph to another. These are two characteristics of the legal domain which we had considered when we applied KeyBERT to extract representative terms from the query case.

Specifically, given a query case, we split it up into its paragraphs. Then, given each paragraph of a query case, we employ KeyBERT to create a list of candidate n-grams (bag-of-words). Having done that, KeyBERT produces an embedding representation for the whole paragraph and an embedding representation for each of its candidate n-grams. To identify the representative terms,

it measures a pairwise cosine similarity score between each n-gram and the obtained paragraph embedding vector. Regarding the paragraph embedding vector, if the number of paragraph tokens exceeds the transformer's token limit, the paragraph's embedding vector is computed using mean pooling on the individual paragraph embedding vectors. However, as reported in [11], the extracted terms are often similar to each other. To overcome this issue, KeyBERT applies an extra step to diversify the extracted terms that relies either on the Maximal Marginal Relevance (MMR) formula [7] or a simpler formula that extracts the least similar terms based on their pairwise cosine similarity.

We employ KeyBERT following the above-mentioned steps and tuning all the required parameters using a training set. Also, we experiment with both domain-specific and non-domain-specific pre-trained models to obtain the word and paragraph embeddings. Further details related to its parameterization are reported in Section 5 while the obtained experimental results are described in Section 4.2.

Abstractive Sumarization The current state of the art in abstractive summarization is based on Transformer models [15,20]. As the input of pre-trained available models of these architectures is limited to 1024 tokens, [6] proposed Longformer-Encoder-Decoder (LED), which is a Transformer variant that supports much longer inputs. For this competition, we fine-tune LED on caselaw summaries of COLIEE 2018 followed by the same implementation by [4] and evaluate the effectiveness of LED (hereafter mentioned as *summaryQ LED fine-tuned*) for case law retrieval using three traditional IR model.

3.2 Ranking models

By experimenting with different query reformulation and summarization methods (see above), we have created several representations for the studied query cases which we have evaluated in combination to various statistical and neural IR models and neural re-rankers. Specifically, we employ BM25, and we fine-tuned its parameters to the peculiarities of the studied task (lengthy documents and queries). Although BM25, and statistical language modelling approaches are the most well-known word-based information retrieval models, recent works do not often tune their associated parameters. Specifically, the BM25 formula contains two parameters ($k1$, and b) associated with the term frequency saturation and document length normalization. In the legal domain, particularly in the task of case law retrieval, tuning them is crucial, as both the considered query cases and the documents are lengthy.

In addition, we experiment with statistical language models and neural models to investigate the effectiveness of domain-specific and non-domain-specific language modelling. Lastly, two submissions are based on a novel re-ranker on the top fifty retrieved documents. Therefore, we wanted to evaluate if some other IR models can retrieve more relevant documents in the first positions. To this

aim, we employ various models that are based on the Divergence from the Randomness framework, proposed by [3]; it has been found that these models can achieve higher precision compared to BM25 for some search tasks.

3.3 Cluster-driven method (re-ranking model)

Transformer-based models are limited in taking into account long documents, so we split the queries and documents at the sentence-level and embed the document in sentence-level using Sentence-BERT (SBERT [21]). We develop a method that exploits clusters of sentences extracted from a query and a document. The proposed method, utilizes three different pre-fine-tuned SBERT's models to obtain appropriate embeddings for each part of the proposed method. Given a query Q_d, we first identify the three most important sentences and use them as query representation, inspired by the Lead-3 model [18]. Lead-3 proposes that the first three sentences of the document are good representatives for extractive summarization. To this aim, we apply K-means (k=3) as our clustering method to find three clusters of query sentences; each cluster contains semantically similar sentences. Then, we compute the centroid of the cluster and select a sentence of cluster that is most close to its centroid as the representation of the cluster. The three representative sentences are denoted as $3S_{Qd}$. Then, given a candidate document d, we find the most similar sentences ($3S_d$) of that document to $3S_{Qd}$, by computing cosine similarity. To do that, we use a bi-encoder model that is pre-fine-tuned for computing cosine similarity between two separately embedded sentences. To compute the overall relevance score between the Q_d and each d we sum the similarity scores of $3S_q$ and $3S_d$ using a cross-encoder model that is pre-fine-tuned to compute probability of relevance between two sentences. We re-rank candidate documents based on this relevance score. We miss no part of candidate query and document into our methodology due to the sentence-level embedding and considering all of sentence embeddings into our methodology for computing relevance score. In summary, our approach consists of three steps: (1) Finding the three most important sentences in the query by clustering the embeddings of query sentences using a pre-fine-tuned bi-encoder to embed sentences and k-means for clustering; (2) Finding the most similar sentences of a candidate document according to the sentences found in (1) by computing cosine similarity (3) Computing the final relevance score by computing a sum over the probability of relevance of pair of query and document sentences found in (2).

3.4 Relevance score aggregation

In the literature, it has been found that aggregating the scores of the initial ranker and re-ranker yields improvements in terms of retrieval effectiveness[4,2,5]. Our experiments also observed this behaviour as it was found that re-ranking improves several queries but hurts others. One of the most common

methods is the linear aggregation of the two obtained relevance scores. The aggregation approach mentioned above associates an importance value to the relevance score of the ranker (S_BM25) and one value to the relevance score of the re-ranker ($S_{cluster-driven}$). Similar to [2], the final relevance score is computed as below to find the optimal value for weighting method (hereafter mentioned as *weighting*) on the validation set where $\alpha, \beta \in \mathbb{R}$:

$$s_{BM25+cluster-driven}(d, Q_d) = \alpha\ s_{BM25}(d, Q_d) + \beta\ s_{cluster-driven}(d, Q_d).$$

4 Experiment design

4.1 Collection

For our experiments, we use the collection provided by the organizers of COLIEE'22 [10]. Precisely, the collection consists of a training and a test set. The test set contains 300 query cases associated with 1,563 relevant documents. The training set includes 898 query cases and 4,415 relevance assessments. To fine-tune our models, we have split the train set into train and validation sets. The validation set in which we report our results was created by selecting the last 250 queries out of 898 training queries. Hereafter, when we refer to the training set, we refer to the 648 remained queries.

4.2 Query pre-processing

Tuning the KLI & PLM methods. Both the KLI and the PLM methods score each document term, creating a term ranking. As a result, one can identify which proportion of this ranking can better represent its content. We found the optimal term proportion based on optimizing it according to F1 score.

Tuning KeyBERT. To extract essential terms for each document paragraph using KeyBERT, we use its official implementation, which is publicly available [4]. Regarding its parametric setup, there are several parameters in this implementation, such as the size of the extracted n-grams, the number of extracted terms, and the pre-trained embedding model. In addition, one can choose between two different term diversification methods (Max Sum Similarity and MMR) and tune their corresponding diversification coefficient. We tuned the parameters on the validation set.

For our experiments, we utilize two pre-trained models to obtain the embedding representations, namely the *all-MiniLM-L6-v2* sentence transformer [5] and *Legal BERT* [8]. In addition, we experimented with both the Maximal Marginal Relevance (MMR) diversification and the Max Sum Similarity formulas and tuned their diversification coefficient value in range [.2-.8] using a step of .1.

[4] https://github.com/MaartenGr/KeyBERT
[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Moreover, we set the parameter related to the n-gram size so that KeyBERT produces uni-grams and bi-grams. Finally, we experimented by altering the number of the extracted term from each paragraph in the range [5-25] using a step of 5. To tune all the required parameters, we have used the train set described in Section 4.1. We concatenated all the obtained n-grams into one query text and used it for retrieval. As optimal parameters, we have chosen those that maximized the F1 score in the considered training set.

4.3 Lexical rankers

BM25 We ran BM25 using the default parameter values of $k1 = 1.2$ and $b = 0.75$, and its implementation from ElasticSearch. In addition, we created *BM25-opt* by tuning its hyperparameter with doing grid search over $b \in \{0, 0.1, 0.2, , 1\}$ and $k = \in \{0, 0.1, 0.2, , 3\}$ on the validation set.

Language modelling We used the built-in similarity functions of Elasticsearch for the implementation of Language Modelling (LM) with two different smoothings: Dirichlet smoothing and Jelinek Mercer (JM) smoothing. We only report the results for JM smoothing since we get similar results from these two smoothing methods. We also optimised the hyperparameter value (λ) for Language Modelling with Jelinek Mercer smoothing (LM JM). We found $\lambda = 0.1$ as the optimal value for LM JM with KLI.

Divergence from Randomness (DFR) To implement the DFR models, we have used PyTerrier [17], and we tuned their associated parameter, using the training set, aiming at optimising the P@4 measure. Specifically, these models are associated with a free parameter c that controls the term frequency normalisation component[3]. We found that for several DFR models —In_expC2, In_expB2, lnL2, lnB2, described in [3]— the default parameter setting (c=.1) was the optimal value.

Cluster-driven method. We use *all-mpnet-base-v2*, *msmarco-bert-base-dot-v5*, and *ms-marco-MiniLM-L-12-V2* models of Sentence-BERT for clustering, computing cosine similarity and computing probability of relevance respectively

5 Results

This section presents the top-performing retrieval models' experimental results on the validation set. Table 1 presents the retrieval effectiveness obtained across several query representations and models, while Table 2 presents the results obtained by the re-ranker and the score weighting method. For every retrieval model and query representation method presented in Tables [1,2], the reported parameters were those that optimized the F1 score on the training set.

Table 1. BM25, LM Jelinek Mercer (JM) and DFR retrieval results for the ranking of candidate documents on the validation set of COLIEE'22. We only report result of optimized DFR In_expC2.

Method	Query representation	P %	R %	F1 %
BM25 (not optimized)	original text	13.10	19.09	15.53
BM25	original text	16.30	19.34	17.69
BM25	KeyBERT (Legal BERT)	6.50	10.29	7.96
BM25	KeyBERT (all-MiniLM-L6-v2)	6.55	9.80	7.85
BM25	KLI (1-gram/40% term portion)	**19.30**	**28.59**	**23.04**
BM25	PLM (1-gram/50% term portion)	17.20	25.91	20.67
BM25	IDF (1-gram/90% term portion)	15.80	23.11	18.76
BM25	summaryQ LED fine-tuned	5.73	12.84	7.92
LM JM (not optimized)	original text	12.19	18.54	14.70
LM JM	original text	17.10	25.32	20.41
LM JM	KeyBERT (Legal BERT)	6.23	10.90	7.92
LM JM	KeyBERT (all-MiniLM-L6-v2)	6.02	10.98	7.77
LM JM	KLI (1-gram/40% term portion)	15.76	29.47	20.53
LM JM	PLM (1-gram/50% term portion)	16.50	24.05	19.57
LM JM	IDF (1-gram/90% term portion)	5.87	12.01	7.88
LM JM	summaryQ LED fine-tuned	5.07	11.73	7.07
DFR In_expC2	original text	10.80	15.87	12.85
DFR In_expC2	KeyBERT (Legal BERT)	9.40	13.39	11.04
DFR In_expC2	KeyBERT (all-MiniLM-L6-v2)	11.00	15.60	12.91
DFR In_expC2	KLI (1-gram/40% term portion)	12.70	19.07	15.24
DFR In_expC2	PLM (1-gram/50% term portion)	12.50	18.18	14.82
DFR In_expC2	IDF (1-gram/90% term portion)	9.80	14.29	11.62
DFR In_expC2	summaryQ LED fine-tuned	7.60	10.93	8.96

Summary of results The results show that the optimal retrieval effectiveness is achieved when the KLI method, with the reported parameters, is combined with the BM25 model. Therefore, this retrieval combination was selected as one of our submitted runs. Moreover, another remark is related to the fact that all of the query representations that were created using embedding based models lead to poor retrieval effectiveness. Finally, we observe that all of the employed query reformulation approaches seem robust and yield similar results across the employed retrieval models.

Retrieval using the KeyBERT terms. From each paragraph of a query case, KeyBERT returns a pre-defined number of extracted n-grams that are later concatenated to create a single query representation used for retrieval. Using the train set, we have identified the optimal parameter setting. In particular, it has been found that the optimal number of extracted n-grams is 20, the diversification coefficient is 0.6, while it was found that the MMR term diversification formula leads to greater improvements in terms of F1 score on the training set compared to the Max Sum formula. Moreover, it has been found

Table 2. Retrieval results for the ranking of candidate documents on the validation set of COLIEE'22, using cluster-driven as re-ranker and weighting as score aggregator between BM25 and cluster-driven method. BM25 optimized reported for comparison.

Method	Query representation	P %	R %	F1 %
BM25	KLI (1-gram/40% term portion)	19.30	**28.59**	23.0
Cluster-driven	original text (sentence-level)	16.20	23.31	19.11
Weighting	-	**19.75**	28.29	**23.26**

that these optimal parameters remained unchanged when we used the domain-specific pre-trained model to obtain the embedding representations. Therefore, we have investigated the effectiveness of the KeyBERT term extraction method, using the above mentioned parameters in combination with either a domain specific pre-trained model or a non-domain specific pre-trained model to obtain the embedding representation. The obtained experimental results on the validation set are presented in Table 1.

Retrieval using DFR models. By altering the basic models used to calculate the probabilities in the generic DFR formula along with the term frequency normalization, one may obtain several DFR models. We have experimented with several variation of DFR retrieval models, using the original query text as input. Results have shown that the Inverse Expected Document Frequency model with Bernoulli after-effect and normalisation 2, namely *In_expC2*, yields the best performance. As a result, we used only this model in combination with the various query representation. The obtained results are presented in Table 1.

Neural re-ranker We re-rank the top-50 candidate documents retrieved using the BM25-optimized model. While, as shown in Table 2, the BM25-optimized outperforms our cluster-driven method in terms of mean effectiveness, our investigation show that there are queries for which our cluster-driven method improves their performance over the BM25-optimized. As a result, the neural re-ranker is effective when considered in a weighting aggregation method (See section 5).

It is noteworthy to mention that we experimented with Vanilla BERT following our previous works on COLIEE in [4]. However, we achieved lower results than BM25, and we found that applying the weighting mechanism on the scores obtained by Vanilla BERT and BM25 could not outperform BM25. The best F1 that we could achieve by BERT was 18.47 on the validation set, which is lower than the results achieved by cluster driven method on the same set (F1 19.19). As a result, we only exploit the proposed cluster-driven method as our neural re-ranker in this paper. We submitted cluster-driven method to assess its effectiveness on test set in the COLIEE 2022 competition as the second submission.

Weighting. We use the weights $[1, 2, ..., 100]$ for α and β values (See section 3.4). Optimal weights found as 36 for cluster-driven method and 48 for BM25. From Table 2 it is clear that aggregating the two relevance scores obtained by the

Table 3. Retrieval results for the ranking of candidate documents on the test set of COLIEE'22, using our top-performing approaches.

Method	Extractor/Sumarizer	P %	R %	F1 %
BM25 optimized	KLI (1-gram)	**29.92**	35.75	32.57
Cluster driven	original text (sentence-level)	23.92	28.92	26.18
Weighting	-	**29.92**	**36.26**	**32.78**

KLI+BM25 retrieval pipeline and the cluster-based re-ranker, further improves the retrieval effectiveness on the validation set. Therefore, that approach was our third submission.

6 Discussion

Proportion of KLI terms Table 1 show the effect of query representation methods on lexical retrieval models. Statistical term extraction methods (KLI, PLM, IDF) shows higher effectiveness in compare to KeyBERT and Longformer-Encoder-Decoder (LED) and KLI shows the most effectiveness within all three rankers (BM25, LM JM, DFR ln_expC2). BM25 achieve highest effectiveness using query terms that are extracted by KLI. We analyze the effect of using different proportion of terms scored by our best term extraction method KLI on BM25 (our best lexical ranker) and DFR to assess the sensitivity of BM25 on using different proportions of terms and compare BM25's sensitivity with DFR as another lexical ranker. The figures 1 show BM25 has higher sensitivity on using different proportion of terms in comarparison to DFR ln_expC2. We interpret this sensitivity as a strength of BM25 in this case because using top-40% of query terms as the query improve the effectiveness of BM25 more than DFR while both rankers receive highest effectiveness with using same proportion of terms extracted by KLI. On both rankers, increasing the proportion of terms till 40% improves the recall and consequently F1 while the precision is almost consistent. This can be related to the fact that enriching the query with more approperiate words can increase the chance of retrieving relevant documents that are mathced with that added words in lexical rankers.

Effect of the cut-off value. As the task is to retrieve the relevant cases to a given query q, we consider the top-k-ranked documents d in the ranked list as relevant and denote k as cut-off value. We evaluate the best cut-off value k depending on the F1 score of the validation set using pytrec_eval[6]. The results are shown in Figures 2 and 3 show the effect of cut-of value on F1 score. We found cut-off 4 as optimal value for BM25 and cluster-driven method and 5 for weighting method. In these figures, BM25 refers to the BM25 optimized.

[6] https://github.com/cvangysel/pytrec_eval

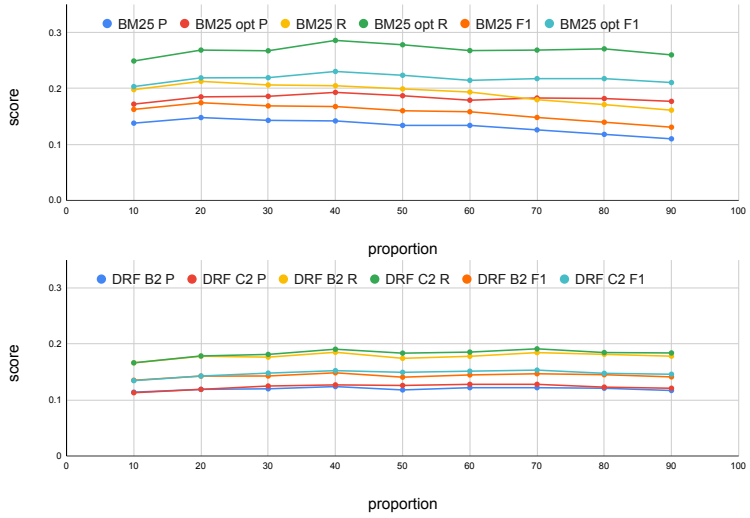


Fig. 1. Retrieval results for BM25 and BM25 opt with default and optimized parameters and DFR (B2 and C2) on the validation set of COLIEE'22 with using different proportion of terms scored by KLI method as a shorter query. P and R refers to Precision and Recall respectively.

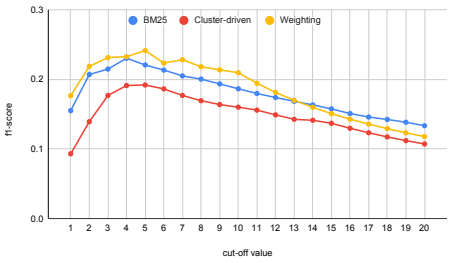


Fig. 2. F1-score for Task 1 on the **validation** set for different re-ranking depth (cut-off value for first-stage ranker)

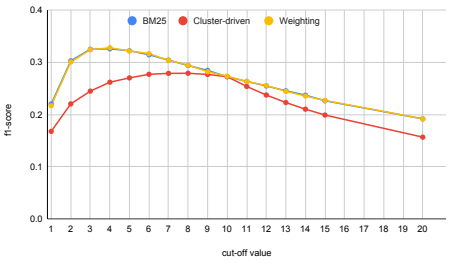


Fig. 3. F1-score for Task 1 on the **test** set for different re-ranking depth (cut-off value for first-stage ranker)

Submitted Experiments We submitted three run files for task 1: (1) BM25-optimized with KLI as our term extraction method; (2) the cluster-driven method as a re-ranker using BM25-optimized with KLI as initial ranker; (3) a weighting model that aggregates the output scores of (1) and (2) using linear aggregation. Table 3 shows that the weighting mechanism could improve BM25 optimization by utilizing both lexical (obtained by BM25) and neural-based matching (obtained by cluster-driven) scores.

7 Conclusion and future work

Our participation at COLIEE 2022 in Task 1 allowed exploring the effect of term extraction methods on lexical and neural models in case law retrieval. We identify that the presence of long documents creates significant issues both for neural re-ranking strategies and statistical models. Therefore we propose, compare and evaluate a cluster-driven BERT-based re-ranker that considers the whole length of the query and candidate documents. The model could outperform Vanilla BERT when used with BM25 as the initial ranker.

We show that optimized BM25 is the best lexical matching model among LM and DFR models. KLI is the best term extraction method compared to PLM, IDF-r, and KeyBERT. We find that the combination of lexical and neural models improves the overall effectiveness of the ranking. In the future, we plan to investigate further the effectiveness of our novel cluster-driven model, by considering the importance of each cluster, in the estimation of the overall relevance score.

8 Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860721 – DoSSIER (H2020-EU.1.3.1.).

References

1. Abolghasemi, A., Verberne, S., Azzopardi, L.: Improving bert-based query-by-document retrieval with multi-task optimization. In: European Conference on Information Retrieval. Springer (2022)
2. Althammer, S., Askari, A., Verberne, S., Hanbury, A.: Dossier@ coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In: Proceedings of the COLIEE Workshop in ICAIL. (2021)
3. Amati, G., van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. (4) (2002)
4. Askari, A., Verberne, S.: Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieva. In: Proceedings of the second international conference on design of experimental search & information REtrieval systems (DESIRES). pp. 162–170. CEUR (2021)
5. Askari, A., Verberne, S., Pasi, G.: Expert finding in legal community question answering. In: European Conference on Information Retrieval. Springer (2022)
6. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
7. Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., Jaggi, M.: Simple unsupervised keyphrase extraction using sentence embeddings. In: Korhonen, A., Titov, I. (eds.) Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018. pp. 221–229 (2018)
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904 (2020)

9. Geist, A.C.J.: Rechtsdatenbanken und Relevanzsortierung. Doctoral dissertation, uniwien, Austria, Vienna (2016)
10. Goebel, R., Kano, Y., Kim, M.Y., Rabelo, J., Satoh, K., Yoshioka, M.: COLIEE 2022: Methods for Legal Document Retrieval and Entailment (2022)
11. Grootendorst, M.: Keybert: Minimal keyword extraction with bert. (2020)
12. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 178–185 (2004)
13. Koopman, B., Cripwell, L., Zuccon, G.: Generating clinical queries from patient narratives: a comparison between machines and humans. In: Proceedings of the 40th international ACM SIGIR conference on Research and development in information retrieval. pp. 853–856 (2017)
14. Lastres, S.A.: Rebooting legal research in a digital age (2015)
15. 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. arXiv preprint arXiv:1910.13461 (2019)
16. Locke, D., Zuccon, G., Scells, H.: Automatic query generation from legal texts for case law retrieval. In: Asia Information Retrieval Symposium. pp. 181–193. Springer (2017)
17. Macdonald, C., Tonellotto, N.: Declarative experimentation ininformation retrieval using pyterrier. In: Proceedings of ICTIR 2020 (2020)
18. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Thirty-first AAAI conference on artificial intelligence (2017)
19. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
20. 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. arXiv preprint arXiv:1910.10683 (2019)
21. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
22. Sekulić, I., Soleimani, A., Aliannejadi, M., Crestani, F.: Longformer for ms marco document re-ranking task. arXiv preprint arXiv:2009.09392 (2020)
23. 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)
24. Verberne, S., Sappelli, M., Hiemstra, D., Kraaij, W.: Evaluation and analysis of term scoring methods for term extraction. Information Retrieval Journal **19**(5), 510–545 (2016)
25. Yang, E., Lewis, D.D., Frieder, O., Grossman, D.A., Yurchak, R.: Retrieval and richness when querying by document. In: DESIRES. pp. 68–75 (2018)
26. Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., Papadias, D.: Query by document. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining. pp. 34–43 (2009)
27. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. In: NeurIPS (2020)

Using Textbook Knowledge for Statute Retrieval and Entailment Classification

Sabine Wehnert[1,2], Libin Kutty[1], and Ernesto William De Luca[1,2]

[1] Otto von Guericke University Magdeburg, Germany
[2] Leibniz Institute for Educational Media | Georg Eckert Institute, Germany
sabine.wehnert@gei.de

Abstract. In this work, we imitate the process of a legal expert studying the situational application of statutes, in order to infer relevance and entailment relationships between a query statement and a statute. While using transformer-based architectures, we extract additional statute information from textbooks and incorporate this knowledge into the original pipeline. Results indicate that there is a benefit of using the textbook knowledge in Statute Retrieval and Entailment Classification tasks.

Keywords: information extraction · document enrichment · semantic similarity · sentence transformer · statute retrieval · entailment classification

1 Introduction

Developing Artificial Intelligence for the legal domain is challenging. From an outside view, legal jargon seems well-regulated and therefore predisposed to be formalized. However, imitating the reasoning of a human has been done so far only in a very limited scope and does not appear to be generalizable soon. How can we support lawyers though, given the ever-increasing amount of norms and jurisdictions they need to consider? Efforts in legal information retrieval shall help a lawyer to find relevant laws, whereas in the future classifiers for entailment may offer insights into scenarios contradicting a given set of laws.

Why are we not there yet? With the advent of transformer-based architectures in recent times, many Information Retrieval and Entailment Classification tasks have been practically solved. The amount of statistical information that those models encode in their high-dimensional embeddings has been shown to generalize well in many areas. Transformer models may perform well in situations, where semantic similarity is sufficiently captured because the involved terms are common enough. Nevertheless, daily tasks in the legal domain require more problem-solving capabilities than large open-domain models currently have. The knowledge that is required to understand statute relevance and entailment relationships can be highly specialized to a given jurisdiction and may not be present in the statutes themselves. Further domain knowledge can help to alleviate this problem. Thus, we focus on obtaining knowledge from textbooks and web resources about the situational use of statutes, similar to the knowledge

a legal expert may have acquired through years of study and practice. We study how this knowledge can be incorporated into transformer-based approaches and whether it offers a benefit in the Statute Retrieval and Entailment Classification tasks of the Competition on Legal Information Extraction and Entailment (COLIEE) 2022. Our contributions are:

- We use a rule-based approach to extract textbook knowledge about statutes.
- We employ this knowledge in our combined 2-stage Term Frequency - Inverse Document Frequency (TF-IDF) and sentence embedding approach for statute retrieval.
- We test different methods for encoding the external knowledge in a Graph Neural Network for entailment classification.

The remainder of this paper is organized as follows: In Section 2, we collect the related research to our approach of obtaining external knowledge about statutes from textbooks and similar documents. Section 3 contains more details about the employed rule-based knowledge extraction process. Our conceptual designs for the downstream tasks of statute retrieval and entailment classification with extracted knowledge are described in Sections 4 and 5, respectively. In Section 6, we present preliminary experimental results, as well as the competition results of our approaches, along with a discussion. The work is concluded in Section 7.

2 Related Work

Regarding related work, this section focuses on legal information extraction about statutes, since using textbook knowledge is the main contribution of this work.

In general, extracting information from textbooks is uncommon in the legal field. We often find researchers work on the statutes themselves to create citation networks [11] or on legal cases to detect references to statutes [16]. The work by Winkels et al. [16] is one of the approaches working on case law. They define so-called *reasons for citing* and extract keywords around a reference to a statute. This basic notion has been adopted in this research, as well. We regard the sentence containing a reference to an article as a possible source of useful information about the situational application of that article. In previous work, we developed a pipeline for extracting statute knowledge from textbooks with GATE[3] [12], and more recently using Apache UIMA Ruta[4] [13]. This work is based on a finite set of patterns that are specified to detect article mentions in the text and then to retrieve the surrounding sentence as potentially useful context information. In the following section, we explain how we made use of that approach for the COLIEE tasks.

[3] `https://gate.ac.uk/`
[4] `https://uima.apache.org/ruta.html`

3 Extracting Knowledge about Statutes from Textbooks

Statutes are written in an abstract manner in order to apply to several cases. Knowing in which case a statute is applicable requires domain knowledge, normally from legal domain experts. However, those experts also obtain their knowledge from sources, which may be partially available in digital format. In that regard, we propose an approach to extract auxiliary information for the civil code articles from textbooks to enrich the articles in subsequent processing steps. To this end, we obtained a collection of English textbooks, journal articles and web resources concerning the Japanese Civil Code:

- Journal Articles [1,10]
- Textbooks [5,9]
- Web Resources [2,3,4,6,8,17]

From this collection, we extracted 152 sentences containing references to articles of the Japanese Civil Code. The main motivation behind extracting those sentences is that they may contain some information about the application of a given article. Hence, if we are prompted with a query describing a special scenario, we assume that chances will be higher to detect the similarity between query and article in case we possess auxiliary information from the textbooks describing that situation in the query. Given our extracted knowledge, the gain we expect from adding this information shall be noticeable.

In the following, we first describe the process of extracting the knowledge about Civil Code articles from textbooks, before testing its benefit on the COLIEE tasks in the Sections 4 and 5. All aforementioned documents are initially obtained in a PDF format. We first perform some preprocessing steps:

1. Convert the PDF to txt format using `pdftotext`[5].
2. Remove hyphens at the end of each line and reconstruct the words that were separated by a hyphen.
3. Remove all line breaks.
4. If there is a table of contents, extract the chapter names. This is either done with `pdfminer`[6] (if there is sufficient metadata) or manually.
5. Parse the fulltext for the chapter titles from the table of contents and use the title strings to section the document hierarchically.
6. Apply the UIMA Ruta rules for detecting references to the Civil Code and extract them along with their surrounding sentence as the context.
7. Map the references and their context to the hierarchical segment they occur in and append the chapter names to the context sentence.
8. Assign the context sentences to the Civil Code articles and encode them according to the subsequent task's pipeline.

Regarding the UIMA Ruta rules, we form patterns based on regular expressions and part-of-speech tagged by the StanfordPosTagger in `DKPro`[7]. We provide

[5] `https://www.xpdfreader.com/pdftotext-man.html`
[6] `https://github.com/pdfminer/pdfminer.six`
[7] `https://dkpro.github.io/dkpro-core/`

some examples for the patterns that our rules capture in Table 1. While there is some variability across the documents regarding the way of referring to the Japanese Civil Code, the authors remain mostly consistent with their own citation style. This property limits the complexity of the patterns, such that the most complex references that we account for are itemizations of multiple articles. Although we do not formally evaluate the coverage of the patterns, manual inspections lead us to estimate an effectiveness of our rules in 90% of the references. Given the extra knowledge from textbooks and the web, we can encode it for the downstream tasks, as described in the following.

Table 1: References to Civil Code articles captured by the extraction rules

Pattern Name	**Example**
Standard	Article 1 (3) of the Civil Code
Abbreviated	Art. 97 CivC
Reversed	Civil Code Art. 120 (1)
Combined	Civil Code Articles 412 and 484
Itemized	Articles 647, 665, 671, 701 et al. of the Civil Code

4 Statute Retrieval Task

The basis for our retrieval approach is our last year's winning contribution in the same task [15]. In short, the approach is a combination of 2 stages of TF-IDF weighting and sentence embeddings, that are combined in a final relevance score R for ranking:

$$R{=}\text{TFIDF}_1{+}\text{TFIDF}_2 + S, \tag{1}$$

where TFIDF_1 refers to the first stage of TF-IDF, while TFIDF_2 is the second stage and S to the cosine similarity between article and query sentence embeddings. Previous experiments have shown that both TF-IDF stages together yield better performance than separately. TFIDF_1 refers to a TF-IDF weighting of articles which are enriched with metadata from the Japanese Civil Code structure (i.e., section titles). Also, relevant queries for the given article are appended to the original article text, based on the information that can be obtained from the COLIEE 2022 training dataset for Task 4 (Statute Entailment Classification). For this stage, we use sub-linear term frequency scaling and L2-normalization. Between this enriched article and a given query in the test data, we then compute the cosine similarity. For the second stage TFIDF_2, we also enrich the article with metadata, but not with relevant query information, and then also compute the cosine similarity between that enriched article and the query. For the last part of Equation 1, we enrich the articles with the same structural metadata as in TFIDF_1 and additionally crawled article commentary from open source

commentary[8]. All this article and query text is then encoded by a sentence transformer and the cosine similarity is determined. Please note that we do not include the relevant queries to a given article while computing S. Given TFIDF_1, TFIDF_2 and S, we add up all these similarity scores. For ranking, the score R is normalized by dividing the term by the maximum score for R. The new aspect which we add in this year is the injection of further external knowledge where sentence embeddings S are computed. Instead of simply appending more text to the article text, we divide the original sentence embedding similarity score S in two parts. At this point, the adapted relevance score R' is composed of:

$$R'=\mathrm{TFIDF}_1+\mathrm{TFIDF}_2 + S', \tag{2}$$

with the same two stages TFIDF_1 and TFIDF_2, and a new sentence embedding similarity score S'. It consists of the following parts:

$$S' = w \cdot \mathrm{S}_{\mathrm{old}} + (1 - w) \cdot \mathrm{S}_{\mathrm{book}} \tag{3}$$

This score consists of the previously used sentence embedding similarity score S_{old} and an additional sentence embedding similarity score called S_{book}. The latter score is computed from the cosine similarity between query and extracted information for a given article encoded by the sentence transformer. In case there are multiple sentences extracted for the same article (i.e., this article was referred to multiple times in the textbooks), we separately encode them with the sentence transformer and then average the embeddings. If there is no textbook information available for a given article, we use the original article text with structural metadata for computing S_{book}. We balance those two sentence embedding similarity scores by introducing a weighting parameter w between 0 and 1. High values of w result in a low influence of the book knowledge on the final relevance score R', and vice versa. To sum up, we incorporate the external knowledge from textbooks as a separate term in the relevance score, which is weighted depending on how much impact the external knowledge shall have on the sentence embedding similarity score S'. We show experimental results in Section 6.

5 Statute Entailment Classification Task

Our approach for the Statute Entailment Classification Task of the COLIEE competition is also based on one of the architectures we employed previously, the Graph Neural Networks (GNNs) [14]. Given the below-average results of last time for this architecture and our assumption that more encoded information could lead to better performance, we extend the GNN-based approach towards the textbook knowledge. The main idea of the original setup is to form a graph from the instances, such that we have a query node and one or more article nodes attached to it. We encode the queries and articles with a sentence transformer

[8] `https://ja.wikibooks.org/`

model. Thereby, we also use the structural metadata to enrich the articles, as in the Statute Retrieval task. Using a unidirectional message passing technique with an average aggregation function [7], we obtain query node embeddings with neighbor information from adjacent article nodes and pass this query node embedding to two linear layers and a final softmax classification layer. Different from the previous COLIEE edition, we now only train one GNN model and do not form an ensemble of models trained on different dataset splits anymore. We implemented three approaches to incorporate external knowledge:

- **GNN_ORIGINAL:** This approach is the basic architecture of the GNN as described above, without the use of any further external textbook knowledge.
- **GNN_CONCAT:** In this approach, we concatenate the article node embeddings with the additional embeddings obtained from the textbooks for a given article. If there is no reference in any textbook to a given article, we use the original article node embedding as a replacement.
- **GNN_AVG**: We average the sentence embeddings for the articles that we obtained in the GNN_ORIGINAL approach with the textbook knowledge for each article, which is also encoded by the sentence transformer model.

6 Evaluation

In this section, we evaluate our approaches for the Statute Retrieval and Entailment Classification tasks of COLIEE 2022. First, we describe the experimental setup, then we share results and finally, we discuss the models' performance. Regarding both tasks, we created internal dataset splits on the English version of the dataset. Our training split contains all provided training data, except for the query-article pairs starting with *R02-**, which form the validation split. Please note that for the Statute Retrieval task we do not train any machine learning model and only use the splits for parameter selection. However, for the Entailment Classification task, we train models on the training split and select the best performing models based on their performances on the training and validation split, without training any model on the validation data.

6.1 Statute Retrieval Task

Experimental Setup According to the COLIEE 2022 rules, the model performance on the statute retrieval task is decided by the F2-score. Therefore we performed our preliminary experiments based on this metric. Given the approach which we described in section 4, we aim to select two parameters through experiments: the sentence transformer model and the weight w which controls the influence of the textbook knowledge on the final relevance score R'. The results of the preliminary experiments are shown in Figure 1. The highest performance on the training and validation split is achieved by the model *"all-mpnet-base-v2"*, with a w value of 0.8 and 0.9, respectively. This indicates that the model benefits from a small influence of the textbook knowledge most, compared to

a w value of 1.0, where no external textbook knowledge is used and the performance drops slightly. Although the differences in the F2-scores are not large, the same trend can be observed for the model *"paraphrase-mpnet-base-v2"*, while this model reaches its best performance on the validation split on a w value of 0.6. The model *"paraphrase-distilroberta-base-v2"* performs slightly better than *"paraphrase-mpnet-base-v2"* on the training split, but worse on the validation split. Interestingly, *"paraphrase-distilroberta-base-v2"* does not benefit from external textbook knowledge at all. Its predecessor *"paraphrase-distilroberta-base-v1"* does neither benefit from the textbook knowledge and was outperformed on both splits by the other models. Given the large difference in performance between the training and validation split, we decided to use a w value of 0.7 for all three runs, and we submitted one run for each model, except *"paraphrase-distilroberta-base-v1"*.

Results The results of the three selected models are shown in Table 2. Our third run with the model *"all-mpnet-base-v2"* achieved the highest performance among our submitted runs with an F2-score of 77.9%. This result is not surprising, given the model's performance on the preliminary experiments. Overall, our best run achieved the third-highest rank in the competition. The best result was attained by the HUKB team with an F2-score of 82.04%.

Table 2: Results on the Statute Retrieval Test Data

Run	Model	F2-score	Precision	Recall
OVGU_run1	paraphrase-distilroberta-base-v2	0.7732	0.7745	0.7962
OVGU_run2	paraphrase-mpnet-base-v2	0.7718	0.7732	0.8008
OVGU_run3	all-mpnet-base-v2	**0.7790**	**0.7781**	**0.8054**

Given that there were only 152 collected article context sentences from textbooks and a much larger corpus of articles in the civil code, we examine the benefit of using the textbook knowledge. Upon further inspection, we noticed that there were not many matches between the articles we had textbook knowledge for and those which were considered relevant in the test data. In several cases, the relevant articles were retrieved regardless of the presence or absence of textbook knowledge. Overall, our retrieval system with its composed relevance score shows higher performance when using textbook knowledge, compared to setting the w value to 1.0. Surprisingly, this time the weighting mechanism itself has played an important role in the performance boost. As described before, in case where we have no textbook information, there will be only the original article with structural metadata encoded and compared to the query, resulting in the S_{book} score. Meanwhile, the S_{old} score uses the same articles and structural metadata, but also relevant queries to a given article based on the entailment label from Task 4. As a consequence, the weighting mechanism strengthened the influence of the original article and structural metadata and improved precision

in a situation where a large portion of articles was not enhanced by textbooks. In other words, the influence of the relevant queries from Task 4 was diminished and therefore, precision increased.

Despite this observation, there can be article context information from textbooks which is closer to a query than the original article text. One example for an article that got enriched with context information from one of our web resources by Dogauchi [2] is Art. 96 of the Civil Code:

"In a situation in which for example, as a result of a manifestation of intention made as a result of the fraud, the transfer of ownership of land from A to B is registered, there will be a need to protect C, a third party who purchases the land from B. Consequently, article 96 (3) of the Civil Code prescribes "The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge."

In this case, we see that there is a specific situation described with three acting persons (A,B and C). Similar types of queries exist in the COLIEE dataset. Consequently, we assume there could be a performance boost when we have more relevant articles enriched with information from textbooks. We find ourselves in a balancing act between introducing external information which may dilute the original article content and which can reduce precision one one hand, and using no external knowledge while risking that the article texts differ syntactically too much from the queries, making similarity-based approaches less effective.

6.2 Statute Entailment Classification Task

Experimental Setup The task of Statute Entailment Classification is a binary classification problem with "Y" for positive entailment and "N" otherwise. Accordingly, the evaluation metric is the model's accuracy. Because of the unstable model performance depending on the weight initialization in the training phase, we performed our preliminary experiments on four sentence transformer models with 10 different random seeds. Figure 2 depicts on violin diagrams how the model performs on the different seeds (each observation is a horizontal line within the "violin") and the respective training and validation split. We have three setups: Subfigure 2a shows how the GNN_ORIGINAL performs without any textbook knowledge. We observe that the models *"all-mpnet-base-v2"* and *"paraphrase-distilroberta-base-v2"* yield similar accuracy scores, however the former model exhibits a lower variance in performance on the validation dataset. Therefore, in this setup model *"all-mpnet-base-v2"* becomes run 1. Subfigure 2b shows how GNN_CONCAT performs. In this case, we also choose *"all-mpnet-base-v2"* for run 2 because its performance on the validation split is the best of all models while being mostly on par with the other models on the training split. Finally, we choose the model *"paraphrase-distilroberta-base-v2"* for run 3 from Subfigure 2c in the GNN_AVG setup, because it has many seeds performing well on the training data and also achieves peak performance on the validation data.

Results The result of our submitted runs is shown in Table 3. Different from what we expected, run 3 performed best among all our runs with the model *"paraphrase-distilroberta-base-v2"* and an accuracy of 57.8%. The best overall performance was achieved by the KIS team with an accuracy of 67.89%. Given this result, it is apparent to us that the external knowledge was helpful to boost performance, since our run 1 without textbook knowledge only achieved 47.71%, despite performing well in many cases on the preliminary experiments. We also attribute the performance of our third run to the model itself, because it was trained on paraphrase data and may be better suited for entailment classification. Compared to the previous year, we achieved our goal to boost the GNN performance, however this performance increase was not sufficient to outperform other approaches. To sum up, adding knowledge from a textbook to a GNN model architecture turns out to be beneficial, and the best performance was achieved when averaging the original article embeddings with the additional textbook embeddings for the respective article.

Table 3: Results on the Entailment Classification Test Data

Run	Sentence Transformer	GNN Model	Accuracy
OVGU_run1	all-mpnet-base-v2	GNN_ORIGINAL	0.4771
OVGU_run2	all-mpnet-base-v2	GNN_CONCAT	0.5321
OVGU_run3	paraphrase-distilroberta-base-v2	GNN_AVG	**0.5780**

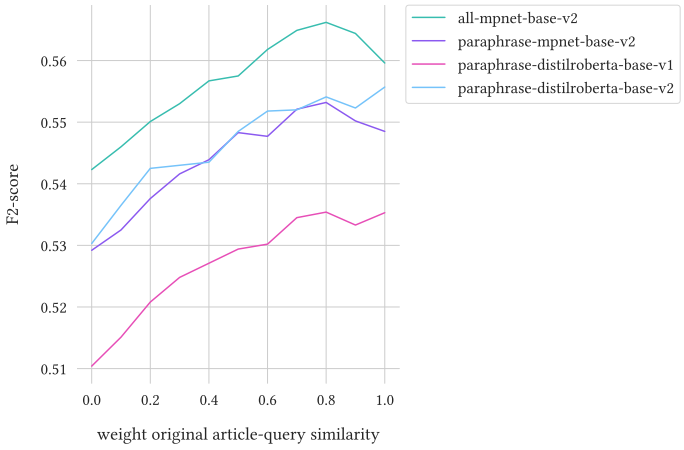
7 Conclusion and Future Work

We studied extracting and employing textbook knowledge about the statutes of the Japanese Civil Code for the Statute Retrieval and Entailment Classification tasks of COLIEE 2022. We described how we used a rule-based method to extract information about the situational application of the statutes from a collection of textbooks and web resources. Despite having limited textbook knowledge not covering all articles in the Japanese Civil Code, using our approach boosted the overall model performance. Future work can focus on using textbook knowledge in other setups, and can also elaborate on the content of the textbooks to explain the exact contribution of the involved textbook passages to the models' predictions when there are more enriched articles.

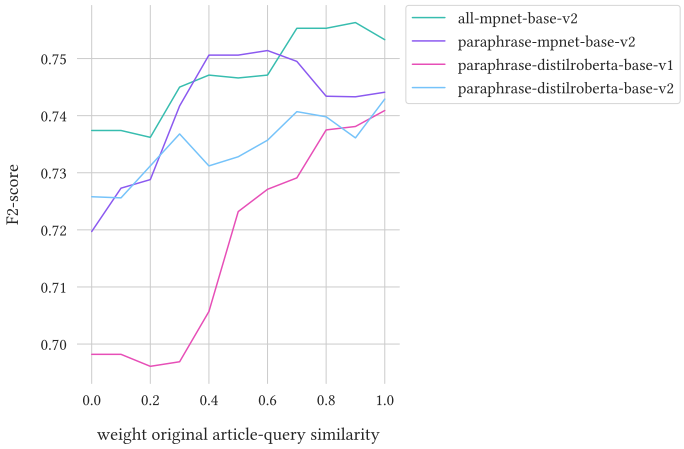
References

1. Danwerth, C.: Principles of japanese law of succession. Zeitschrift für Japanisches Recht **17**(33), 99–118 (2012)
2. Dogauchi, H.: Outline of contract law in japan, `http://www.law.tohoku.ac.jp/kokusaiB2C/overview/contract.html`

3. Hashimoto, Y.: Supreme court judgement on the constitutionality of article 750 of the civil code, which requires a husband and wife to adopt the surname of either spouse at the time of marriage (2017), https://www.waseda.jp/folaw/icl/news-en/2017/03/29/5720/
4. Iimura, T., Takayabashi, R., Rademacher, C.: The binding nature of court decisions in japan's civil law system (2015), http://cgc.law.stanford.edu/commentaries/14-Iimura-Takabayashi-Rademacher
5. Ishikawa, H.: Codification, decodification, and recodification of the japanese civil code. In: The Scope and Structure of Civil Codes, pp. 267–285. Springer (2013)
6. Kanazawa, T.: The case in which a part of article 733(1) of the civil code, stipulating a prohibition period for remarriage specific to women, was ruled as unconstitutional. (2017), https://www.waseda.jp/folaw/icl/news-en/2017/03/29/5722/
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
8. Shiraishi, D.: Payment of guarantee obligation, made after a guarantor has inherited principal obligation, can interrupt the extinctive prescription of the principal obligation (2014), https://www.waseda.jp/hiken/en/jalaw_inf/topics2014/precedent/001shiraishi.html
9. Steiner, K.: Postwar changes in the japanese civil code. Washington Law Review **25**(3), 286 (1950)
10. Taniguchi, Y.: The 1996 code of civil procedure of japan-a procedure for the coming century? The American Journal of Comparative Law **45**(4), 767–791 (1997)
11. Waltl, B., Landthaler, J., Matthes, F.: Differentiation and empirical analysis of reference types in legal documents. In: Legal Knowledge and Information Systems, pp. 211–214. IOS Press (2016)
12. Wehnert, S., Broneske, D., Langer, S., Saake, G.: Concept hierarchy extraction from legal literature. In: Cuzzocrea, A., Bonchi, F., Gunopulos, D. (eds.) Proceedings of the CIKM 2018 Workshops co-located with 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), Torino, Italy, October 22, 2018. CEUR Workshop Proceedings, vol. 2482. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2482/paper33.pdf
13. Wehnert, S., Durand, G.C., Saake, G.: ERST: leveraging topic features for context-aware legal reference linking. In: Araszkiewicz, M., Rodríguez-Doncel, V. (eds.) Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11-13, 2019. Frontiers in Artificial Intelligence and Applications, vol. 322, pp. 113–122. IOS Press (2019). https://doi.org/10.3233/FAIA190312, https://doi.org/10.3233/FAIA190312
14. Wehnert, S., Dureja, S., Kutty, L., Sudhi, V., De Luca, E.W.: Applying bert embeddings to predict legal textual entailment. The Review of Socionetwork Strategies pp. 1–23 (2022)
15. Wehnert, S., Sudhi, V., Dureja, S., Kutty, L., Shahania, S., De Luca, E.W.: Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. p. 285–294. ICAIL '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3462757.3466104, https://doi.org/10.1145/3462757.3466104
16. Winkels, R., Boer, A., Vredebregt, B., van Someren, A.: Towards a legal recommender system. In: JURIX. pp. 169–178 (2014)
17. Yamashiro, K.: The treatment of liability to refund overpayment in case of assignment of operating assets of the moneylender (2012), https://www.waseda.jp/folaw/icl/public/bulletin/back-number31-35/

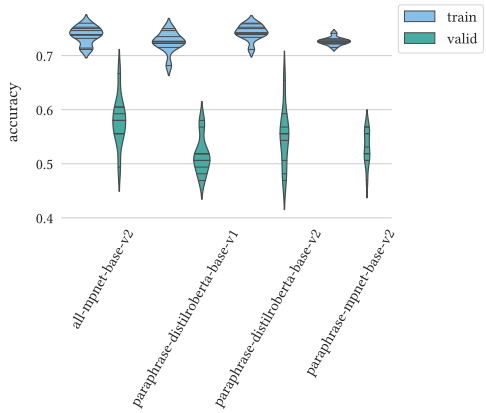


(a) training split

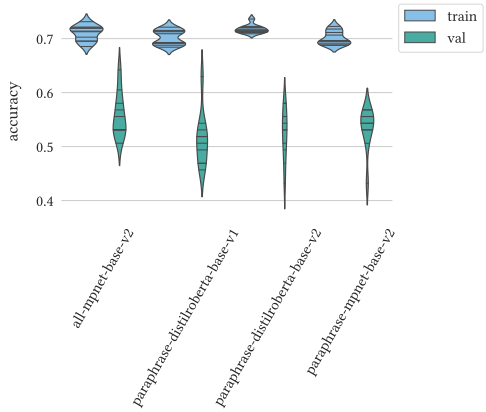


(b) validation split

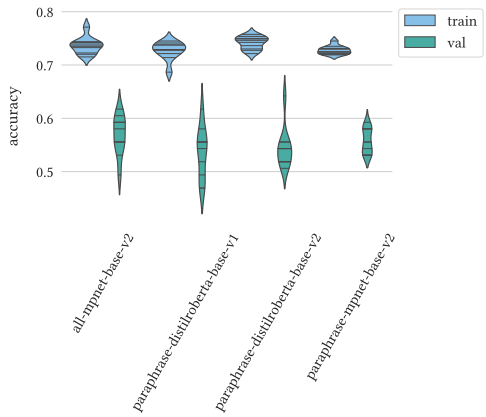
Fig. 1: Performance based on how the original article-query similarity is weighted. Higher weights indicate a lower contribution of the external data to the similarity.



(a) `GNN_ORIGINAL`: without external knowledge



(b) `GNN_CONCAT`: with external knowledge, concatenated with article embeddings



(c) `GNN_AVG`: with external knowledge, averaged with article embeddings

Fig. 2: GNN experiments on the training/validation splits over 10 random seeds.

Statute-enhanced lexical retrieval of court cases for COLIEE 2022

Tobias Fink, Gabor Recski, Wojciech Kusa, and Allan Hanbury

TU Wien, Faculty of Informatics, Research Unit E-commerce

Abstract. We discuss our experiments for COLIEE Task 1, a court case retrieval competition using cases from the Federal Court of Canada. During experiments on the training data we observe that passage level retrieval with rank fusion outperforms document level retrieval. By explicitly adding extracted statute information to the queries and documents we can further improve the results. We submit two passage level runs to the competition, which achieve high recall but low precision.

Keywords: Information Retrieval · Information Extraction · Legal Domain.

1 Introduction

In the legal domain, court cases play an unique role as they often contain the last say on a particular legal subject. This is especially true in Common Law systems, such as the legal systems of North America, where court cases play a large role in shaping the law. While statutes are the foundation of the legal system, it is often necessary to look through precedent court cases for detailed information that is not available in statutes to reach a decision. However, not only are court cases long and difficult to read, the number of potentially relevant court cases is ever increasing. As such, the need for development of automated methods for retrieval of legal information to aid legal experts is equally increasing.

The Competition on Legal Information Extraction/Entailment (COLIEE)[1] evaluates legal information retrieval (IR) systems for a variety of legal retrieval tasks. We participate in the COLIEE 2022 Task 1, which deals with Canadian law precedent retrieval (notice cases). We experiment with lexical methods for retrieval, focusing on ways of improving established methods with domain-specific fine-tuning. Considering that statutes are still the foundation of the legal system, we add statute information to the search to focus the models on information that is typically defining relevancy in the legal domain. Although not all cases contain statute information, we observe that making use of this information will overall improve retrieval performance.

[1] `https://sites.ualberta.ca/~rabelo/COLIEE2022/`

2 Task Description

In Task 1 of the COLIEE 2022, the goal is to retrieve supporting court cases (notice cases) for new court cases (query cases). Notice cases can be understood as precedent cases that are highly relevant for a query case. Each query case is supported by at least one notice case. For this task, cases from the Federal Court of Canada are used for both query and notice cases. A training collection as well as a test collection is provided (see Table 1), both having their own respective query cases, which are part of the collection. The training collection provides labels for relevant notice cases for each query, while the test collection only provides query cases without labels. Cases have been edited to have references to other cases removed and replaced by placeholder tokens. The task is to retrieve notice cases from the test collection using the queries of the test collection. The performance is measured using F1 score.

Table 1. Training and test collection statistics. Tokens per document and notice cases are per query.

	Training	Test
Total Cases	4415	1563
Query Cases	898	300
Max # of tokens	90567	61065
Median # of tokens	3658	3573
Mean # of tokens	4778	4979
Max # of notice cases	34	N/A
Median # of notice cases	3	N/A
Mean # of notice cases	4.68	N/A

The length of the query documents makes the task challenging in a few ways. Not only are many IR methods better suited for shorter queries, due to the length of the documents, the relationship between query cases and notice cases is also difficult to understand without expert knowledge.

3 Method

We approach this task with the assumption that there is a topical overlap between query and notice cases, but that not all parts of a query case are equally important. It has been shown in past legal retrieval workshops (see AILA [4,5], COLIEE [7,6,1]) that lexical methods, such as BM25 or IR language models (LM), yield competitive results, even when compared to newer neural network based approaches. We build on top of these lexical methods and adapt them to the legal collections of this task.

3.1 Document-Level.

First, we experiment with using the models out-of-the-box. We preprocess the training collection by removing special characters and tokens with two characters or less and index the documents using Elasticsearch. Numbers are also removed, except when they are part of a statute section citation. During indexing, the text is also lowercased, stemmed and stopwords removed, including the task-specific placeholder tokens. To convert case documents to queries, we try to extract the most informative terms from the case. As a naive approach for this term extraction, we calculate the TF-IDF score for each token in the query case and then use the top T tokens with the highest score as query terms. We compare the performance of the Elasticsearch implementations of BM25 and the LM Jelinek Mercer similarity [8], which calculate a score s for each document. For each query, 100 documents are retrieved and the precision, recall and F1 scores calculated for each rank. Although query cases are part of the collection, they are skipped during retrieval. We perform a random search to find the best hyperparameters for BM25 ($k1$, b) and LM Jelinek Mercer (λ) as well as T. While searching for the best hyperparameters, we only use the first 700 queries of the training collection (*training set*). We determine the best cutoff rank k using the F1 micro-averages for each rank. We use the remaining 198 queries (*dev set*) to evaluate the best hyperparameters and cutoff rank k.

3.2 Passage-Level.

Next, we experiment by changing the way how queries are created from query cases and change how documents are retrieved by using passage level retrieval. The information on where a passage starts and ends is already present in the case files and just needs to be utilized. Similar to the lexical baseline of [2], we split each case c in the collection C into passages $p_1, ..., p_{n_c}$ and index the passages instead of the whole case, using the same preprocessing method as before. Now, the score s is calculated for each passage instead of each document. A query case q is also split into passage queries $pq_1, ..., pq_{n_q}$ which retrieve a set of passage level rankings R with $|R| = n_q$. We aggregate the passage level rankings to case level using Reciprocal Rank Fusion, a method of aggregation that outperforms other methods, such as Condorcet Fuse or CombMNZ [3]:

$$RRFscore(c \in C) = \sum_{r \in R} \frac{1}{k_{rrf} + r(c)} * p_b \tag{1}$$

We set $k_{rrf} = 60$, the same value as in [3]. We also add a passage boost factor p_b that is set to 1 for now. For this passage level ranking approach, we again perform the same method as before to find the best hyperparameters for BM25 ($k1$, b) and LM Jelinek Mercer (λ) as well as T and k, using the same *training set / dev set* split of queries for evaluation. For all further experiments, we the values of the hyperparameters are fixed to the best result of this random search (excluding cutoff rank k).

3.3 Statute Field.

Finally, we experiment with adding additional domain knowledge to the search by extracting statute sections mentioned in the case documents and adding them to the documents explicitly. For this purpose, we scrape the titles of Canadian rules, regulations, orders and acts from the Canadian Justice Law Website[2]. This scraped list of titles also contains parts that would not be typically found in statute citations (e.g. text fragments that the law has been repealed). Consequently, we clean the titles by only considering text up to the first mention of *regulations*, *order*, *act* or *rules*. Further, since some statutes are only mentioned as acronyms, we create acronym candidates for each statute by taking the first upper-case letter of each token in the title. We identify the statutes of a case based on mentions of titles and generated title acronyms in the text. Additionally, we use regular expressions to detect statute section numbers in the text. We map statutes to section numbers by counting the number of passages in a case where a section number co-occurs with a statute mention, and then assigning the most frequently co-occurring statute to a section number.

These extracted statute sections are then added to the case passages and indexed as an additional statute-section field in Elasticsearch. We combine the original passage query with the extracted statute sections of the passage using a compound query. The score for the statute field $s_{statute}$ is calculated using BM25 and added to the overall score for each passage (the Elasticsearch default for a compound query), resulting in a new total score s_{total}:

$$s_{total} = s + s_{statute} * s_b \tag{2}$$

To further control the influence of the statute-section field on the similarity calculation, we adjust the weight of the similarity score of the statute-section field with the factor s_b (using the ElasticSearch *boost* functionality). We assume that query passages that mention statute-sections are more likely to contain information that is of particular importance for a case. If the number of statute-sections s_n that are present in the query passage is at least 1, we now set the earlier introduced passage boost factor

3.4 Document-Level.

First, we experiment with using the models out-of-the-box. We preprocess the training collection by removing special characters and tokens with two characters or less and index the documents using Elasticsearch. Numbers are also removed, except when they are part of a statute section citation. During indexing, the text is also lowercased, stemmed and stopwords removed,p_b to the hyperparameter P_b:

$$p_b = \begin{cases} P_b, & \text{if } s_n \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

[2] `https://laws-lois.justice.gc.ca/eng/`

The best values for P_b and s_b are determined using a random search and k is determined as before.

4 Results

Table 2. Results for our experiments using the *training set* and *dev set* queries.

	Training Set			Dev Set		
Method	Precision	Recall	F1	Precision	Recall	F1
Document level BM25 ($k = 7$)	0.1193	0.1944	0.1479	0.0871	0.1825	0.1179
Document level LM ($k = 8$)	0.1200	0.2200	0.1553	0.0802	0.1892	0.1127
Passage level BM25 ($k = 8$)	0.1214	**0.2226**	0.1571	0.0898	0.2116	0.1261
Passage level LM ($k = 8$)	0.1210	0.2218	0.1565	0.0993	**0.2341**	0.1395
Passage level LM + Statute Field ($k = 7$)	**0.1282**	0.2090	**0.1589**	**0.1073**	0.2249	**0.1453**

Table 3. Excerpt of the task 1 ranking showing selected runs and our results.

Rank	Team	Run	F1 Score	Precision	Recall
1	UA	pp_0.65_10_3.csv	**0.3715**	0.4111	0.3389
2	UA	pp_0.7_9_2.csv	0.3710	**0.4967**	0.2961
3	siat	siatrun1.txt	0.3691	0.3005	**0.4782**
7	LeiBi	run_bm25.txt	0.2923	0.3000	0.2850
15	TUWBR	TUWBR_LM_law	0.2367	0.1895	0.3151
17	TUWBR	TUWBR_LM	0.2206	0.1683	0.3199

The results of the experiments on the training set and dev set are shown in Table 2. On *document level*, *BM25* achieved the highest F1 using the parameters $T = 200$, $k1 = 1.09$, $b = 0.99$ while the *LM* Jelinek Mercer achieved the highest F1 using $T = 200$, $\lambda = 0.64$, $b = 0.99$. On *passage level*, *BM25* achieved the highest F1 using the parameters $T = 100$, $k1 = 0.66$, $b = 0.59$ while the *LM* Jelinek Mercer achieved the highest F1 without limiting T and $\lambda = 0.56$. This means *LM* uses every token of a passage as query, but with duplicate tokens removed. In our experiments, all passage level methods with rank fusion outperform document level retrieval methods. For this reason we did not continue experimenting on document level. While passage level BM25 achieved a higher F1 score on our training set, the LM model performed better on the dev set. The best overall F1 score was achieved by the LM model with inclusion of the statute field. Especially the dev set performance could be improved by adding this information.

For the task submission, we submitted two runs, using the *Passage level LM* setup as run **TUWBR_LM** and using the *Passage level LM + Statute Field* setup as run **TUWBR_LM_LAW**. The results for our submitted runs and a

selection of top scoring runs for the task are shown in Table 3. Our methods achieve a high level of recall but perform poorly regarding precision. However, our runs are only situated in the bottom half of the F1 score sorted ranking.

One weakness of our method is certainly that our naive term extraction approach was insufficient. Further, we were only able to produce a ranking of court cases and determined relevancy based on a fixed cutoff value (rank 7 or 8). Since most query cases cite fewer notice cases than our cutoff value, our precision is low. However, extracting statute-section information produced positive results. If we compare our two runs, we can see that adding statute information can yield a higher precision while recall is only reduced minimally. We expect that results can be improved further with better strategies for utilizing this information.

5 Conclusion

For the COLIEE 2022 Task 1 case retrieval, we used passage-level LMs to retrieve notice cases for case queries. Our methods achieved a high recall but low precision. We showed that a simple method making use of statute-section mentions in passages can achieve a higher precision with only a minor decrease in recall. Overall, low precision remained a problem for our methods and they were outperformed by other methods in the competition. We expect that our lexical approach could still be improved by different query term extraction strategies.

Acknowledgments. Project partly supported by BRISE-Vienna (UIA04-081), a European Union Urban Innovative Actions project.

References

1. Althammer, S., Askari, A., Verberne, S., Hanbury, A.: DoSSIER@ COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. arXiv preprint arXiv:2108.03937 (2021)
2. Althammer, S., Hofstätter, S., Sertkan, M., Verberne, S., Hanbury, A.: PARM: A Paragraph Aggregation Retrieval Model for Dense Document-to-Document Retrieval. arXiv preprint arXiv:2201.01614 (2022)
3. Cormack, G.V., Clarke, C.L., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 758–759 (2009)
4. Leburu-Dingalo, T., Motlogelwa, N.P., Thuma, E., Modongo, M.: UB at FIRE 2020 Precedent and Statute Retrieval. In: FIRE (Working Notes). pp. 12–17 (2020)
5. Liu, L., Liu, L., Han, Z.: Query Revaluation Method For Legal Information Retrieval. In: FIRE (Working Notes). pp. 18–21 (2020)
6. Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., Ma, S.: Retrieving Legal Cases from a Large-scale Candidate Corpus (2021)
7. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: Yes, BM25 is a Strong Baseline for Legal Case Retrieval. arXiv preprint arXiv:2105.05686 (2021)

8. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM SIGIR Forum. vol. 51, pp. 268–276. ACM New York, NY, USA (2017), issue: 2

Mining Data to Measure a "Fair Justice": A Conceptual Data-Driven Approach

João Araújo Monteiro Neto[1,2], Vasco Furtado[3,4], Fernando Siqueira[5], Francisco das Chagas Jucá Bomfim[6,7], André Câmara Ferreira da Costa[8] and Ana Lourdes Teixeira Matos[9]

[1] University of Fortaleza, Fortaleza, Ceará, 60.811-905, Brazil, joaoneto@unifor.br
[2] FUNCAP, Oliveira Paiva Avenue, 941, Fortaleza, Ceará, 60.822-130
[3] University of Fortaleza, Data Science and Artificial Intelligence Lab, Fortaleza, Ceará, 60.811-905, Brazil, vasco@unifor.br
[4] FUNCAP, Oliveira Paiva Avenue, 941, Fortaleza, Ceará, 60.822-130
[5] University of Fortaleza, Data Science and Artificial Intelligence Lab, Fortaleza, Ceará, 60.811-905, Brazil, fsiqueira@edu.unifor.br
[6] University of Fortaleza, Fortaleza, Ceará, 60.811-905, Brazil, fjuca@unifor.br
[7] FUNCAP, Oliveira Paiva Avenue, 941, Fortaleza, Ceará, 60.822-130
[8] FUNCAP, Oliveira Paiva Avenue, 941, Fortaleza, Ceará, 60.822-130, andrecamarafc@gmail.com
[9]University of Fortaleza, Fortaleza, Ceará, 60.811-905, Brazil, analourdestm@gmail.com

Abstract. The 2018 Brazilian Justice in Numbers Report shows that more Brazilian citizens are aware of their rights and, when necessary, access the judiciary system to guarantee protection and the enforcement of legal provisions. This phenomenon, also nominated as the increasing of judicialization in the Brazilian society [1]. The increased demand has been overloading the Brazilian Judicial system, what steamed the debate about the performance of the courts. Assessing and improving the quality of the Judiciary is an important element supporting the establishment of policies able to enhance justice distribution and Court's efficiency. One of the main challenges of evaluating the quality of justice-making activities is not only the conceptual variances of what is a good justice, but under a data driven approach, to mine the necessary data to measure key performance indicators. Data about courts functioning is normally structured in non-interoperable formats and are dispersed in different information systems used by the judiciary. Using a proof-of-concept approach and a qualitative-exploratory rationale the paper presents the processes used to extract and process data related to a group of indicators developed in the *Justiça Justa* (Fair Justice) working group of the research project Data Science and Artificial Intelligence in the Court of Justice of Ceará. The paper also explores alternatives for data visualization instruments and techniques able to promote the exploration and analysis of information that are new to Court Officers and legal scholars and has the potential to contribute to develop Court management as it can provide more accuracy to decision-making process regarding resources allocation and workload distribution between judges, contributing this way to data-driven and efficient Judiciary.

Keywords: Legal data mining, Justice assessment, Data-driven indicators.

1 Introduction

The debate about quality and efficiency in the Brazilian public services has been largely overlooked. Only with the enactment of the 1988 Brazilian Constitution the concerns about the quality of public services began to be raised and investigated [2]. Despite its initial cost-saving rationale, the topic was slowly introduced in the government agenda and progressively expanded its scope and maturity. In this context public agencies and bodies introduced monitoring processes and instruments to analyses the quality and efficiency of services and policies provided by governmental bodies [3]. Nevertheless, only recently the Judiciary Power and its operational actors (Courts, judges, public officers) were called to assess and better evaluate how Brazilian courts are delivering its main services, particularly distributing justice by processing legal cases.

This article explores preliminary findings from the *Justiça Justa* research activities. Developed under the research project Data Science and Artificial Intelligence in the Court of Justice of Ceará, Brazil, the research presents data mining and processing techniques that are being developed to evaluate how justice is being made in the State of Ceará judiciary system. It looks to two different novel elements: a) how to create and use data-driven indicators to measure and evaluate, in a multi-dimensional perspective, the activity of "doing justice" (processing of legal cases); and b) how to define and extract the correct data elements that can represent the processing of a legal case and uses this data to represent data-driven indicators of a fair justice system. One of the main challenges of evaluation process using data-oriented matrixes is to mine the necessary data to measure the indicators of fair and good Justice. Most of the data are not easy to identify on the several information systems used by the Brazilian Courts.

This paper presents the process performed by the research team to design, extract and process the different types of data used to represent one of the Fair Justice indicators developed to measure, under a data-drive rationale, the quality of justice-making in Brazilian courts. It also presents user-friendly alternatives for viewing and analyzing the data supporting the indicators and the levels of efficiency and fairness of the courts assessed. This is an important output as the ability to present complexes sets of data and information in simple and meaningful formats will have a stronger impact on the usefulness of the indicators as instruments to guide the implementation of strategic actions and resources allocation.

The study has a quantitative approach, making use of data from information systems that represent certain phenomena in their context. The research is classified as descriptive because it proposes to study and describe existing characteristics, properties, and relationships, aiming to identify representations and behaviors, as well as obtain information on a problem or question [4 – 6]

The first section introduces the conceptual debate about the assessment of quality and fairness in judicial processes, particularly presenting the rationale supporting the indicators developed. The second session presents the techniques used to mine and

process the data used to feed the indicators. The final session presents the main insights and impacts noticed during the use of data-driven indicators to evaluate the quality of Court´s justice-making processes.

2 Measuring the "fairness" of Justice – Conceptual notes

The assessment of public services and policies implementation effectiveness involves monitoring and evaluation. Qualified assessments, that produce reliable results, make possible to improve public services efficiency and justify investments or cost-saving measures. These monitoring activities show whether the expected results are being achieved or the inefficacy of resources allocation and use. In the context of public services, these analyzes play an essential role in determining the level of efficiency of public institutions responsible for providing services or operating public policies, like education, health, or justice distribution.

The literature dedicated to the study of quality standards and the effectiveness of the judiciary is notably inclined towards using a quantitative approach or, in an opposite way, is guided by the predominance of qualitative elements. The first approach privileges quantifiable elements that, in most cases, ignore aspects such as the complexity of the court case, the relevance and social impact of the legal case, and the legality and fairness of the decisions. On the other hand, measurements predominantly grounded on qualitative aspects are criticized for their high levels of subjectivism and for using criteria that are incapable of being generally measured and compared.

The adequate measurement of the quality of activities developed by the judiciary has significant social relevance and has a direct impact on economic and social aspects such as the modulation of the credit market, the level of contract enforcement, the protection of intellectual property, the distribution of social justice, and crime control [7-9].

The review of the literature on the evaluation of judiciary, not surprisingly, pointed to a positive correlation between efficiency (productivity) and quality of decisions [8,9]. These elements, embodied on the trust on the courts is materialized in aspects like outcomes predictability and timely decisions as important vectors of quality [10].

In the Brazilian context, the creation of a framework using data-driven indicators representing a "Fair Justice" necessarily involves the combination of these elements with other aspects established by the Brazilian Judiciary System, like the elements established by the National Council of Justice (CNJ) in the Ordinance 88 /2020 that instituted the Judiciary Quality Seal. The model used by the CNJ presents four dimensions that need to be assessed in other to evaluate the quality of Brazilian Courts: Governance, Productivity, Transparency and, Data and technology [11].

Understanding Fairness as one of the key elements supporting the activity of doing "Justice" Velazques et.al. [12] points that:

> "Justice means giving each person what he or she deserves or, in more traditional terms, giving each person his or her due. Justice and fairness are closely related terms that are often today used interchangeably. There have, however, also been more distinct understandings of the two terms. While justice usually has been used with reference to a standard of rightness, fairness often has been used with regard to an ability to judge without reference to one's feelings or interests; fairness has also been used to refer to the ability to make judgments that are not overly general but that are concrete and specific to a particular case. In any case, a notion of being treated as one deserves is crucial to both justice and fairness."

Observing these elements and the better practices revealed in the literature review, the research team developed an experimental matrix of indicators to assess the fairness of the justice-making practices of Courts in Brazil. It seeks to evaluate the quality of the judiciary based on the data produced by the judiciary itself. The matrix also respects the guidelines indicated by the World Bank [9] which indicates the need to observe the level of efficiency of case management practices, the level of automation of case-management practices, and the use of alternative conflict resolution methods as elements of a good and fair justice-making system.

The data-driven matrix combines three observation dimensions, Input (Access to justice), Throughput (Management of the decision-making process) and Output (Court decisions) and two factors, Efficiency (Performance) and Quality (Fairness). Within each dimension-factor, a set of indicators was elaborated to reflect different aspects of a "Fair Justice" (Table 1):

Table 1. Indicator´s matrix.

Dimension	Fator	Indicador
Input (Access)	Efficiency/Performance	I1 - Time to access courts
		I2 - Ratio of specialised judges
	Quality/Fairness	I6 - Early rulings percentage
		I7 - Pre-trial and ADRS mechanisms
Throughput (Judicial administration and judicial decision-making)	Efficiency/Performance	T1 - Ruling time (Between ingress and decision)
		T2 - Level of automatization
	Quality/Fairness	T6 - Number of contacts between judge and litigants
		T7 - Percentage of incidents, complains and inspection (CNJ, corregedoria)
Output (Delivery – outcomes)	Efficiency/Performance	O1 - Process duration
		O2 - Enforcement time
	Quality/Fairness	O6 - Appeals rate
		O7 - Ruling reformation rate

Once the preliminary evaluation matrix was developed, the team began to structure the process of gathering the data necessary to feed each indicator. The design of a methodology for collecting and validating the data is the main objective of this work. The mining process involved a series of actions that include the collection design, especially regarding data identification and characterization, as well as its relationship with the indicators to be measured.

3 Developing data-driven indicators of Justice "fairness"

In the next section, this paper presents the methodology developed to capture data related to the T1 indicator – Time for merit decision, as well as the techniques used to allow a better visualization and significance of the measurements performed.

The population of this work comprises data from the Justice Automation System (SAJ), which is responsible for managing the processing of judicial proceedings in the 1st and 2nd degrees of Ceará state justice. The SAJ database contains information that gives characteristics to the judicial processes. From this pool the research team selected the following categories: Case Number, Protocol Date, Procedural nature, Class of the case, Main subject, Indicator whether the case has free justice, Forum, Court, Municipality and Procedural Movements. The database provided by the Court of Justice of the State of Ceará (TJCE) contains 140,000 case records and 18 million records of procedural movements.

For the specification of the business rules of the T1 indicator (Time for decision on the merits), the Unified Procedural Tables (TPU) [13] of the Brazilian National Council of Justice (CNJ) were used as a basis for defining the codes that would be used as initial and final milestones of each step of process. This piece of information is labelled as process movement and is connected to different data sources and tables used to control cases management in Brazilian courts (see fig. 2)

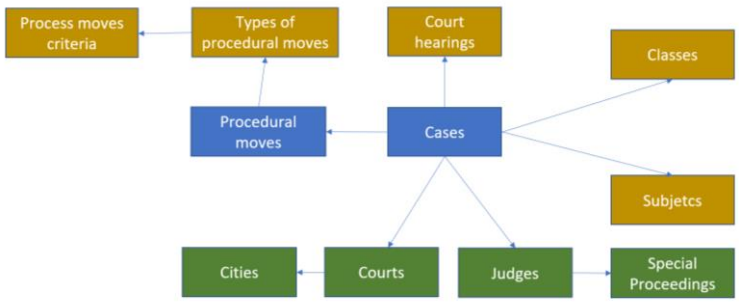


Fig. 2. Data sources

As mentioned earlier, the extraction of indicators is based on the table of unified codes (TPU) of the national justice council (CNJ), considering that all transactions in

the systems used by the courts, uniformly, use these codes in each procedural movement, you can use the movement codes to calculate the indicators. Therefore, it was defined as a starting point, the movement group of code 51 (conclusion) was used (see fig. 3) which records the presentation of the case to the Judge so that he can make his decision (see fig. 4). As a final milestone, we verified all processes that had transactions at a level lower than 385 (judgment with merit resolution), as well as sub-levels of codes 210 (concession), 214 (partial concession), 12433 (aggravation and special appeal set), 12326 (consultation), 212 (denial), 973 (extinction of punishment, 12450 (challenge to execution), 12661 (challenge of candidacy record), 12321 (opinion), 12678 (candidacy record). In this way, the period in days between the conclusion and the taking of the decision (Sentence, or interlocutory decision) can be established.

proc_ move_code	proc_move_detail	procedural move
51	40145	Concluso ao corregedor
51	40160	Concluso ao juiz corregedor
51	40162	Concluso ao presidente do órgão julgador
51	40163	Concluso ao relator
51	40164	Concluso ao revisor
51	40175	Concluso ao juiz convocado
51	40176	Concluso ao juiz
51	40161	Concluso ao presidente
51	40182	Concluso ao vice-presidente
51	50212	Concluso para Despacho
51	50213	Concluso para Sentença
51	50214	Concluso para Decisão Interlocutória

Fig. 3 Unified Procedural Tables (TPU) of procedural moves

Process id	date_proc_move_initial	#initial_seq	initial move code	initial_proc_move	date_proc_move_end	#end_seq	end_proc_move	#Days
3700003Q50000	02/09/2019	18	50.213	Concluso para Sentença	17/08/2021	37	Julgado procedente o pedido	715
1Z00006RB0000	08/10/2019	14	50.213	Concluso para Sentença	02/08/2021	53	Julgado procedente o pedido	664
0100102SI0000	19/08/2019	15	50.213	Concluso para Sentença	24/11/2020	38	Julgado improcedente o pedido	463
3E00002A60000	12/05/2020	15	50.213	Concluso para Sentença	11/08/2021	16	Julgado procedente em parte do pedi...	456
1U00002Z20000	21/01/2020	18	50.213	Concluso para Sentença	21/04/2021	21	Julgado procedente o pedido	456
010014CK00000	11/11/2019	15	50.213	Concluso para Sentença	08/02/2021	48	Extinto o processo por abandono da ...	455

Fig. 4. Sample of suits records to the judgement after data processing

Data processing was divided into three phases. Data cleaning, data extraction and the indicator generation (see Fig. 5). After the completion of the first two phases, the sample was produced with the 1st degree processes downloaded in 2019.

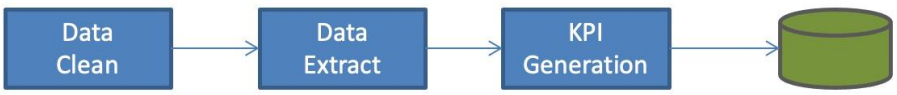


Fig. 5. Data Processing Flow

The first phase, data cleaning, detected and corrected (or removed) corrupted and inaccurate records. That is, it performed the activities of identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data. And then overwrite and delete the dirty or inconsistent data. This data cleaning can be performed either with the application of

data transformation tools or in batch processing through scripts (small data manipulation programs in a programming language). In this case, a mix of scripts and a data extraction, transformation, and load tool (ETL - Extract, Transform and Load were used.

Processes containing inconsistent values were found in the initial sample. Protocol dates with values before the mid-20th century and later. Procedural movements with date values like those of the process protocol, duplicate records, and others without specifying the type of movement were also found. To circumvent this situation, it was established to extract data only from processes from 2000 to 2020. Duplicate records were discarded, as well as records of procedural movements without indication of the type of movement. Finally, movement records that represented logically excluded records and were physically present in the database, these were also excluded in the cleaning process. All these cleaning activities were performed by scripts developed in the Python programming language.

This cleaning process resulted in the selection of 188,601 records of lawsuits downloaded in the period from 2018 to 2021 (out of a total of 440,000 from the entire SAJ database). The steps to perform this selection and the cleaning of inconsistent data were implemented in a series of scripts (computer programs) in the Python language. The processing of the execution of these scripts resulted in the scope of records of procedural movements for the next phase. In this case, there were 1,048,576 movement records, inputs for the data extraction phase of the data flow of this work.

The second phase comprised the extraction of the raw data needed to generate the indicators, the sample. This extraction was developed on the Tableau Prep Builder platform with the production of an ETL flow [14].

This flow represents the implementation of business rules defined by TJCE's legal and IT specialists, to produce the data required in the composition of the time indicator of the first judgment of the 1st processes. degree. The total flow of this phase comprises Figures 5 and 6, always starting from left to right. The first steps are the links between the auxiliary tables that make up the records of the processes, such as: class, subject, forum, court, and municipality - Part I (see Fig.6). Then, the result of Part I is combined with the records of procedural movements, already properly sanitized - Part II (Fig. 7).

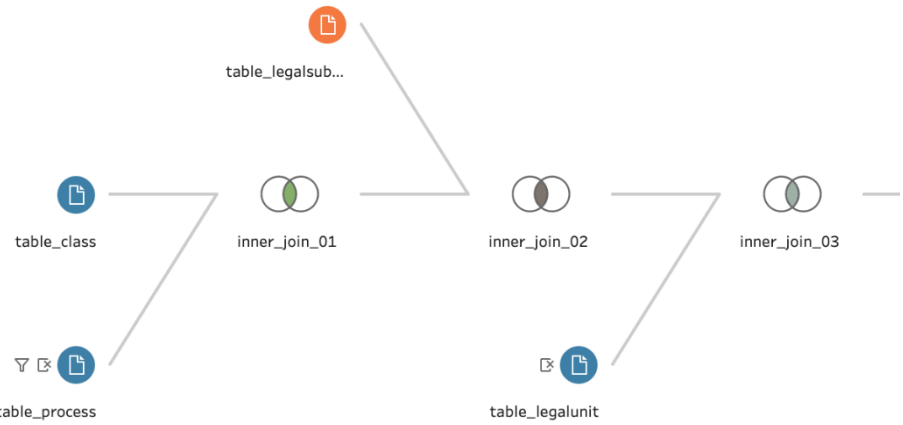


Fig. 6. Data Extraction Phase - Processes

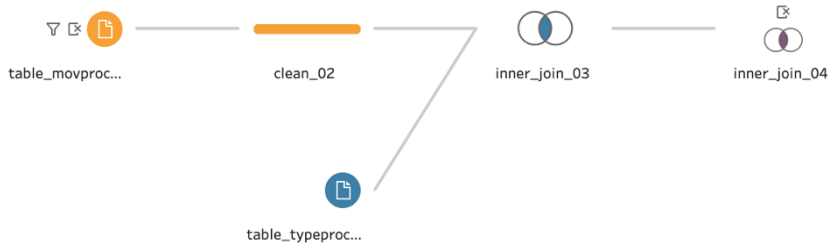


Fig. 7. Indicator Generation Phase 1

Finally, the data produced at the end of the ETL flow of the data extraction process (Part II) served as input to start the third phase illustrated in Figures 8 - Generation of Indicators. The indicator generation phase combines this data to obtain the 1st. procedural movement considered as the starting point for calculating the time of the 1st trial, as well as the movement that represents the final milestone. Once these records are obtained, the elapsed time is calculated and combined with other procedural data to compose the data set to be analyzed and explored through an analytical visualization tool. In the following section, more details about the analysis of these data will be described.

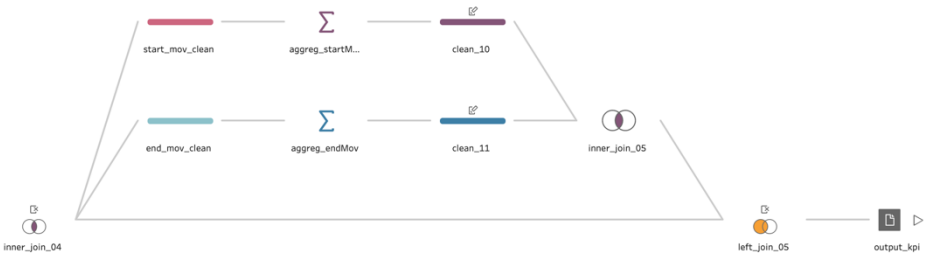


Fig.8. Indicator Generation Phase 2

4 Analysis of Results and Final Considerations

After completing the three phases of data processing, panels were built for the exploration and analysis of data by specialists in the legal field. These panels or *dashboards* were created using the Tableau data visualization tool, providing an interactive environment for obtaining insights (see Fig. 9).

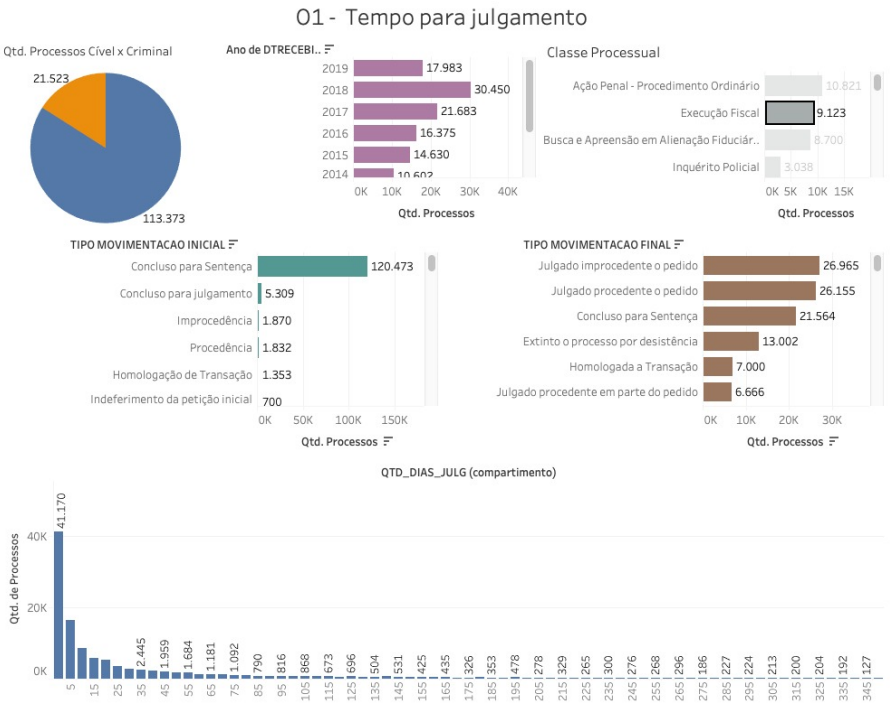


Fig. 9. Panel - Trial Times Indicator: From the left to the right the panel provides information about the nature of the court case (civil or criminal), the year of the case, the type of the case, and the number of days it took between being ready for judgment and the effective judgment.

The data provided on the dashboard reflects the indicator T1, which was designed to measure the time between a case is assigned to a judge and is final decision. In a clearer way the indicator measures what is called in the legal professions the trial length. This is an important piece of information that provides insights that could be used to analyses the case workload of each judge or Court, but also its efficiency. The dashboard presents the distribution of the cases by its nature (civil or criminal proceedings - first pie chart), and using a temporal rationale, the distribution cases over the years (bar chart). This information is very importan to develop a better resources allocation like clerks or even the use of artificial intelligence to automatize some internal.

The methodology developed to collect, process, and visualize the data presented, indicates not only the viability of using data-driven indicators that can evaluate multiple dimensions of a "Fair Justice", but also allows the access to crucial information much needed for guiding a more efficient, transparent, and fair Judiciary in Brazil

.

A good example is the observation of information provided in the indicator explored in this paper, the T1 or Length of Trial. The dashboard indicates that despite most of the cases are decided in short period of time, between 1 and 30 days, there is a high number of cases, approximately 8.000 take took between 35 and 100 days to be judged. Moreover, there is a long tail effect that creates court cases waiting to be judged for more than 180 days. This reveals the need to intervene and evaluate more carefully why these cases are waiting so much time do be decided, particularly assessing the need of workload distribution or staff allocation,

The process of extracting, sanitizing, and connecting the data, as presented in this paper, is a fundamental step in the development of automated routines for evaluating the levels of quality, efficiency and fairness of jurisdictional activities and must be increasingly encouraged, tested and developed.

References

1. Conselho Nacional de Justiça, Brasil.: Dados Estatíticos do Conselho Nacional de Justiça, Homepage https://www.cnj.jus.br/sgt/consulta_publica_classes.php, last accessed 2022/04/11.
2. Trevisan, P. A., Van Bellen, H. F. Avaliação de políticas públicas: uma revisão teórica de um campo em construção. Rev. Adm. Pública (42), 529-550 (2008). https://doi.org/10.1590/S0034-76122008000300005
3. Melo, M. A., Estado, governo e políticas públicas. In: Miceli, S. (Ed.). O que ler na ciência social brasileira, vol. 3, pp 59-100. ENAP, São Paulo (1999).

4. Creswell, J. W. W.: Projeto de pesquisa: Métodos qualitativo, quantitativo e misto. 2nd edn. Bookman, Porto Alegre (2010).
5. Collis, j., Hussey, r.: Pesquisa em Administração: Um Guia Prático para Alunos de Graduação e Pós-Graduação. 2nd. edn. Bookman, Porto Alegre (2005).
6. Yinn, R. K.: Estudo de Caso: Planejamento e Métodos. 3rd edn. Bookman, Porto Alegre (2005).
7. Palumbo, G., Giupponi, G., Nunziata, L., Mora-Sanguinetti, J.S.: Judicial Performance and its Determinants: A Cross-Country Perspective. OECD Economic Policy Papers, vol. 5. OECD (2013).
8. Castro, A. S., Indicadores básicos e desempenho da Justiça Estadual de Primeiro Grau no Brasil. IPEA, Brasília (2011).
9. World Bank. Making justice count: measuring and improving judicial performance in Brazil. Washington, DC, (2004).
10. Velasquez, Manuel., et.al. Justice and Fairness. Markkula Center for Applied Ethics website: https://www.scu.edu/ethics/ethics-resources/ethical-decision-making/justice-and-fairness/, last accessed 2022/04/01
11. Garoupa, N., Magalhaes, P. C.: Public trust in the European legal systems: independence, accountability, and awareness. West European Politics, 44(3), 690-713 (2021).
12. Brasil, Conselho Nacional de Justiça. Portaria 88/2020. Avaliable at https://www.cnj.jus.br/wp-content/uploads/2020/06/Portaria-88-GP-2020.pdf, last accessed 2022/04/01.
13. Brasil, Conselho Nacional de Justiça. Sistema de Gestão de Tabelas Processuais Unificadas. Avaliable https://www.cnj.jus.br/programas-e-acoes/priorizacao-do-1o-grau/dados-estatisticos-priorizacao, last accessed 2021/04/01.
14. Costello, T. K., Blackshear, L.: Prepare Your Data for Tableau: A Practical Guide to the Tableau Data Prep Tool. Apress, Berkeley, (2000).

Differential-aware Transformer for Partially Amended Sentence Translation

Takahiro Yamakoshi[1,3], 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
[3] Legal AI Inc.
JP Tower Nagoya 23F, Meieki 1-1-1, Nakamura-ku, Nagoya, 450-6323, Japan
Email: yamakoshi@legalai.jp

Abstract. We propose a Transformer-based differential translation architecture that targets statutory sentences partially modified by amendments. In translating post-amendment statutory sentences, translation *focality*—modifying only the amended expressions in the translation and retaining the others—is important to avoid misunderstanding the amendment's contents. To sharpen the focality of translation, we introduce a neural network architecture called a *Copiable Translation Transformer* that can copy expressions in the pre-amendment translated sentence as needed and generate expressions from the post-amendment original sentence. In experiments, we showed that our method outperformed the naive Transformer with a training corpus of partially amended sentences.

Keywords: Differential Translation, Amendment, Transformer, Copying Mechanism

1 Introduction

In the world's globalized society, governments must quickly publicize their statutes to facilitate international trade, financial investments, legislation support, and so on. In April 2009, the Japanese government addressed this issue by launching the Japanese Law Translation Database System (JLT) [10] where it publicizes English translations of its statutes. After amending a statute, its translation must be promptly updated to avoid confusion among foreigners about whether it remains in effect. Unfortunately, the pace of translation work lags behind statutory amendments. According to Yamakoshi et al. [15], as of January 2020, only 23.4% (163/697) of the translated statutes in JLT correspond to their latest versions. On the other hand, translating post-amendment statutory sentences is crucial because most enacted statutes in Japan's legislative bodies are amendment laws. For example, 78% (73/94) of the acts enacted in 2019 are amendment laws.

Updating the translations of statutory sentences according to an amendment resembles a differential translation, where we must consider the *focality* of translations [14]. Focal translations modify the expressions related to the amendment and keep those not related to it. For example, consider the following sentence: "申立ては、事故の事実を示

して、書面でこれをしなければならない。” (The request shall be made in a document stating the facts of the accident.) Its amendment changed “事故” (*jiko*; accident) to “海難” (*kainan*; marine accident). The following revision satisfies the focality requirement: “The request shall be made in a document stating the facts of the marine accident” because it contains minimum modifications. On the other hand, although “The petition shall be made in a document describing the facts of the marine accident” is a fluent and adequate translation, it is unsuitable as a revision from the focality perspective because “申立て” (*moshitate*; request) and “示して” (*shimeshite*; stating), which are irrelevant to the amendment, were changed.

Studies on machine translation methods retain the focality of post-amendment statutory sentences. Kozakai et al. [6] proposed a method for post-amendment Japanese statutory sentences based on a statistical machine translation (SMT) method from Koehn et al. [5]. These methods are template-aware: they can reuse expressions in a given target-language sentence (translation template). Yamakoshi et al. [14] proposed a machine translation method that generates translation candidates by Transformer [11], a neural machine translation (NMT) architecture, and selects the best one by comparing the candidates with the output of a template-aware SMT model (e.g., Koehn et al. [5] and Kozakai et al. [6]). The combination of an NMT model and a template-aware SMT model complements the weaknesses of each one: an NMT model can output more fluent sentences than an SMT model; a template-aware SMT model can keep expressions that are common in the amendment, which a naive Transformer cannot do. However, the latter advantage is weak in the method because it just chooses a candidate from a naive Transformer and does not postedit it. That is, when the Transformer model did not output an expression that should be retained in any candidate, the expression cannot be salvaged even if the template-aware SMT model successfully translated it.

In this paper, we propose a new NMT architecture for partially amended Japanese statutory sentences and address the issue in Yamakoshi et al.'s method. We call our architecture *Copiable Translation Transformer*. Our NMT architecture is based on the naive Transformer in terms of using an attention mechanism. The differences are that it receives a pre-amendment translated sentence with a post-amendment original sentence and copies a word in the pre-amendment translated sentence or generates a word for each word output. This mechanism resembles a copying mechanism for text summarization (e.g., Gu et al. [1] and Xu et al. [13]). However, their mechanisms are for monolingual sentence generation, denoting that it takes a single sentence and copies words from that sentence. On the other hand, our mechanism is for bilingual text generation, denoting that it takes two sentences of different languages—a post-amendment original sentence and a pre-amendment translated sentence—and copies the words from the latter.

This paper contributes to the post-amendment statutory sentence translation task by proposing a Transformer-based machine translation architecture for the task and describing its performance. Although we apply our Copiable Translation Transformer to the Japanese-English translation, it can be used with any language pairs.

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

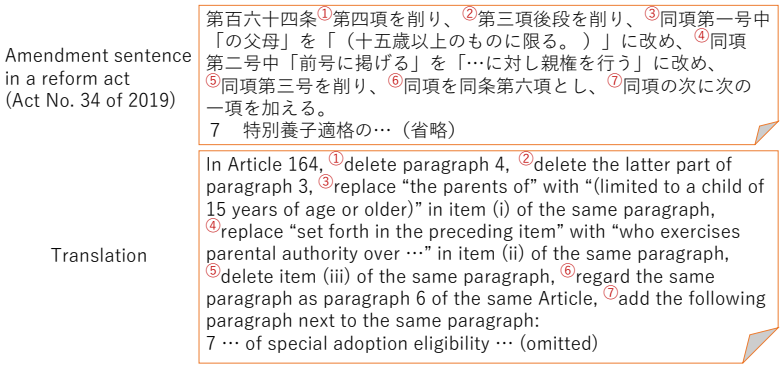


Fig. 1. Amendment sentence

and describe our neural architecture in Section 4. Sections 5 and 6 present our evaluation experiments and discussions. Finally, we summarize and conclude in Section 7.

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 identify the amendment process as a differential translation task.

2.1 Partial Amendments in Japanese Legislation

In Japanese legislation, partial amendments are done by “patching” modifications to the target statute, prescribed as amendment sentences in a reform statute. Such modifications can be categorized as follows [8]:

1. Modification of part of a sentence: (a) replacement, (b) addition, and (c) deletion.
2. Modification of such statute structure elements as sections, articles, items, etc.: (a) replacement, (b) addition, and (c) deletion.
3. Modification of element numbers: (a) renumbering, (b) attachment, and (c) shift.
4. Combined modification of an element’s renumbering and a replacement of its title string.

In modifying part of a sentence, the Japanese legislation rules [3] indicate 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 its seven modifications can be assigned to one of the above categories:

- 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. (a);

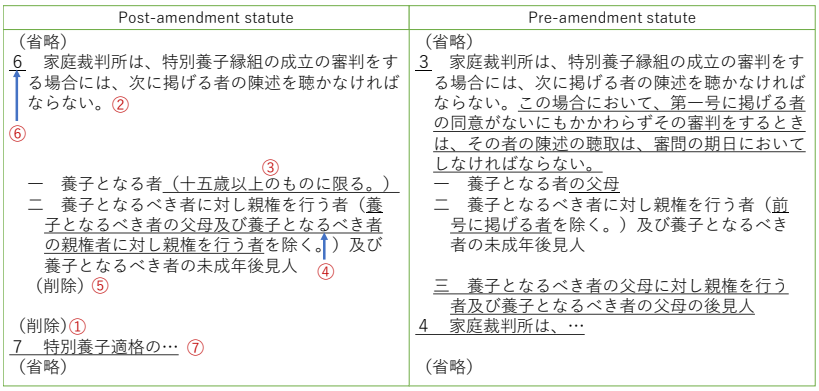


Fig. 2. Comparative table: red circled numbers correspond to modifications described in Fig. 1

– Modification ⑦ belongs to 2. (b).

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

2.2 Differential Translation for Partial Amendments

We focus on the sentence-level differential translation task proposed by Yamakoshi et al. [14]. It targets category 1 (i.e., replacement, addition, and deletion of part of a sentence), described in the previous section. In the case of Fig. 1, our targets are modifications ③ and ④. It also targets 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 obviously affect the main sentence. For example, modification ② in Fig. 1, which removed the latter part, is such a case.

Figure 3 illustrates this task. It involves four sentences from bilingual documents paired by two amendment versions. It 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 by the original sentence called *a post-amendment translated sentence*.

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 and help international readers understand them. Second, since the expressions in pre-amendment translated sentences are reliable, reusing them ensures 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 the other expressions that 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*

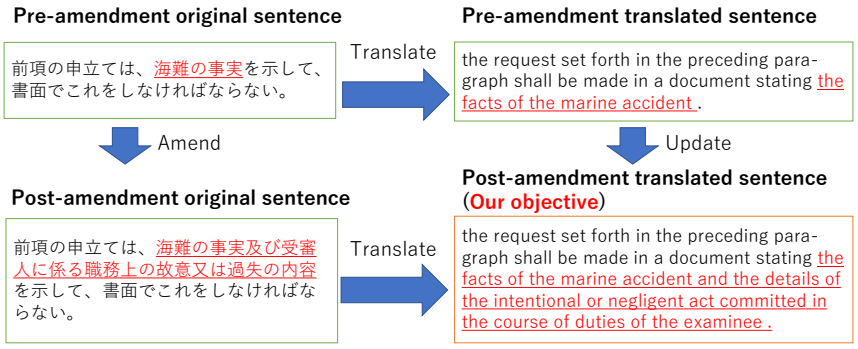


Fig. 3. Differential translation in an amended statutory sentence

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 summarize our task as follows:

Input:
- Pre-amendment original sentence S_{PrO};
- Post-amendment original sentence S_{PoO};
- Pre-amendment translated sentence S_{PrT}.

Output: Generated post-amendment translated sentence $\hat{S}_{\mathrm{PoT}}$.

Requirements:
- focality: $\hat{S}_{\mathrm{PoT}}$ should be generated based on the expressions of S_{PrT}, and $\hat{S}_{\mathrm{PoT}}$ should exactly reflect the amendment from S_{PrO} to S_{PoO};
- fluency: $\hat{S}_{\mathrm{PoT}}$ should have natural phrasing and syntax;
- adequacy: $\hat{S}_{\mathrm{PoT}}$ should contain the contents in S_{PoO} without excesses or inadequacies.

3 Related Work

In this section, we review related work. We respectively introduce methods, evaluation criteria, and corpora suitable for difference translations in Sections 3.1, 3.2, and 3.3.

3.1 Translation Methods

Koehn et al. [5] proposed a statistical machine translation (SMT) based method that can be applied to differential translation. The method can reuse expressions in a sentence chosen from translation memory (TM). It determines common expressions between the input sentence and the TM original sentence using an edit distance algorithm. Such common expressions are retained in the output sentence, while only other expressions are generatively translated by an SMT system, such as Moses [4]. Kozakai et al. [6] 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 TM. Second, to determine the common expressions, they used underlined information in the comparative table to highlight the differences between pre- and post-amendment sentences chunk by chunk 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. These above template-aware SMT-based methods can retain unchanged expressions well, although they cannot fluently generate translations because of the limitations of SMT. The fluency problem can be solved by using a neural machine translation (NMT) such as Transformer [11].

Some Transformer-based methods introduce additional components to utilize expressions in target language sentences that are suitable for differential translation. Xia et al. [12] proposed a method that incorporates the Transformer network with TM by embedding TM-reference sentences using a graph network. However, since the method does not have a mechanism to output the input as it is, it may make unnecessary modifications to unchanged expressions. Yamakoshi et al. [14]'s method combined a template-aware SMT with Transformer. It generates translate candidates by Transformer and chooses the best candidate by comparing them with the output of a template-aware SMT method. It tries to keep as many unchanged expressions in the pre-amendment translated sentence as it can. However, it cannot salvage such expressions if the Transformer model did not output them in any candidate.

The copying mechanism for neural networks offers the choice to directly output input words. Gu et al. [1] proposed this mechanism with the usage in the sequence-to-sequence NMT. Xu et al. [13] utilized it in the Transformer architecture. However, because these methods are designed for text summarization, they assume that input and output sentences share a language. Therefore, we cannot directly use the copying mechanism for such translations as our task.

3.2 Evaluation Criteria

As discussed in Section 2.2, fluency, adequacy, and focality are the three requirements of differential translation quality. For fluency and adequacy, we can use the standard criteria for machine translation, including BLEU [9] and RIBES [2]. However, these criteria are not sensible to focality. To evaluate focality, Yamakoshi [14] proposed a focality score that uses n-gram recall between a pre-amendment translation and the system output. Yamakoshi [15] proposed an integrated criterion for differential translation named Inclusive Score for DIfferential Translation (ISDIT). ISDIT consists of two factors: (1) the n-gram recall of expressions unaffected by the amendment and (2) the n-gram precision of the output compared to the reference.

3.3 Corpora

Kozakai et al. [6] used JLT bilingual resources to compile corpora for their experiment. For training data, they gathered 158,928 Japanese-English sentence pairs from 407 statutes provided in JLT. For test data, they selected 17 amendments available in JLT [4] from which they compiled 158 examples of sentence amendments, each of which

[4] JLT has a function for browsing statutes and the translations of different amendment versions.

consists of quartets (S_{PrO}, S_{PrT}, S_{PoO}, S_{PoT}). However, some of these examples are not focal because they contain modifications irrelevant to the amendment.

Yamakoshi et al. [15] compiled a corpus of focal quartets by the following procedure. First, they listed versions of statutes in JLT and e-LAWS[5]. Then they chose statutes whose JLT version lags behind its e-LAWS version and collected sentence-level amendments for such statutes. Finally, they manually translated the post-amendment sentences so that they retained their focality. Their corpus consists of 1,483 differential translation examples from 62 amendment cases.

4 Proposed Architecture

In this section, we describe our proposed architecture, *Copiable Translation Transformer*, which expands the Transformer network so that it can copy expressions in pre-amendment translated sentences. Copiable Translation Transformer has a copying mechanism that takes two sentences of different languages: the post-amendment original sentence S_{PoO} and the pre-amendment translated sentence S_{PrT}. That is, it requires a dataset with triplets (S_{PrT}, S_{PoO}, S_{PoT}) for training, where S_{PrT} and S_{PoO} are the input to the system, and S_{PoT}, the post-amendment translated sentence, is the answer. Copiable Translation Transformer does not use pre-amendment original sentences S_{PrO}, following the insights of Xia et al. [12]: The source-side TM-reference sentence (S_{PrO} in our task) is not so informative because most words in it also appear in the input sentence (S_{PoO} in our task), and ignoring the former sentence reduces memory and time consumption.

Figure 4 shows the network connection of the Copiable Translation Transformer. It generates i-th word output's representation R^i_{PoT} by decoding the words of the post-amendment translated sentence that already output $\hat{S}^{<i}_{\mathrm{PoT}}$ using encoded representation of post-amendment original sentence S_{PoO}:

$$R^i_{\mathrm{PoT}} = \mathrm{decoder}(\mathrm{encoder}_{\mathrm{PoO}}(S_{\mathrm{PoO}}), \hat{S}^{<i}_{\mathrm{PoT}}), \tag{1}$$

where R^i_{PoT} is a vector with dimension d. From R^i_{PoT}, it generates the word likelihood over vocabulary V that we call the *word generation likelihood* $\boldsymbol{l}^i_{\mathrm{gen}}$:

$$\boldsymbol{l}^i_{\mathrm{gen}} = \mathrm{ff}(R^i_{\mathrm{PoT}}), \tag{2}$$

where $\boldsymbol{l}^i_{\mathrm{gen}}$ is a vector with dimension of vocabulary size $|V|$ and ff is a feed-forward module. These steps are identical to those of the naive Transformer.

The Copiable Translation Transformer additionally encodes pre-amendment translated sentence S_{PrT} to earn its word representations R_{PrT}:

$$R_{\mathrm{PrT}} = \mathrm{encoder}_{\mathrm{PrT}}(S_{\mathrm{PrT}}), \tag{3}$$

where R_{PrT} is a matrix with dimension $d \times |S_{\mathrm{PrT}}|$. It then calculates the word likelihood over S_{PrT} that we call the *word copy likelihood* $\boldsymbol{l}_{\mathrm{copy}}$ by multiplying R_{PrT} and R^i_{PoT}:

$$\boldsymbol{l}^i_{\mathrm{copy}} = R^i_{\mathrm{PoT}} R_{\mathrm{PrT}}, \tag{4}$$

[5] `https://elaws.e-gov.go.jp/` The e-Legislative Activity and Work Support System (e-LAWS) provides an open governmental database of the most recent, original national statutes (i.e., written in Japanese).

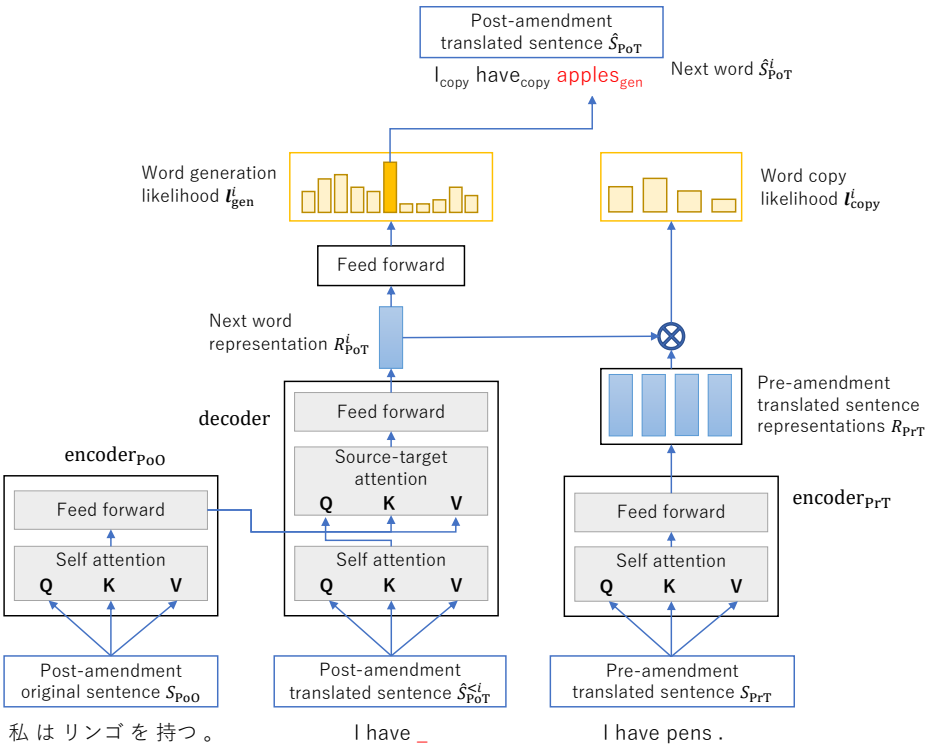


Fig. 4. Network connection of Copiable Translation Transformer

where $\boldsymbol{l}^i_{\mathrm{copy}}$ is a vector with dimension $|S_{\mathrm{PrT}}|$. Finally, it chooses next word $\hat{S}^i_{\mathrm{PoT}}$ with the maximum likelihood among $\boldsymbol{l}^i_{\mathrm{gen}}$ and $\boldsymbol{l}^i_{\mathrm{copy}}$:

$$x = \arg\max[\boldsymbol{l}^i_{\mathrm{gen}}; \boldsymbol{l}^i_{\mathrm{copy}}], \tag{5}$$

$$\hat{S}^i_{\mathrm{PoT}} = \begin{cases} x\text{-th word in } V & (\text{if } x \leq |V|) \\ (x-|V|)\text{-th word in } S_{\mathrm{PrT}} & (\text{otherwise}) \end{cases}, \tag{6}$$

where x is the index of the maximum value in concatenated vector $[\boldsymbol{l}^i_{\mathrm{gen}}; \boldsymbol{l}^i_{\mathrm{copy}}]$. If a word from the vocabulary is selected, then it *generates* the word; otherwise, it *copies* it.

5 Experiment

Next we experimentally evaluated the effectiveness of our method.

5.1 Outline

We built training and test datasets from a focal differential translation corpus [15] with 1,483 differential translation examples from 62 amendment cases. The corpus contains

many quartets (S_{PrO}, S_{PrT}, S_{PoO}, S_{PoT}), where we used S_{PrT}, S_{PoO}, and S_{PoT} for the Copiable Translation Transformer and S_{PoO} and S_{PoT} for the naive Transformer. We divided the corpus into 1,150 examples (46 amendment cases) for the training dataset and 201 examples (eight amendment cases) for the test dataset[6]. To evaluate the effect of augmenting the training dataset, we newly compiled a focal differential translation corpus with 1,670 examples (34 amendment cases) with the same protocol as Yamakoshi et al. [15] and used these examples as an additional training dataset.

We compared the performance of our Copiable Translation Transformer and the naive Transformer [11] with/without the additional training dataset. That is, we have the following options:

1. Naive Transformer without the additional training dataset
2. Naive Transformer with the additional training dataset
3. Copiable Translation Transformer without the additional training dataset
4. Copiable Translation Transformer with the additional training dataset

We used these architectures under the following settings: six encoder/decoder hidden layers, eight attention heads, 512 hidden vectors, a batch size of eight, a dropout rate of 0.1, and an input sequence length of 256. For all the models, we set the iteration number to 20,000. We used SentencePiece [7] as a tokenizer and set the vocabulary size to 8,192. We implemented the training and prediction codes based on the TensorFlow official model [7].

We combined those NMT models with Kozakai et al.'s template-aware SMT model (hereinafter "Kozakai model") as Yamakoshi et al. [14] proposed: Each NMT model outputs 4-best candidates from which we chose the best candidate by calculating the BLEU scores between a candidate and the output from the template-aware SMT model. For simplification, we do not apply Monte Carlo dropout, the candidate augmentation trick, unlike their proposed method.

We evaluated the fluency and adequacy with BLEU and RIBES. For the focality evaluation, we utilized ISDIT [15]. We set maximum n-gram length N to 4 for calculating ISDIT and BLEU.

5.2 Results

Table 1 shows the experimental results. "Naive," "Copiable, " and "Kozakai" in the table indicate naive Transformer, Copiable Translation Transformer, and Kozakai models, respectively. The Copiable Translation Transformer-based methods achieved better performance in all of the evaluation criteria. With the additional dataset, the Copiable Translation Transformer one is superior to the naive Transformer one by 31.80 BLEU, 33.20 RIBES, and 37.74 ISDIT points.

The additional dataset also raised its performance. For example, the Copiable Translation Transformer with the addition dataset achieved better performance than without it by 6.49 BLEU, 4.41 RIBES, and 12.79 ISDIT points. Hereinafter, we discuss and inspect the models trained with the additional dataset.

[6] We kept the remaining 132 examples (eight amendment cases) for a development dataset for future use.

[7] `https://github.com/tensorflow/models/`

Table 1. Experimental results

Model	BLEU	RIBES	ISDIT
Naive + Kozakai w/o additional dataset	18.40	50.75	8.37
Naive + Kozakai w/ additional dataset	22.32	53.89	10.52
Copiable + Kozakai w/o additional dataset	47.63	82.68	35.47
Copiable + Kozakai w/ additional dataset	**54.12**	**87.09**	**48.26**

Table 2. Successful case

Model	Output
(Sentence S_{PrO})	第百四十四条の三第五項において準用する第百二十四条の規定による手続が終了した日
(Sentence S_{PrT})	the day that the process under article 124 as applied mutatis mutandis pursuant to article 144-3 , paragraph (5) is complete ;
(Sentence S_{PoO})	第百四十四条の三第六項において準用する第百二十四条の規定による手続が終了した日
(Sentence S_{PoT})	the day that the process under article 124 as applied mutatis mutandis pursuant to article 144-3 , paragraph (6) is complete ;
Naive + Kozakai	the day on which the procedure under the provisions of article 13-3 , paragraph (6) of the companies act as applied mutatis mutandis pursuant to article 156
Copiable + Kozakai	the day that the process under 124 as applied mutatis mutandis pursuant to article 144-3 , paragraph (6) is complete .

Table 2 shows a successful case. This example contains a modification of an element number: “paragraph (5)” to “paragraph (6)”. The Copiable Translation Transformer one copied most of the unchanged expressions and modified the paragraph number. The naive Transformer one output such irrelevant expressions as “the companies act.”

Table 3 shows a failure case of both Transformers. This example contains a deletion of the expression, “or the petitioner for objection.” The Copiable Translation Transformer one output meaningless words at the beginning of the sentence and the expression to be deleted. The naive Transformer one output lacked such unchanged expressions as “must specify” and “notify the requestor.”

6 Discussion

We further examine our proposed architecture in this section.

6.1 Using Pre-amendment Sentences in Naive Transformer

The focal differential translation corpus [15] contains quartets (S_{PrO}, S_{PrT}, S_{PoO}, S_{PoT}). That is, we can arrange two translation pairs, (S_{PrO}, S_{PrT}) and (S_{PoO}, S_{PoT}), from the corpus, which doubles the training dataset. To assess the effect of doubling, we trained the naive Transformer with the doubled dataset.

Table 4 shows the result. We trained each model and combined it with the Kozakai model as the hyperparameters specified in Section 5.1. The naive Transformer one with the doubled dataset slightly outperformed the conventional dataset by around 2 points of BLEU, RIBES, and ISDIT. However, the Copiable Translation Transformer one achieved a far better performance than the two other models.

Table 3. Failure case

Model	Output
(Sentence S_{PrO})	経済産業大臣は、前条の意見の聴取の期日及び場所を定め、審査請求人又は異議申立人に通知しなければならない。
(Sentence S_{PrT})	the minister of economy , trade and industry must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review or the petitioner for objection .
(Sentence S_{PoO})	経済産業大臣は、前条の意見の聴取の期日及び場所を定め、審査請求人に通知しなければならない。
(Sentence S_{PoT})	the minister of economy , trade and industry must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review .
Naive + Kozakai	the minister of economy , trade and industry shall , in accordance with the procedures provided for in the preceding item , and the day on which the label set forth in the preceding article .
Copiable + Kozakai	in of ofy , and notify the foreign national must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review or the petitioner for review .

Table 4. Effect of doubled dataset

Model	BLEU	RIBES	ISDIT
Naive + Kozakai	22.32	53.89	10.52
Naive + Kozakai w/ doubled dataset	24.02	56.10	12.55
Copiable + Kozakai	**54.12**	**87.09**	**48.26**

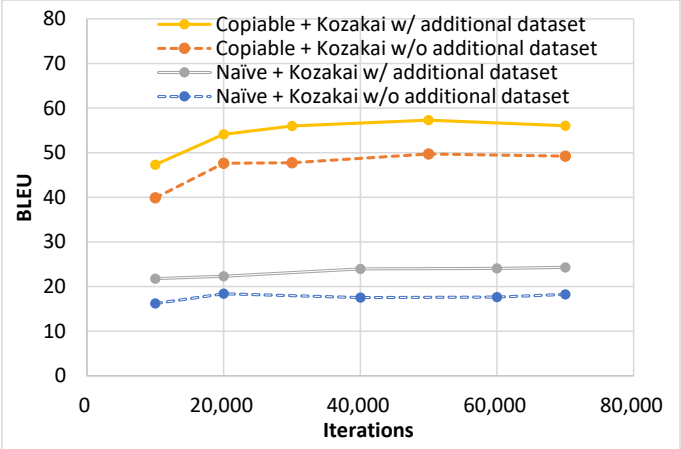


Fig. 5. BLEU scores in different iterations

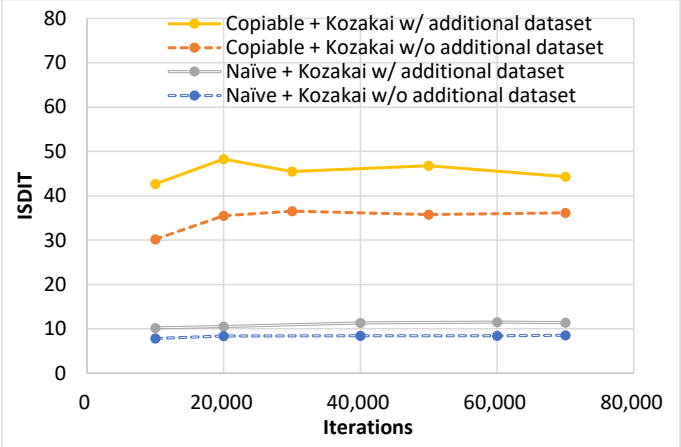


Fig. 6. ISDIT scores in different iterations

Table 5. Performance of naive Transformer with a large-scaled corpus

Model	BLEU	RIBES	ISDIT
Naive + Kozakai	22.32	53.89	10.52
Naive + Kozakai w/ large-scaled corpus	**83.00**	**94.16**	**74.14**
Copiable + Kozakai	54.12	87.09	48.26

6.2 Training Iterations

We investigated the transition of the performance on different training iterations to exclude the results from chance. Figures 5 and 6 show BLEU and ISDIT with different iterations. The tendency of the translation performance discussed above was kept in all the iterations. In all the models and criteria, the performance did not drastically improve as the iteration number grew.

6.3 Using a Large-scaled Bilingual Corpus

We can acquire bilingual corpora with hundreds of thousands of Japanese-English statutory sentence pairs [6, 14, 15]. In this section, we trained a naive Transformer with a large-scaled corpus containing 391,758 sentence pairs [15]. We set the iteration number to 2,000,000 for training the model and identical values as Section 5.1 for the other hyperparameters. We combined the naive Transformer model with the Kozakai model, as explained in Section 5.1.

Table 5 shows the experimental result. The large-scaled corpus drastically raised the naive Transformer one's performance, which supersedes the best performance from our Copiable Translation Transformer. However, researching our method remains valuable since it achieved an acceptable performance with less than 1% of the training data of the large-scaled corpus.

Table 6. Output of naive Transformer trained with a large-scaled corpus

Model	Output
(Sentence S_{PrO})	経済産業大臣は、前条の意見の聴取の期日及び場所を定め、審査請求人又は異議申立人に通知しなければならない。
(Sentence S_{PrT})	the minister of economy , trade and industry must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review or the petitioner for objection .
(Sentence S_{PoO})	経済産業大臣は、前条の意見の聴取の期日及び場所を定め、審査請求人に通知しなければならない。
(Sentence S_{PoT})	the minister of economy , trade and industry must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review .
Naive + Kozakai	the minister of economy , trade and industry shall , in accordance with the procedures provided for in the preceding item , and the day on which the label set forth in the preceding article .
Naive + Kozakai w/ large-scaled corpus	the minister of economy , trade and industry must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the applicant of the request for review .
Copiable + Kozakai	in of ofy , and notify the foreign national must specify the date and place of the hearing of opinions prescribed in the preceding article , and notify the requestor for review or the petitioner for review .

Table 6 shows the system outputs of the example in Table 3. The naive Transformer one with a large-scaled corpus output a quite accurate translation. However, it is not completely focal because of an unnecessary modification from "requestor" (審査請求人; *shinsaseikyunin*) to "applicant of the request." On the other hand, the Copiable Translation Transformer one retained that word.

7 Summary

We proposed architecture for Transformer that aims for differential translations for amended statutory sentences. Our architecture, Copiable Translation Transformer, has a copying mechanism tuned for translation tasks. In the prediction of the next word, it can either generate a word from the vocabulary or copy one from a pre-amendment statutory sentence. The copying mechanism raises the focality of the system output compared to the naive Transformer. With a dataset of a few thousand examples, our Copiable Translation Transformer outperformed the naive Transformer in not only IS-DIT but also BLEU and RIBES. However, it failed to reach the performance of the naive Transformer trained by a large-scaled bilingual corpus with hundreds of thousands of examples.

Our future work will address the following two tasks. First, we will search for a method to artificially augment the amount of training data for the Copiable Translation Transformer because manually translating post-amendment statutory sentences is very expensive. Second, we will search for a method to utilize the knowledge of a large-scaled corpus in training a Copiable Translation Transformer model.

Acknowledgments This work was partly supported by JSPS KAKENHI Grant Number 21H03772.

References

1. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1631–1640 (2016)
2. Hirao, T., Isozaki, H., Sudoh, K., Duh, K., Tsukada, H., Nagata, M.: Evaluating translation quality with word order correlations. Journal of Natural Language Processing. vol. 21. no. 3, 421–444 (2014) (In Japanese)
3. Hoseishitsumu-Kenkyukai: Workbook Hoseishitsumu (newly revised second edition). Gyosei (2018) (In Japanese)
4. 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)
5. 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)
6. 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)
7. 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)
8. Ogawa, Y., Inagaki, S., Toyama, K.: Automatic consolidation of Japanese statutes based on formalization of amendment sentences. New Frontiers in Artificial Intelligence. JSAI 2007. Lecture Notes in Computer Science. vol. 4914, 363–376 (2008)
9. 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)
10. 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)
11. 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)
12. 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)
13. Xu, S., Li, H., Yuan, P., Wu, Y., He, X., Zhou, B.: Self-attention guided copy mechanism for abstractive summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1355–1362 (2020)
14. Yamakoshi, T., Komamizu, T., Ogawa, Y., Toyama, K.: Differential translation for Japanese partially amended statutory sentences. New Frontiers in Artificial Intelligence. JSAI-isAI 2020. Lecture Notes in Computer Science. vol. 12758, 162–178 (2021)
15. Yamakoshi, T., Komamizu, T., Ogawa, Y., Toyama, K.: Evaluation scheme of focal translation for Japanese partially amended statutes. In: Proceedings of the 8th Workshop on Asian Translation (WAT2021). pp. 124–132 (2021)

Text Classification of Modern Mongolian Legal Documents

Garmaabazar Khaltarkhuu[1][0000-0003-2303-5815], Biligsaikhan Batjargal[2][0000-0002-3068-2634] and Akira Maeda[3][0000-0002-5494-132X]

[1] Graduate School of Information Science and Engineering, Ritsumeikan University.
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan
garmaabazar@gmail.com
[2] Research Organization of Science and Technology, Ritsumeikan University.
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan
biligsaikhan@gmail.com
[3] College of Information Science and Engineering, Ritsumeikan University.
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan
amaeda@is.ritsumei.ac.jp

Abstract. This paper investigates the application of state-of-the-art deep-learning-based natural language processing techniques in modern Mongolian legal documents. In particular, we explore several approaches that apply Bidirectional Encoder Representations from Transformers (BERT) models for classifying modern Mongolian legal documents. Based on our findings, we propose BERT-based models called LEGAL-BERT-Mongolian. We demonstrated two variants of LEGAL-BERT-Mongolian, i.e., uncased-LEGAL-BERT-Mongolian and cased-LEGAL-BERT-Mongolian, for classifying modern Mongolian legal documents. The uncased-LEGAL-BERT-Mongolian model achieved the best results, with a precision of 0.91, recall of 0.87, and F1 score of 0.89, whereas the cased-LEGAL-BERT-Mongolian model achieved a precision of 0.87, recall of 0.83, and F1 score of 0.85. To investigate how the capital letters in Mongolian legal text harm the classification, further detailed analyses are necessary. Moreover, because the LEGAL-BERT-Mongolian models demonstrated a certain degree of confusion among the "legal," "economy," and "politics" categories, some distinct deep features need to be explored to properly distinguish these classification categories. Furthermore, the proposed LEGAL-BERT-Mongolian models need to be evaluated on larger datasets.

Keywords: Document classification, Deep learning, Mongolian legal documents, Legal document analysis.

1 Introduction

In organizational management, making decisions at the executive level by analyzing various contracts or legal documents is an important task. There have been increasing demands from researchers, lawyers, executives, and managers to analyze legal documents on a massive scale with prompt and accurate results. Furthermore, a

computerized system that makes expert decisions is a long-awaited device for executives and managers who want to avoid problems, conflicts, disputes, and other types of issues. We have an ambitious goal to develop a decision support system (DSS) for helping executives and managers resolve complicated problems and/or questions in a Mongolian context. There has been virtually no deployment of such a system in Mongolia. Specifically, to the best of our knowledge, an approach that applies deep learning techniques to modern Mongolian legal documents has yet to be developed.

1.1 Problems and Current Situation in Mongolia

In his speech at the intermediate evaluation's discussion panel of the "New development" mid-term action plan, the Chief Cabinet Secretary of the Government of Mongolia stated that "Over the past 20 years in Mongolia, the government, ministries, and their agencies have issued 517 short or long-term development plans and strategy papers. Currently, 203 of these documents are effective, and many of them overlap or contradict with each other significantly. Only 132 of them are enforced, which has led to less than 26% efficiency" [1]. In addition, according to Worldwide Governance Indicators of the World Bank, the quality of governance in Mongolia as of 2020 is relatively low at 39.9% in effectiveness, 49.5% in regulatory quality, and 45.7% in rule of law [2].

In general, all decisions in Mongolia are highly dependent on the individual decision maker, and many opportunities are lost forever owing to poor and incorrect decisions. It is therefore particularly important to make a professional and prompt decision backed up with decent research using artificial intelligence (AI) systems.

1.2 Mongolian Language

Mongolian is spoken by the populations in Mongolia and Inner Mongolian and by other ethnic Mongols who live in several provinces of the People's Republic of China and the Russian Federation [3]. Mongolians have used numerous writing systems, and the Mongolian language has been written phonetically with Chinese characters, a traditional Mongolian script, Square or Phags-pa script, Todo or Clear script, Soyombo script, Horizontal square script, Latin, and Cyrillic script [4].

Based on the language reform in 1946, the Cyrillic script was adapted to Mongolian with two additional characters, and Cyrillic became the official Mongolian language script. At that time, the spelling of modern Mongolian in the Cyrillic alphabet was based on the pronunciations in the dialect of the Khalkha, the largest Mongol ethnic group [5] [6]. This was a radical change because traditional Mongolian script preserves the old Mongolian language, whereas modern Mongolian in Cyrillic script reflects the unique pronunciations in modern dialects. Although the sounds of words changed as the Mongolian language evolved, the spelling remained unchanged in traditional Mongolian script. Therefore, there are occasional differences between writing in traditional Mongolian script and written documents in modern Mongolian in Cyrillic script. Furthermore, Mongolian language is an agglutinative language. The presence of suffixes is different between modern Mongolian in Cyrillic script and traditional

Mongolian script. In traditional Mongolian script, most of the inflectional suffixes excluding a few plural suffixes are written separately from the stem of a word or from other suffixes. However, in modern Mongolian in Cyrillic script, inflectional suffixes such as plural suffixes, case suffixes, reflexive suffixes, voice suffixes, tense suffixes, aspect suffixes, and mood suffixes are concatenated with the stem. Thus, stemming of modern Mongolian in Cyrillic script is more difficult than traditional Mongolian script. Moreover, modern Mongolian in Cyrillic script is case-sensitive, whereas traditional Mongolian is not.

Mongolian legal documents are mainly written in modern Cyrillic script, although the traditional Mongolian script is used in both Mongolia and the Inner Mongolia Autonomous Region of China. Legal documents from the Inner Mongolia Autonomous Region are written in Chinese along with the traditional Mongolian script. This research considers Mongolian legal documents written in modern Cyrillic script and not Chinese legal documents written in traditional Mongolian script.

1.3 Scope of the Present Paper

To achieve our goal of developing a DSS, we investigate how to apply deep-learning-based state-of-the-art natural language processing (NLP) techniques to modern Mongolian legal documents. The current situation and challenges in Mongolia have led us to conduct comprehensive research in developing a new deep learning model for analyzing modern Mongolian legal documents.

In this paper, we report our progress in classifying modern Mongolian legal documents. We believe that text classification is one of the most important tasks in the DSS that we aim to develop.

This paper is organized as follows: In Section 2, some related studies are introduced. Text classification tasks of modern Mongolian legal documents are then introduced in Section 3. The experiments and their results are detailed in Section 4. Finally, some concluding remarks are given in Section 5.

2 Related Work

A new form of AI, i.e., deep learning, has shown breakthrough results in various fields [7]. It has also achieved a high level of performance across many different NLP tasks in English [8]. State-of-the-art NLP systems for English produce a near-human performance [9].

Within the legal domain, the LawGeex's contract review platform outperforms the 85% accuracy achieved by experienced lawyers on average, with a 94% accuracy rate in reviewing the samples of unseen non-disclosure agreements (NDAs) in English [10]. Chalkidis et al. proposed LEGAL-BERT, which is achieved by fine-tuning the English version of BERT. LEGAL-BERT has been trained on EU, UK, and US legal datasets written in English [11]. Moreover, Chalkidis et al. introduced a collection of datasets, called the Legal General Language Understanding Evaluation (LexGLUE) benchmark, for evaluating the model performance across a diverse set of legal natural language

understanding tasks in a standardized manner [12]. Shaghaghian et al. analyzed how different transformer-based language models trained on general-domain corpora can be customized for multiple legal documents reviewing various tasks in English [13].

Estonia set an ambitious goal to deploy AI to help clear its legal backlog. An AI-powered judge will analyze Estonian legal documents and propose a decision based on the analysis. Finally, a human judge will revise the decision of the AI and make the final determination [14]. Caled et al. proposed a hierarchical deep learning model for classifying Portuguese legal documents. They also built a legal corpus of 220K documents written in Portuguese [15]. Using the combination of BERT and Bidirectional Long Short-Term Memory (BiLSTM) models, Wang et al. proposed a classification method for legal questions in China that are written in Mongolian using the traditional Mongolian script [16]. However, their corpus and language model are for the traditional Mongolian script and not for modern Mongolian written in Cyrillic script.

Nevertheless, there has been little research conducted on modern Mongolian NLP, and to the best of our knowledge, no studies have considered AI analyses on modern Mongolian legal documents owing to a lack of research in such areas. We believe that conducting deep-learning-based analyses of modern Mongolian legal documents is an excellent opportunity to demonstrate AI-driven NLP techniques in the legal domain and develop an advanced state-of-the-art system.

As part of its legal system, Mongolia started following civil law in 1990 immediately after abandoning a socialist legal system that had been implemented during the previous 68 years. Most English-speaking countries such as the United States, England, Australia, and Canada have a common law system. The jurisdiction process varies depending on the legal system. We should therefore conduct a more detailed analysis for processing modern Mongolian legal documents rather than adopting an as-is language model in English, such as LEGAL-BERT.

3 Text Classification of Modern Mongolian Legal Documents

Because BERT has shown a breakthrough performance in various NLP tasks in English, Erdene-Ochir et al. publicized a Mongolian BERT model pre-trained on 500M Mongolian words extracted from Mongolian news articles with 700M words and a Wikipedia dump as of December 20, 2018 [17]. Among the cased-BERT-Mongolian-base, cased-BERT-Mongolian-large, uncased-BERT-Mongolian-base, and uncased-BERT-Mongolian-large models, we mainly considered cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large, which have 16 attention heads, 1024 hidden dimensions, and 24 hidden layers with a learning rate of 1e-4, as the original BERT implementation. Mongolian BERT models use Google's SentencePiece [18] with a vocabulary size of 32,000 and a maximum sequence of the first 512 tokens.

3.1 Existing Text Classification Tasks for Mongolian Language

Gunchinish [19] demonstrated the use of text classification tasks in labeling modern Mongolian news text by fine-tuning pre-trained Mongolian BERT models such as cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large. These tasks were fine-tuned by training 75,661 Mongolian news articles in 9 categories (labels): 1) art and culture, 2) economy, 3) health, 4) legal, 5) politics, 6) sports, 7) technology, 8) education, and 9) nature and environment. The distribution of Mongolian news articles can be seen in Fig. 1. The most frequently appearing news articles in this dataset are in the areas of sports, politics, and economy related news, which account for 17.8%, 17.7%, and 16.4% of all articles, respectively. Among these 75,661 Mongolian news articles, there are 8,285 news articles in the "legal" category, which make up 10.9% of the entire dataset.

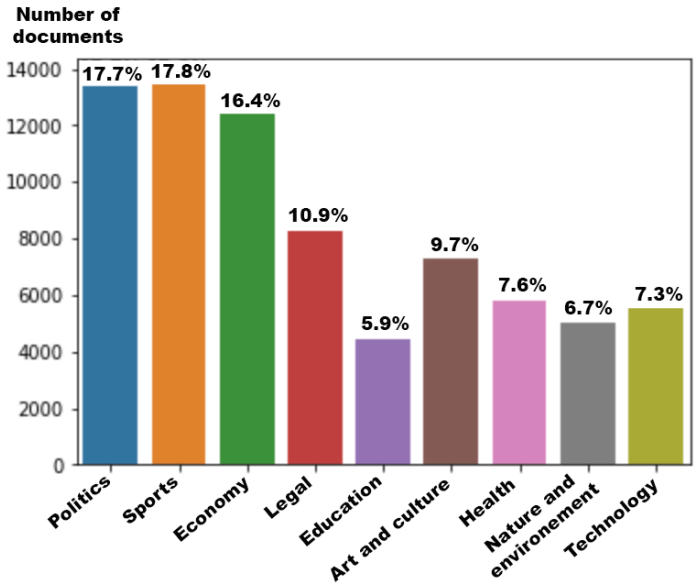


Fig. 1. Numbers of Mongolian news articles in each category.

The performances of these text classification tasks utilizing cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large are shown in Table 1 and Table 2, respectively, based on the precision, recall, and F1 score. The confusion matrices of cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large-based text classification tasks are shown in Fig. 2 and Fig. 3, respectively. In these experiments, we split all 75,661 news articles with a training-test split ratio of 80:20 through a random shuffling.

Table 1. Performances in classifying modern Mongolian news articles using the cased-BERT-Mongolian-large.

Category	Precision	Recall	F1 score
Art and culture	0.81	0.70	0.75
Economy	0.57	**0.84**	0.68
Education	0.74	0.44	0.55
Health	0.70	0.76	0.73
Legal	0.67	0.78	0.72
Nature and environment	0.82	0.42	0.55
Politics	0.78	0.78	0.78
Sports	**0.91**	0.78	**0.84**
Technology	0.76	0.64	0.70

Table 2. Performances in classifying modern Mongolian news articles using the uncased-BERT-Mongolian-large.

Category	Precision	Recall	F1 score
Art and culture	0.89	0.90	0.89
Economy	0.79	0.78	0.79
Education	0.72	0.72	0.72
Health	0.86	0.82	0.82
Legal	0.82	0.82	0.82
Nature and environment	0.71	0.72	0.71
Politics	0.84	0.87	0.85
Sports	**0.95**	**0.95**	**0.95**
Technology	0.87	0.83	0.85

As shown in the above results, the uncased-BERT-Mongolian-large model is capable of labeling sports news more accurately with an F1 score of 0.95, whereas the performance for nature and environment related news was the lowest with an F1 score of 0.71. The F1 score in labeling the "legal" news category was 0.82. The "legal" category of these 75,661 news articles contains not only Mongolian news articles about legal matters but also some parts of official legal documents. In general, uncased-BERT-Mongolian-large achieves a better performance than cased-BERT-Mongolian-large in classifying modern Mongolian news articles. The performance of the BERT-Mongolian-large models in classifying the official Mongolian legal documents is described in the next section.

As shown in Fig. 2 and Fig. 3, in the confusion matrices of the text classification tasks based on cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large, the most confusing categories were "economy" versus "politics," "politics" versus "economy," and "nature and environment" versus "economy." The numbers except the

diagonal values in the confusion matrices indicate the numbers of incorrect predictions for each category. The higher numbers correlate to the most confusing categories.

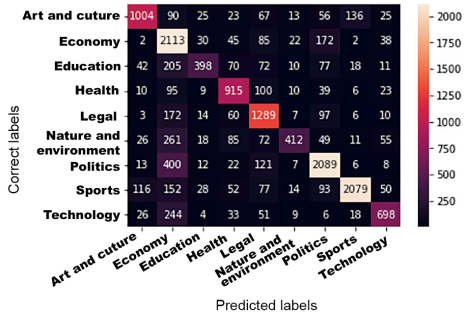


Fig. 2. Confusion matrix of the cased-BERT-Mongolian-large-based text classification task.

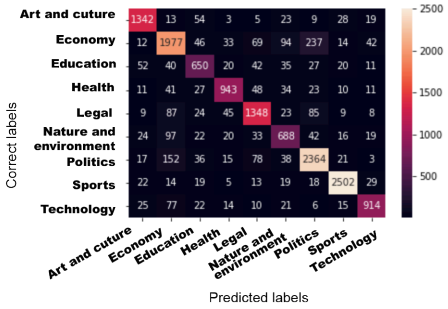


Fig. 3. Confusion matrix of the uncased-BERT-Mongolian-large-based text classification task.

Experimental Results in Classifying Modern Mongolian Legal Documents using Existing Text Classification Models. Further, we conducted experiments to classify modern Mongolian legal documents using the text classification models introduced in Section 3.1. During these experiments, we used the existing fine-tuned text classification models with training batch sizes of 16 and 5 epochs. The modern Mongolian legal documents used are described in Section 3.2. All 11,323 Mongolian legal documents of mostly unseen data were presented to the fine-tuned Mongolian BERT models, which were trained on Mongolian news articles. The results of classifying these modern Mongolian legal documents using cased-BERT-Mongolian-large and uncased-BERT-Mongolian-large are shown in Fig. 4 and Fig. 5, respectively.

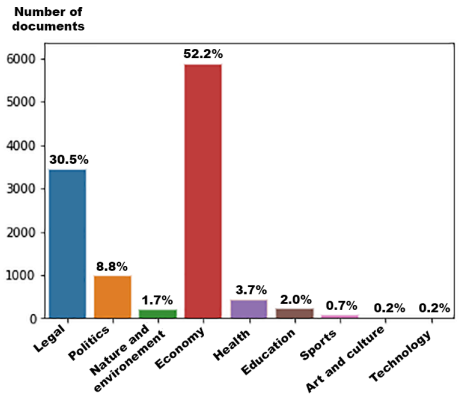


Fig. 4. Classification results of cased-BERT-Mongolian-large-based text classification model over modern Mongolian legal documents.

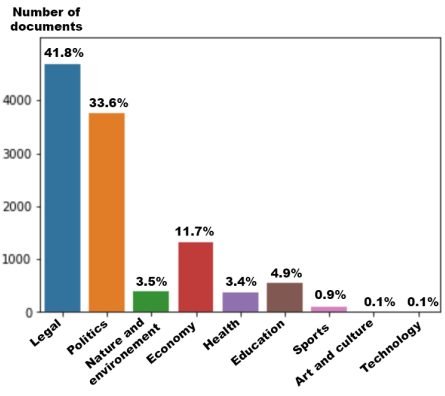


Fig. 5. Classification results of uncased-BERT-Mongolian-large-based text classification model over modern Mongolian legal documents.

When the fine-tuned cased-BERT-Mongolian-large model was utilized for classifying modern Mongolian legal documents, 30.5% of the dataset was correctly labeled as "legal." However, 52.2% were labeled as "economy" and 8.8% were labeled as "politics." By contrast, when the fine-tuned uncased-BERT-Mongolian-large model was utilized for classifying modern Mongolian legal documents, 41.8% of the dataset was correctly labeled as "legal." Still, 33.6% were labeled as "politics" and 11.7% were labeled as "economy." In general, uncased-BERT-Mongolian-large shows a better performance than cased-BERT-Mongolian-large in classifying modern Mongolian legal documents. However, existing text classification models become confused among the categories of "legal," "economy," and "politics." Therefore, further investigations into a decent language model for classifying Mongolian legal documents are necessary.

3.2 Modern Mongolian legal documents

Many modern Mongolian legal documents have recently been made publicly available in digital format. However, analyses of these legal documents have not been conducted, mainly owing to a lack of NLP tools that can handle modern Mongolian legal documents. Further computerized analyses are therefore necessary. In this study, legal documents including Mongolian laws and decrees of government organizations were prepared for analyzing modern Mongolian legal documents. The legal documents listed in Table 3 were freely downloaded from the public domain website "Unified Legal Information System" [20] of the National Legal Institute of Mongolia. There are 11,323 modern Mongolian legal documents in 13 legal categories.

Table 3. Utilized modern Mongolian legal documents.

Legal category	Number of documents
Resolutions of the government of Mongolia	4,716
Resolutions of the State Ikh Khural (Parliament of Mongolia)	2,066
Resolutions of self-governing bodies (the Citizens' Representative Hurals) of the provinces and capital city Ulaanbaatar	996
Laws of Mongolia	778
Ministerial decrees	748
International treaties concluded or ratified by Mongolia	654
Regulations of government agencies	324
Decisions of the Constitutional Court of Mongolia	295
Decrees of the president of Mongolia	211
Resolutions of the State Supreme Court	174
Decisions of various councils, committees and other collectives	169
Decisions of the heads of the bodies appointed by the Parliament	114
Orders of the governors of the provinces and the mayor of the capital city Ulaanbaatar	78

3.3 LEGAL-BERT-Mongolian: Text Classification of Modern Mongolian Legal Documents

We developed BERT-based models, which we call LEGAL-BERT-Mongolian, for classifying modern Mongolian legal documents. The legal domain is a specialized domain, and a legal text has distinct vocabularies and characteristics. Thus, as described in the Section 3.1, the performance of the existing text classification models in modern Mongolian legal text is unsatisfactory. With LEGAL-BERT-Mongolian, we adapted BERT-Mongolian-large with additional training on modern Mongolian legal documents, and pre-trained BERT-Mongolian-large on only such documents. We adapted both cased and uncased versions of BERT-Mongolian-large with additional training on modern Mongolian legal documents, as introduced in Section 3.2, along with 75,661 Mongolian news articles. The experimental setup and results of classifying modern Mongolian text, including modern Mongolian legal documents, using LEGAL-BERT-Mongolian are described in Section 4.

4 Experiments

The performance in classifying modern Mongolian legal documents using the cased and uncased versions of LEGAL-BERT-Mongolian were evaluated by calculating the precision, recall, and F1 score.

4.1 Experimental Setup and Datasets

Experimental Setup. Our experimental setup has 16 attention heads, 1,024 hidden dimensions, and 24 hidden layers with a learning rate of 1e-4 and a layer dropout of 0.5. We utilized Google's SentencePiece with a vocabulary size of 32,000 and AutoTokenizer using the maximum sequence of the first 512 tokens, similar to the original BERT implementation. All training was applied with a training batch size of 16 and 5 epochs. There are 337,449,481 parameters in total, 1,843,721 of which are trainable.

Datasets. We prepared a dataset by combining 11,323 modern Mongolian legal documents, as introduced in Section 3.2, along with 75,661 Mongolian news articles. The data distribution for each category can be seen in Fig. 6. In this dataset, the "legal" category data containing 1) Mongolian news articles on legal matters and 2) official Mongolian legal documents were increased to 22.5% of the entire dataset. Sports, politics, and economy related news make up 15.4%, 15.4%, and 14.2% of the dataset, respectively. The classification labels are the same as the categories of the news dataset, i.e., 1) art and culture, 2) economy, 3) health, 4) legal, 5) politics, 6) sports, 7) technology, 8) education, and 9) nature and environment. We split all 86,984 documents with a training-testing split ratio of 80:20 through random shuffling.

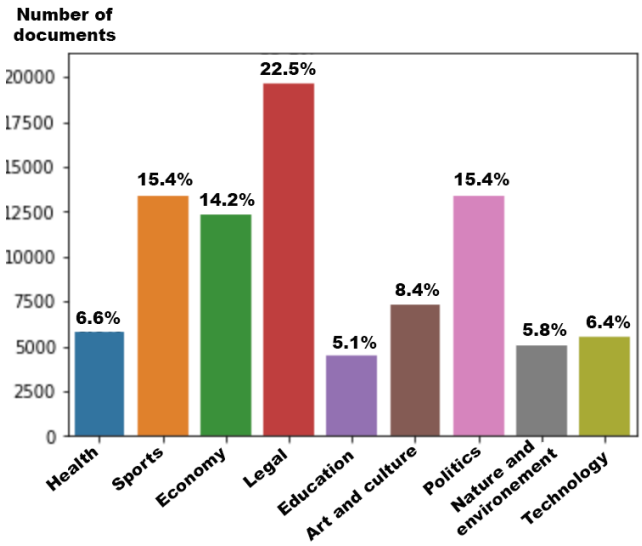


Fig. 6. Data distribution of modern Mongolian documents for each category.

4.2 Experimental Results

The experimental results of classifying modern Mongolian legal documents using cased-LEGAL-BERT-Mongolian and uncased-LEGAL-BERT-Mongolian are shown in Table 4 and Table 5, respectively, based on the precision, recall, and F1 score. The confusion matrices of cased-LEGAL-BERT-Mongolian and uncased-LEGAL-BERT-Mongolian are shown in Fig. 7 and Fig. 8, respectively.

Table 4. Performances in classifying modern Mongolian documents using the cased-LEGAL-BERT-Mongolian.

Category	Precision	Recall	F1 score
Art and culture	0.68	0.83	0.75
Economy	0.57	0.74	0.66
Education	0.53	0.48	0.51
Health	**0.89**	0.40	0.55
Legal	0.87	0.83	**0.85**
Nature and environment	**0.89**	0.34	0.49
Politics	0.72	0.83	0.77
Sports	0.77	**0.90**	0.83
Technology	0.76	0.57	0.65

Table 5. Performances in classifying modern Mongolian documents using the uncased-LEGAL-BERT-Mongolian.

Category	Precision	Recall	F1 score
Art and culture	0.88	0.91	0.89
Economy	0.71	0.82	0.76
Education	0.76	0.63	0.69
Health	0.84	0.81	0.82
Legal	0.91	0.87	0.89
Nature and environment	0.79	0.64	0.71
Politics	0.84	0.83	0.84
Sports	**0.95**	**0.95**	**0.95**
Technology	0.78	0.86	0.81

Although the LEGAL-BERT-Mongolian models were trained mainly on modern Mongolian legal documents, the uncased-LEGAL-BERT-Mongolian model showed the best precision, recall, and F1 score of 0.95 in labeling the "sports" category documents. As shown in Table 3, the performance of cased-LEGAL-BERT-Mongolian model varied. The highest F1 score, 0.85, was achieved for labeling the "legal" category documents. The highest precision was found for labeling the "health" and "nature and environment" category documents at 0.89. Finally, the highest recall result of 0.90 was achieved for labeling the "sports" category documents. Although there are many case-sensitive proper nouns and legal terms in Mongolian legal text, uncased-LEGAL-BERT-Mongolian achieved a better performance in classifying modern Mongolian legal documents. Moreover, as shown in the confusion matrices of both cased and uncased LEGAL-BERT-Mongolian models shown in Fig. 7 and Fig. 8, respectively, a certain degree of confusion remains in the "legal" category in comparison to the "economy" and "politics" categories.

Because our target is the Mongolian legal domain, we needed to compare the performance of LEGAL-BERT-Mongolian against the existing BERT-Mongolian-large models. In Table 6, we compare the precision, recall, and F1 score of cased-BERT-Mongolian-large, uncased-BERT-Mongolian-large, cased-LEGAL-BERT-Mongolian, and uncased-LEGAL-BERT-Mongolian in classifying modern Mongolian legal documents. The uncased-LEGAL-BERT-Mongolian model shows the best results, with a precision of 0.91, recall of 0.87, and F1 score of 0.89.

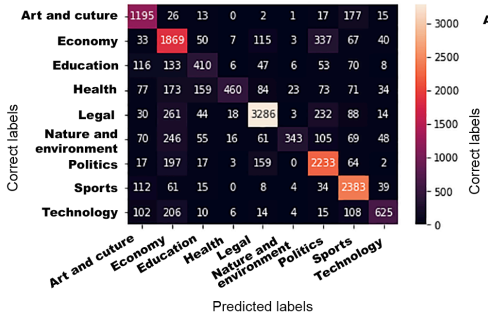


Fig. 7. Confusion matrix of the text classification task of the cased-LEGAL-BERT-Mongolian.

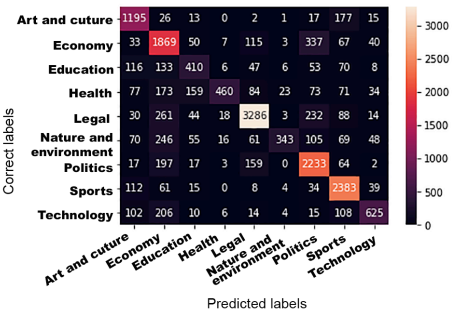


Fig. 8. Confusion matrix of the text classification task of the uncased-LEGAL-BERT-Mongolian.

Table 6. Performance comparison in classifying modern legal Mongolian documents using various models.

Model	Precision	Recall	F1 score
Cased-BERT-Mongolian-large	0.67	0.78	0.72
Uncased-BERT-Mongolian-large	0.82	0.82	0.82
Cased-LEGAL-BERT-Mongolian	0.87	0.83	0.85
Uncased-LEGAL-BERT-Mongolian	**0.91**	**0.87**	**0.89**

5 Summary

In this paper, we investigated how to apply existing deep learning models for classifying modern Mongolian legal documents. We proposed a group of BERT-based models called LEGAL-BERT-Mongolian, which we believe are an important aspect of the DSS that we aim to develop at a future date. The uncased-LEGAL-BERT-Mongolian model achieved the best results with a precision of 0.91, recall of 0.87, and F1 score of 0.89, whereas cased-LEGAL-BERT-Mongolian achieved a precision of 0.87, recall of 0.83, and F1 score of 0.85 in classifying modern Mongolian legal documents. Uncased BERT models showed a better performance than the cased-BERT models in classifying modern Mongolian legal documents, despite many case-sensitive proper nouns and legal terms in modern Mongolian legal text. For instance, the first letter of places, organizations, and personal names are always capitalized. Thus, the words written in capital letters have different meanings in a particular context. Legal writing typically requires strict formatting. Likewise, there are some strict rules in the formatting of modern Mongolian legal documents, such as document titles and part/chapter names and their cross references, which should be written in all capital letters.

As a future study, a further detailed error analysis is necessary to investigate how much the capitalized proper nouns and legal terms of Mongolian legal text negatively

affect the text classification. Moreover, because the LEGAL-BERT-Mongolian models show a certain degree of confusion between the "legal," "economy," and "politics" categories, some distinct features need to be applied in the proposed models to distinguish the classification categories. Many common or similar sentences may occur in different categories of modern Mongolian text. Another limitation that we need to consider is the token limit of 512 for BERT. Some important parts of Mongolian legal documents might be omitted because the AutoTokenizer utilized in the LEGAL-BERT-Mongolian models automatically truncated texts of longer 512 tokens. Among the various different modern Mongolian legal documents, Mongolian laws have 2,260 words on average, whereas longer laws have many more tokens. Moreover, we should compare the results and classification performance of various neural and non-neural classifiers against LEGAL-BERT-Mongolian. Finally, we also intend to test the proposed LEGAL-BERT-Mongolian model on a dataset of 288K Mongolian court decisions.

References

1. The State Great Hural of Mongolia: Mid-term evaluation's discussion of the medium-term action plan "New development," http://parliament.mn/n/gico, https://www.n24.mn/a/2010, last accessed 2022/03/21.
2. The World Bank Group.: Worldwide Governance Indicators, https://data-bank.worldbank.org/source/worldwide-governance-indicators, last accessed 2022/03/21.
3. Hirschberg, J., Manning, C. D.: Advances in natural language processing. Science 349(6245), 261–266 (2015).
4. Poppe, N. N.: Grammar of written Mongolian. University of Washington/Seattle, Washington: Wiesbaden: Otto Harrassowitz (1954).
5. Shagdarsuren, T.: Study of Mongolian Scripts (Graphic Study of Grammatology). National University of Mongolia/Ulan Bator, Ulan Bator: Urlakh Erdem Khevleliin Gazar (2001) (in Mongolian).
6. Sečenbaγatur, Q., Tuyaγ-a, B., Ying, U.: Mongγul kelen-ü nutuγ-un ayalγun-u sinǰilel-ün uduridqal. Hohhot: Öbür mongγul-un arad-un keblel-ün qoriy-a (2005) (in Mongolian).
7. Svantesson, J., Tsendina, A., Karlsson, A. & Franzén, V.: The Phonology of Mongolian. New York: Oxford University Press (2005).
8. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M., Chen, S., Iyengar, S. S.: A Survey on Deep Learning: Algorithms, Techniques, and Applications. ACM Computing Surveys (CSUR) 51(5), 1–46 (2018).
9. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine 13(3), 55–75 (2018).
10. LawGeex, Comparing the Performance of Artificial Intelligence to Human Lawyers in the Review of Standard Business Contracts, https://images.law.com/contrib/content/uploads/documents/397/5408/lawgeex.pdf, last accessed 2022/03/21.
11. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The Muppets straight out of Law School. Findings of ACL: EMNLP 2020, (2020).
12. Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D. M., Aletras, N.: LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. In: The 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), Dublin, Ireland (2022).

13. Shaghaghian, S., Feng, L., Jafarpour, B., Pogrebnyakov, N.: Customizing Contextualized Language Models for Legal Document Reviews. In: IEEE International Conference on Big Data (Big Data 2020), pp. 2139–2148. IEEE Xplore, Atlanta, GA, USA (2020).
14. Niiler, E.: Can AI Be a Fair Judge in Court? Estonia Thinks So. Wired, https://www.wired.com/story/can-ai-be-fair-judge-court-estonia-thinks-so/, last accessed 2022/03/21.
15. Caled, D., Won, M., Martins, B., Silva, M. J.: A Hierarchical Label Network for Multi-Label EuroVoc Classification of Legislative Contents. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (eds.) The 23rd International Conference on Theory and Practice of Digital Libraries (TPDL 2019), LNCS, vol. 11799, pp. 238–252, Springer, Heidelberg (2019).
16. Wang, G., Bao, F., Wang, W.: Mongolian Questions Classification in the Law Domain. In: International Conference on Asian Language Processing (IALP 2020), pp. 56–59. IEEE Xplore, Kuala Lumpur, Malaysia (2020).
17. Erdene-Ochir, T., Gunchinish, Sh., Bataa, E.: BERT Pretrained Models on Mongolian Datasets, https://github.com/tugstugi/mongolian-bert/, last accessed 2022/03/21.
18. Gunchinish, Sh.: Mongolian text classification, https://github.com/sharavsambuu/mongolian-text-classification, last accessed 2022/03/21.
19. Google.: SentencePiece, https://github.com/google/sentencepiece, last accessed 2022/03/21.
20. Unified Legal Information System, https://legalinfo.mn/en, last accessed 2022/03/21.

Mapping Similar Provisions between Japanese and Foreign Laws

Hiroki Cho[1], Aki Shima[2], and Makoto Nakamura[1]

[1] Faculty of Engineering, Niigata Institute of Technology, Japan
mnakamur@niit.ac.jp
[2] The Institute of Education and Student Affairs, Niigata University, Japan
shimaaki@me.com

Abstract. A system that automatically maps similar provisions between Japanese law and foreign law is useful in comparative legal research. The purpose of this study is to develop the system for automatically mapping similar provisions. This study uses the similarity of document vectors generated by the BERT model to find similar provisions. Through experiments, we found that (1) the BERT model is effective in mapping similar provisions, (2) BERT is also effective for English translations of Japanese legal texts, and (3) LEGAL-BERT, a BERT model specific to the legal domain, may not be effective for English translations of Japanese laws. In addition to this, we performed a mapping between Japanese and foreign laws in English and showed that the performance exceeded that of word-based mapping.

Keywords: comparative law · similar provisions · BERT

1 Introduction

When conducting business with or actually going abroad, it is necessary to have a good grasp of the laws of the other country. In particular, paying attention to similarities with the laws of one's own country makes it easier to understand what is common and what is different. Here, we consider comparisons with foreign laws from an academic perspective. Comparative law is the study of laws belonging to different legal systems that are compared with each other, with the aim of identifying differences between them. This study compares the laws of one's own country with those of a foreign country on a provision-by-provision basis, and considers the correspondence of similar provisions (Figure 1). It has been costly for experts to map similar provisions. Even for experts, it is difficult to find a correspondence between similar provisions in multiple laws. The reasons for this are as follows.

1. Different languages are used,
2. The sheer number of combinations to be compared makes it difficult to manually check them all, and
3. It is necessary to read not only the text but also its social context.

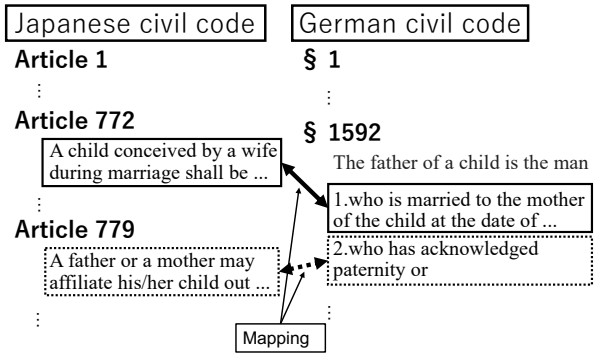


Fig. 1. Example of mapping between Japanese and foreign laws

These are the order in which non-specialists are most likely to try to solve the problem. Item 1 can be addressed by using government-provided English translation data or machine translation. In this case, we have confirmed that the accuracy of the correspondence depends on the accuracy of the translation [2]. The problem in item 2 is serious, and it is a slave task to check all the correspondences [5]. If this can be solved, the cost of correspondence can be reduced. Therefore, the purpose of this study is to develop a tool to automate the correspondence of similar provisions in comparative law research. We expect that automating items 1 and 2 will make it easier to address the issue of item 3.

In order to map similar provisions, it is first necessary to resolve the correspondence on a law-by-law basis. However, for large laws such as the Civil Code (Civil Code), for example, the mapping of similar provisions can be viewed as a similar document search with the provisions as a single document. One method of similar document retrieval is a ranking using the Jaccard coefficient [5]. This is a typical method that focuses on the number of word matches.

While methods that use word match counts are simple and fast, they do not take into account the meaning or context of the words. Therefore, two documents that use different words are judged to be completely different, even if they have similar content. To solve this problem, we proposed a method that uses word vectors obtained from language models. This study uses the latest language model, BERT [3].

In this study, experiments were conducted with the goal of mapping Japanese law to foreign law. First, we performed mapping between Japanese laws and evaluated BERT's performance in mapping similar provisions. Next, English translations of Japanese laws were mapped to verify the performance of the law-specific model. Finally, we mapped between Japanese and German laws and evaluated the results.

We explain similar document retrieval in Section 2. Based on this, the proposed method is described in Section 3. Section 4 shows experimental settings

(Purpose)
Article 1 The purpose of this Act is to establish fair control over foreign ...
(Definition)
Article 2 (1) The term "foreign national" as used in this Act means a person ...
(2) A person who has two or more nationalities other than Japanese ...
(Initial Registration)
Article 3 (1) All foreign nationals in Japan apply for registration with the ...
(i) one application form for alien registration;
(ii) passport;
(iii) two phtographs.
(2) In the case of the application under the preceding paragraph, a person ...
(3) If the head of a municipality finds unavoidable circumstances exist in the ...
(4) Where a foreign national has filed the application provided for in ...

Fig. 2. Basic Structure of Japanese Law (Alien Registration Act (Act No. 125, 1952))

and how to make correct data, and The experimental results are shown in Section 5. Finally, we conclude in Section 6.

2 Similar document search

The purpose of this study, i.e., the mapping of similar provisions, can be interpreted as a similar document search by considering each article as a single document. There are two methods of similar document retrieval: one is to treat a document as a set of words and calculate the similarity, and the other is to calculate the similarity from a distributed representation of documents obtained using a neural network.

2.1 Document Unit

In order to map similar provisions, it is necessary to consider whether *article* or *paragraph* should be used as the basic unit. As shown in Figure 2, an article consists of a set of paragraphs. An article that does not contain a paragraph, such as Article 1, can be regarded as an article that actually consists of a single paragraph. Thus, it might be better to consider a paragraph as a document.

In this paper, however, based on the results of preliminary experiments, an article is used as the basic unit. The reasons are as follows:

- There are extremely short paragraphs, which reduce the mapping performance during the vectorization process described in Section 2.3.
- Laws between two countries do not necessarily deal with the same subject matter. Many of them do not map to each other in terms of provisions. The nature of the problem requires that as many relevant provisions as possible be shown, so using a paragraph as the basic unit is too detailed.

– Even if a paragraph is used as the basic unit, it does not solve the problem of documents frequently exceeding the 512-token limit in BERT. In fact, it is better than using an article as the unit of measure, but it does not solve the implementation problem, since there are many documents that exceed 512 tokens even in a single paragraph.

2.2 Similarity of Bag of Words

The Jaccard coefficient in Eqn (1) is a method of expressing the similarity between two sets.

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

In similar document retrieval, the Jaccard coefficient can be calculated by considering each document as a set of words. However, while the Jaccard coefficient is simple to compute, it has the disadvantage that it does not take into account the importance of words, and even synonyms are treated as completely different words.

2.3 Similarity of Vectors

Based on the distribution hypothesis [4] that the meaning of a word is formed by its surrounding words, a neural network (NN) can represent the meaning of a word as a vector by learning with a large amount of text data. As an example, it is known that a vector close to "queen" can be obtained by performing the vector operation "king-man+woman."

BERT is a language model proposed by Devlin et al. [3] that has shown high performance on many NLP tasks. Currently, several pre-trained models are publicly available and can perform highly accurate classification simply by performing fine tuning. The output of the final layer can be extracted to obtain a word vector for each token of the input document.

However, the basic BERT model is based on a corpus of general documents such as wikipedia and has been reported to perform poorly in certain domains such as medicine [6]. In the legal domain, the LEGAL-BERT model [1] has been published as an English model.

3 Proposed Method

In this study, we used the article as the smallest unit of provisions. The flow of mapping to foreign laws is shown in Figure 3.

1. Prepare English translations into a common language for each law
2. Input each article as a document into BERT
3. Make the average of the outputs a document vector
4. Calculate cosine similarity between document vectors of home and foreign laws

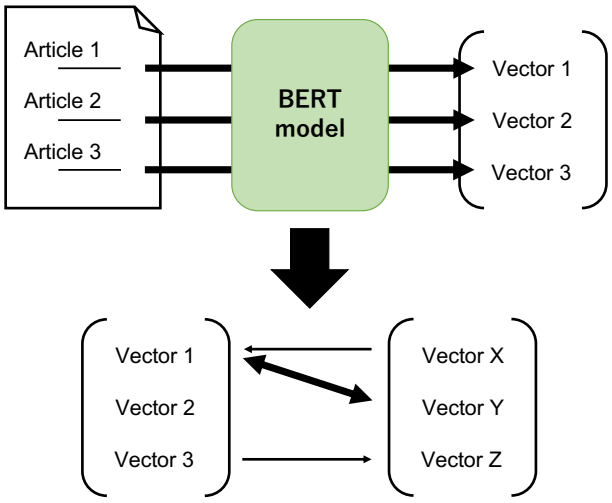


Fig. 3. Vector

5. Map articles based on cosine similarity

First, a translation into the unified language is needed. It is sufficient to select a language based on the availability of translations in the target country or on the accuracy of the machine translation. In this study, we used English translation laws provided by the governments. In item 5, the system calculates the similarity of each of the two sets of documents to all documents in the other set, and assigns a mapping to the document pair with the largest similarity in both directions, which is shown in the lower part in Figure 3.

4 Experiments

4.1 Purpose of the Experiments

In this section, we have mapped similar provisions between the Japanese Civil Code (Family Law) and the German Civil Code. The purpose of this experiment is to verify whether the proposed method is effective in mapping Japanese law to foreign law.

In this study, a three-step experiment is conducted as shown in Figure 4.

1. Mapping of articles between Japanese laws in Japanese
2. Mapping of articles between Japanese laws in English
3. Mapping of articles between Japanese and foreign laws in English

Since it is difficult to make correct data when mapping to a foreign law, this experiment tests the mapping of Japanese laws to each other. Here, we take the Electricity Business Act (Act No. 170 of 1964) and the Gas Business Act

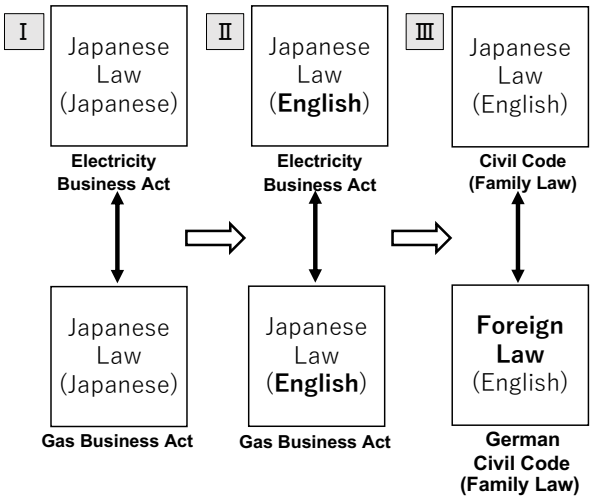


Fig. 4. Plan of Experiments

(Act No. 51 of 1954), which have similar legal contents. Although similarity of document vectors is used to map similar articles, vectorization by BERT is not always effective for legal documents as well as general documents. Therefore, we verified the performance of BERT in mapping similar articles through the first experiment by comparing it with the mapping by Jaccard coefficients.

Second, since it is necessary to use English translations of documents when mapping to a foreign law, we performed the mapping using English translations of Japanese laws and examined whether the mapping could be performed in the same way as in the Japanese case. Furthermore, since the LEGAL-BERT model, which is specific to the legal domain, is publicly available for the English BERT model, we compared its performance with the commonly used BERT-BASE.

Finally, following the above steps, a correspondence was made between the family law of the Japanese Civil Code and the German Civil Code. Although we have created the correct data using the method described below, not all correspondences are necessarily listed, so the evaluation cannot be guaranteed.

4.2 Procedure of Experiments

Documents from the experimental data were input into a pre-trained BERT model in Japanese, and the output, excluding [CLS] and [PAD] in the final layer, was averaged into a document vector. The pre-trained model was cl-tohoku/bert-base-japanese-whole-word-masking. the maximum number of input tokens for BERT was set to 512 and tokens beyond that were ignored. Cosine similarity between document vectors of different laws was calculated for all combinations. Every document refers to the document with the highest cosine similarity. If the referenced document has the highest similarity to the source document, the

Table 1. The number of articles (as of February 24, 2022)

	Language	#Articles
Electricity Business Act	Japanese	315
	English	304
Gas Business Act	Japanese	207
	English	221

two documents are mapped. The results of the mapping between the Electricity Business Act and the Gas Business Act were then evaluated using the correct data.

Next, in order to compare the results with those using the Jaccard coefficients, the same set of documents was divided into words by the morphological analyzer MeCab, and the Jaccard coefficients between two documents were used for similarity to map the documents. Neologd was used for the dictionary of MeCab.

Finally, we conducted a mapping experiment with BERT on the English translation of the laws, using bert-base-uncased and legal-bert-base-uncased as the English pre-trained models of BERT, and compared their performance.

For experiment 3, each law was entered into the BERT model (BERT-BASE) on an article-by-article basis and vectorized and mapped using the same method as in Experiment 2. At this time, we found that the one-to-one mapping using the method of Experiment 2 gave almost no correct answers. Therefore, we changed the method to map one article of Japanese law to the top five similar articles of German law. The results were evaluated based on the data of correct answers. For comparison, a method using the Jaccard coefficient was also evaluated using the same procedure.

4.3 How to Create a Correct Dataset: Japanese Laws

For the correspondence between Japanese laws, we used the Electricity Business Act (Act No. 170 of 1964) and the Gas Business Act (Act No. 51 of 1954). Both laws are similar in structure because they relate to infrastructure.

The laws used in these experiments are listed in Table 1. As of April 2022, there are 315 and 207 articles in the Electricity Business Act and the Gas Business Act, respectively. Note that the last article, Article 123 in the former and Article 207 in the latter, respectively, does not give the correct number because of deletions and additions by branch number in both laws.

Japanese law data and English translation data of Japanese laws were obtained from e-Gov[3] and Japanese Law Translation Database System[4], respectively, in xml format. The text under the article tag was extracted from each file and each was treated as one document.

[3] https://www.e-gov.go.jp
[4] http://www.japaneselawtranslation.go.jp

Table 2. Laws Compared in the Experiment

	Articles
Japanese Civil Code	725 - 886
German Civil Code	1297 - 1921

We showed the two laws to two Japanese workers and asked them to create a correspondence table. The workers were a lab student and a part-time university employee, neither of whom was familiar with the law.

Since the number of articles in the Electricity Business Act is larger than that in the Gas Business Act, we looked at the former law article-by-article, and if there was an article with similar content in the latter one, we mapped those two articles. Correspondence with articles in multiple locations was allowed.

The results showed that the two workers made the same correspondence for 113 out of 315 articles. There were 153 articles for which both workers considered that there was no correspondence. Thus, the agreement between their results was 0.84.

4.4 How to Create a Correct Dataset: Foreign Laws

In this paper, we focus on family law. Table 2 shows the laws and their scope used in the experiment. We used the English translations provided by the governments[5].

We used legal commentaries to create correct data. Thus far, three editions of the legal commentaries have been published by a leading Japanese publisher as follows.

1. Annotated Civil Code: The original series began publication in 1964. It does not correspond to the legal changes since then.
2. New edition of the Annotated Civil Code: Sequel series of the Annotated Civil Code published in 1988 to respond to changes in the law. However, more than 30 years have passed since then.
3. New Annotated Civil Code: The series began publication in 2016. The content of this book is different from the above two series because it was planned as an independent book.

We attempted to create correct data using the latest New Annotated Civil Code (3) [8], but only the first half had been published yet. Therefore, the latter part was extracted from the new edition of the Annotated Civil Code (2) [7, 9]. For this reason, there is a possibility that some of the entries do not correspond to the current law or that the policies for comparative laws are not consistent.

The New Annotated Civil Code states like in Figure 5 at the beginning of each article.

In this study, we took out this [Contrast] section and made a list by article.

[5] https://www.gesetze-im-internet.de/Teilliste_translations.html

(Sharing of Living Expenses)
Article 760 A husband and wife shall share the expenses that arise from the marriage taking into account their property, income, and all other circumstances.
[Contrast] German Civil Code 1360-1361, Swiss Civil Code 163-165, French Civil Code 214
[Amendment] 798

Fig. 5. Example of an entry in the New Annotated Civil Code [8]

As a result, 108 articles of the scope shown in Table 2 correspond to one or more articles of the German Civil Code.

It should be noted, however, that this list of comparative laws extracted from the legal commentaries is not entirely correct. For example, Article 766 (Determination of Matters regarding Custody of Child after Divorce etc.), which was amended in 2011, should be mentioned. Because of the detailed description of trends in foreign laws on approximately two pages, a list of comparative laws starting with '[Contrast]' is not provided. The worker mechanically created the correct data from the [Contrast] section, so processing this description was out of the scope. Therefore, the list of comparative laws extracted from the detailed description is not included in the correct data at this experiment.

4.5 Evaluation Method

The results of the mapping of the Electricity Business Act to the Gas Business Act were compared with the correct data.

The number of mappings included in both the results and the correct data was defined as TP, the number of mappings included in the results but not in the correct data as FP, the number of mappings not included in either the results or the correct data as TN, and the number of mappings not included in the results but in the correct data as FN.

In addition, since the correct data were created by two workers, semi-correct answers were introduced for the mappings in which the two workers did not agree. Accordingly, TP, TN, FP, and FN were defined as the number of cases in which the two workers agreed, SP as the number of cases in which the mapping was the same as that of one of the workers, and SN as the number of cases in which there was no such mapping as that of one of the workers. Accuracy, Recall, Precision, and F-measure were calculated as follows:

$$\text{Accuracy} = \frac{2TP + SP + 2TN + SN}{2TP + 2SP + 2FP + 2TN + 2SN + 2FN} \tag{2}$$

$$\text{Precision} = \frac{2TP + SP}{2TP + 2SP + 2FP} \tag{3}$$

$$\text{Recall} = \frac{2TP + SP}{2TP + SP + 2FN + SN} \tag{4}$$

Table 3. Result of the experiment 1

	Acc	Pre	Rec	F1
Jaccard	0.858	0.864	0.757	0.807
BERT	0.816	0.856	0.679	0.757

Table 4. Result of the experiment 2

	Acc	Pre	Rec	F1
BERT-BASE	0.803	0.814	0.679	0.740
LEGAL-BERT	0.792	0.786	0.679	0.729

$$\text{F-measure} = \frac{2\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

5 Experimental Results

5.1 Experiment 1: Japanese Laws in Japanese

Table 3 shows the results of the evaluation of the mapping between the Electricity Business Act and the Gas Business Act in Japanese.

Both methods yielded scores above 0.7 for all the evaluation metrics except the recall rate in the BERT model; when comparing the two methods, the Jaccard coefficient scored higher. This is because of the word-to-word similarity between the text of the Electricity Business Act and the Gas Business Act. However, a preliminary experiment using the Jaccard coefficients to map Japanese law to German law on a provision-by-provision basis did not yield good results. This is because the words used are different even if the contents are similar, and it is expected that the similarity of document vectors obtained using NN can be used to map articles with similar meanings.

This experimental results show that the performance of the two methods is comparable.

5.2 Experiment 2: Japanese Laws in English

Table 4 shows the results of the evaluation of the mapping between the Electricity Business Act and the Gas Business Act in the English translation.

Even with the English translated data, both BERT model-based mappings yielded scores similar to those in the Japanese case. This suggests that the English BERT model can be used to actually map Japanese law to foreign law after both are translated into English.

In addition, comparing BERT-BASE and LEGAL-BERT, Recall was higher in LEGAL-BERT and the rest in BERT-BASE. This may be because LEGAL-BERT is a model specialized for the legal domain, but only legal documents written in English such as EU law, UK law, and US contract documents were used for training, and Japanese law data were not used for training.

Table 5. Result of Experiment 3

	Correct answers (%)
Jaccard	21 / 108 = 19.4 (%)
BERT	32 / 108 = 29.6 (%)

Japan: Civil Code (Act No. 89 of 1896) **Article 792** A person who has attained the age of majority may adopt another as his/her child.
Germany: Civil Code **Section 1742** An adopted child may, as long as the adoption relationship exists, in the lifetime of an adoptive parent only be adopted by that parent's spouse.

Fig. 6. Example of mapping by BERT

5.3 Experiment 3: Japanese Civil Code and German Civil Code in English

The results of the mapping evaluation are shown in Table 5. The correct data consists of 108 mappings. In this experiment, we considered as correct if at least one correct answer was included in the articles up to the fifth place of prediction; the BERT-based method output more correct correspondences.

Figure 5 shows the correspondence that could not be found by the method using the Jaccard coefficients, but could be found by BERT.

6 Conclusion

In this study, we used BERT to map similar provisions between Japanese law and foreign law. In the experiment, we performed mapping between Japanese laws in Japanese and Japanese laws in English, as well as mapping between Japanese laws and German laws in English. The results showed that the BERT-based method succeeded in mapping similar provisions. The results also showed that BERT-BASE performed better than LEGAL-BERT in mapping Japanese law.

In this study, the proposed method enabled a computer to automatically map similar provisions. In this experiment, the mapping process can be completed in a few minutes using a commercial laptop computer, which is expected to significantly reduce the cost of the process compared to the manual process. If the accuracy of the correspondence can be further improved, it will be possible to construct a system that can be put into practical use.

In the future, the results of the mapping with German law need to be evaluated more accurately. First, the correct data will be reworked. Then we will modify the model and the mapping conditions to improve the accuracy of the mappings.

Acknowledgements This research was partly supported by JSPS KAKENHI Grant Number JP19H04427.

References

1. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261
2. Cho, H., Nakamura, M.: The relationship between translation accuracy and mapping of similar provisions to foreign laws in Comparetive Law Study (in Japanese). In: Proceedings of IEICE Shin-Etsu Section Conference. pp. 8B–2 (2021)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) NAACL-HLT (1). pp. 4171–4186. Association for Computational Linguistics (2019)
4. Harris, Z.: Distributional structure. Word **10**(2-3), 146–162 (1954). https://doi.org/10.1007/978-94-009-8467-7_1
5. Koyama, K., Sano, T., Takenaka, Y.: The legislative study on Meiji civil code by machine learning. In: Proceedings of the 15th International Workshop on jurisinformatics (JURISIN2021). pp. 41–53 (2021)
6. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C., Kang, J.: BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (Feb 2020). https://doi.org/10.1093/bioinformatics/btz682, funding Information: This research was supported by the National Research Foundation of Korea(NRF) funded by the Korea government (NRF-2017R1A2A1A17069645, NRF-2017M3C4A7065887, NRF-2014M3C9A3063541). Publisher Copyright: © 2020 Oxford University Press. All rights reserved.
7. Nakagawa, Z., Yamahata, M.: New Annotated Edition-The Civil Code 24, Family 4 (in Japanese). Yuhikaku Publishing Inc., Japan (1994)
8. Ninomiya, S.: New Annotated Civil Code 17, Family 1 (in Japanese). Yuhikaku Publishing Inc., Japan (2017)
9. Oho, F., Nakagawa, J.: New Annotated Edition-The Civil Code 25, Family 5 (in Japanese). Yuhikaku Publishing Inc., Japan (2004)

Effects of Relations between Normal Logic Programs and Defeasible Logic Programs on Contrary Prioritized Policy

Wachara Fungwacharakorn[1], Kanae Tsushima[1], and Ken Satoh[1]

National Institute of Informatics, Sokendai University, Tokyo, Japan
tonwachara@gmail.com,{k_tsushima,ksatoh}@nii.ac.jp

Abstract. There are two common types of logic programs for representing statutes, one is a normal logic program and another is a defeasible logic program. However, a literal in a defeasible logic program can be ambiguous, which means a program may support both the literal and its contrary. This paper studies a *contrary prioritized* policy to resolve such a conflict. The policy restricts a defeasible theory to prioritize all pairs of rules in which heads are contrary to each other. This paper investigates how relations of normal logic programs and defeasible logic programs affect the policy. We found that, for a translation of a defeasible logic program to a stratified normal logic program, the contrary prioritized policy is necessary. In the same manner, a translation of a normal logic program to a defeasible logic program induces the policy indirectly and variously from case to case.

Keywords: Legal reasoning · Legal representation · Priority

1 Introduction

For a long time, researchers have been interested in using logic programs to represent statutes. There are two types of logic programs that are commonly used to represent statutes and legal theories. The first type is a normal logic program, in which each rule is an extended Horn clause with negation as failure. This includes many programming languages for representing statutes such as Prolog [26, 28, 29], Proleg [24], Catala [17]. The second type is a defeasible logic program, in which each rule is a defeasible rule that does not use negation as a failure, but instead uses a priority between rules. This includes several frameworks for legal reasoning such as legal reasoning in Defeasible Logic [19] and legal reasoning in ASPIC+ [21].

One different feature between a normal logic program and a defeasible logic program is that a literal in a defeasible logic program can be *ambiguous* [4] when there are at least two chains of monotonic reasoning such that one supports the literal but another supports its contrary, and a priority cannot resolve this conflict. In this paper, we explore a *contrary prioritized* policy – a policy which prioritizes all pairs of rules in which heads are contrary to each other, hence

ambiguous literals can be resolved if a dependency graph of a program is acyclic and a priority is transitive. In this paper, we show that the policy is required for a translation of a defeasible logic program to a stratified normal logic program and the policy is implicitly induced and may vary from case to case in a translation of a normal logic program to a defeasible logic program.

This paper is structured as follows. Section 2 provides basic definitions of normal logic programs and defeasible logic programs. Section 3 explains ambiguity in defeasible logic programs and the contrary prioritized policy. Section 4 explores effects of relations between normal logic programs and defeasible logic programs on the policy. Section 5 compares the contrary prioritized policy to other policies dealing with ambiguity. Finally, section 6 provides a conclusion of this paper.

2 Basic Definitions

In this paper, we restrict attention to propositional logic programs for representing laws in civil law systems, where statutes are the primary source of a decision. In civil law systems, judges usually develop legal theories from statutes in order to make consistent interpretations across courts. For example, judges in the Japanese Legal Training Institute developed a legal theory known as the Japanese Presupposed Ultimate Facts Theory or *Yoken-jijitsu-ron* [13]. Because a legal theory is very similar to a computational logic program, it is simple to codify a legal theory into a logic program.

2.1 Normal Logic Program

There are two types of logic programs that are commonly used to codify a legal theory. One common type is a normal logic program or a PROLOG program. PROLOG is one of the most popular logic programming languages and it was the first logic programming language for representing laws [28]. Although recent programming languages for laws, such as PROLEG[24] and CATALA[17], typically separate exception parts from requisite parts, it has been demonstrated that such separation has the same expressive power as original PROLOG [25].

Definition 1 (Normal Logic Program). *A normal logic program is a set of rules of the form*

$$h \leftarrow b_1, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n. \tag{1}$$

where $h, b_1, \ldots, b_n$ are propositions. Let R be a rule of the form (1), we have

- *h as a head of a rule or a conclusion of a rule denoted by $head(R)$,*
- *$\{b_1, \ldots, b_m\}$ as a positive body of a rule denoted by $pos(R)$ (each element of a positive body is called a requisite),*
- *$\{b_{m+1}, \ldots, b_n\}$ as a negative body of a rule denoted by $neg(R)$ (each element of a negative body is called an exception),*

We express h. (called a fact) if the body of the rule is empty. A rule is called a Horn clause *if the negative body of the rule is empty.*

not in a rule is a negation as failure, namely *not* p is derived if p cannot be derived. In legal representation, negations as failure typically represent rules with explicit exceptions expressed by "except", "unless", etc.

In civil litigation, a judge takes factual situations in a case as inputs. Then, the judge concludes a legal decision based on related statutes. To reflect this civil litigation, we divide propositions into *fact propositions*, which must not occur in the head of a rule (except the rule with empty body i.e. a fact), and *rule propositions* for otherwise. Let F be a set of fact propositions and R_n be a set of rules. $\langle F, R_n \rangle$ is a normal logic program built by appending facts (rules with empty bodies) constructed from all fact propositions in F into R_n. A set of propositions which can be concluded $\langle F, R_n \rangle$ is called an *answer set*. In other words, an answer set indicates a legal decision in the case (represented by F) based on the statute (represented by R_n).

2.2 Defeasible Logic Program

A defeasible logic program [18] is another common logic program for codifying a legal theory. In a defeasible logic program, there are three kinds of rules: *strict* rules, *defeasible* rules, and *defeaters*. For the sake of simplicity, in this paper, we focus on a defeasible logic program that only contains defeasible rules.

Definition 2 (Defeasible Logic Program). *A defeasible logic program is a set of defeasible rules of the form*

$$l_0 \Leftarrow l_1, \ldots, l_n. \tag{2}$$

where $l_0, \ldots, l_n$ are literals, which are propositions or classical negations of propositions. Let l be a literal, we say $\overline{l}$ is contrary to l, namely $\overline{p} = \neg p$ and $\overline{\neg p} = p$ where p is a proposition. Let R be a rule of the form (2), we have

- *l_0 as a head of a rule or a conclusion of a rule denoted by head(R),*
- *$\{l_1, \ldots, l_n\}$ as a body of a rule denoted by body(R).*

Rules in defeasible logic programs interact via priorities. Hence, knowledge in defeasible logic is structured as a defeasible theory defined as follows.

Definition 3 (Defeasible Theory). *A* defeasible theory *is a triple $(F, R_d, >)$ where*

- *F is a consistent set of literals constructed from fact propositions (that is p and $\neg p$ cannot occur in the same F)*
- *R_d is a set of defeasible rules*
- *$>$ is a* priority *- an ordering relation on R_d [1] such that for all $r_i, r_j, r_k \in R_d$, $r_i \not> r_i$ (irreflexive) and if $r_i > r_j$ and $r_j > r_k$ then $r_i > r_k$ (transitive). If $r_i > r_j$, we say r_i has a higher priority than r_j or r_i overrides r_j.*

[1] Generally, $>$ in defeasible logic is not required to be transitive. However, in this paper, we assume the transitivity to ensure that a closure of the priority is acyclic.

Defeasible logic has its own proof theory [18]. Generally speaking, a conclusion is derived when it is supported by some rules and there are no counter-attacks or all counterattacks are from lower prioritized rules. There are several types of derivation in defeasible logic but we focus only on defeasible derivations. Following [4], general defeasible derivations are denoted by a proof tag d_f i.e. $(F, R_d, >) \vdash +d_f l$ if a literal l is defeasibly provable and $(F, R_d, >) \vdash -d_f l$ if a literal l is defeasibly refuted. In this paper, we restrict only a *decisive* defeasible theory, of which a dependency graph is acyclic. In such a theory, a literal can be either defeasibly provable or defeasibly refuted (but not both) [10].

In computational legal reasoning, a defeasible theory with a priority is useful for representing legal rules that do not have explicit exceptions. Example 1 demonstrates an example of such rules from the Brazilian Penal Code.

Example 1 (The Brazilian Penal Code). Considering these two articles in the Brazilian Penal Code regulating about abortion [16].

> **Article 124** ... *abortion is forbidden* ...
> **Article 128** ... *It is permitted to abort upon consent if the woman's life is endangered or if pregnancy is the result of sexual abuse* ...

The articles can be represented by the following defeasible logic program [2].

r_1 : guilty-due-to-abortion $\Leftarrow$ abort.

r_2 : $\neg$guilty-due-to-abortion $\Leftarrow$ abort, upon-consent, woman-life-endangered.

r_3 : $\neg$guilty-due-to-abortion $\Leftarrow$ abort, upon-consent, pregnancy-from-sexual-abuse.

From the example, there is no explicit exception. However, legal scholars often interpret that Article 128 overrides Article 124 as it covers more specific cases than Article 124. This corresponds to a legal principle *lex specialis* – a specific act can override a general act. It is one of many legal principles that express priorities among rules [12]. Given a case $F = \{abort, upon\text{-}consent, woman\text{-}life\text{-}endangered\}$ and a priority $\{r_2 > r_1, r_3 > r_1\}$ expressing the interpretation by legal scholars, we have that $(F, R_d, >) \vdash +d_f \neg guilty\text{-}due\text{-}to\text{-}abortion$ and $(F, R_d, >) \vdash -d_f guilty\text{-}\ due\text{-}to\text{-}abortion$ since r_2 overrides r_1.

3 Ambiguity and Contrary Prioritized Policy

In contrast to normal logic programs, some defeasible theories may contain an ambiguous literal l such that there is a monotonic chain of reasoning that supports l, another that supports the contrary $\bar{l}$, and the priority does not resolve this conflict [4]. Typically, there are two common policies to deal with ambiguity in defeasible logic. One is an *ambiguity blocking* (denoted by a proof tag ∂), which intuitively means both derivations of such l and $\bar{l}$ are determined as

[2] This is a simplified representation for ease of exposition. A proper representation requires considering deontic modalities and strengths of permission (see [11] for example).

failure. Another is an *ambiguity propagation* (denoted by a proof tag $\int$), which intuitively means both derivations of such l and $\bar{l}$ are determined as success. Example 2 shows a defeasible theory with an ambiguous literal.

Example 2. Let R_d and $>$ be a set of defeasible rules and a priority as follows.

$$r_1 : \quad p \Leftarrow a.$$
$$r_2 : \neg p \Leftarrow b.$$
$$r_3 : \quad p \Leftarrow c.$$
$$r_2 > r_1$$

If $F = \{b, c\}$, we have that p is ambiguous in $(F, R_d, >)$ since the heads of r_2 and r_3 are contrary to each other and the priority cannot resolve this conflict. If we consider an ambiguity blocking, we have that $(F, R_d, >) \vdash -\partial p$ and $(F, R_d, >) \vdash -\partial \neg p$. If we consider an ambiguity propagation, we have that $(F, R_d, >) \vdash +\int p$ and $(F, R_d, >) \vdash +\int \neg p$. However, p may not be ambiguous in other cases F' such as $F' = \{a, b\}$.

A literal in a defeasible theory can be ambiguous because defeasible logic allows classical negations in the heads of the rules and a priority may not be covered enough to resolve the conflicts between two chains of reasoning. In this paper, we explore a *contrary prioritized* policy to reduce ambiguities in defeasible theories. The policy refers to a restriction of having a priority for every pair of rules in which heads are contrary to each other.

Definition 4 (Contrary Prioritization). *Let $D = (F, R_d, >)$ be a decisive defeasible theory of which a dependency graph is acyclic. We say D is* contrary prioritized *if for all $r_i, r_j \in R_d$ and $head(r_i) = \overline{head(r_j)}$, $r_i > r_j$ or $r_j > r_i$.*

Continuing from Example 1, Example 3 demonstrates the contrary prioritized policy in the fictional example from the Brazilian Penal Code.

Example 3. Suppose there was a case that a woman had a pregnancy as a result of sexual abuse and went to a non-authorized abortion clinic for having an abortion upon her consent. The court may consider that going to a non-authorized abortion clinic shows an intention of abortion hence it should be guilty. Then, legislators may consider this case and introduce a rule r_4 overriding r_3 ($r_4 > r_3$) to reflect the court decision. However, since we do not know a priority between r_2 and r_4, it leads to ambiguity in legal interpretation when there is a case in which a woman's life is endangered but she goes to a non-authorized clinic.

r_1 : guilty-due-to-abortion $\Leftarrow$ abort.
r_2 : $\neg$guilty-due-to-abortion $\Leftarrow$ abort, upon-consent, woman-life-endangered.
r_3 : $\neg$guilty-due-to-abortion $\Leftarrow$ abort, upon-consent, pregnancy-from-sexual-abuse.
r_4 : guilty-due-to-abortion $\Leftarrow$ non-authorized-abort.
$(r_2 > r_1), (r_3 > r_1), (r_4 > r_3)$

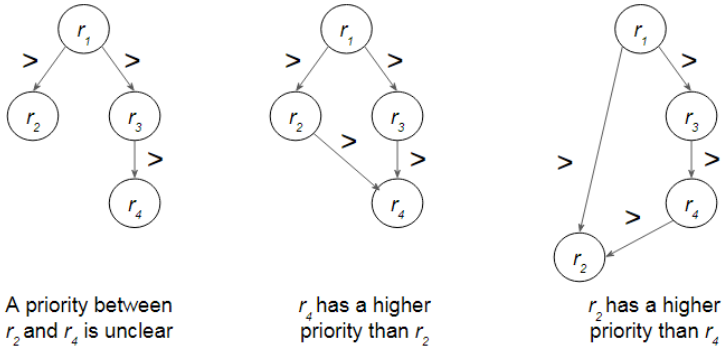


Fig. 1. Possible priorities between r_2 and r_4

From Example 3, there are three possible priorities between r_2 and r_4 as in Fig 1. In compliance with the contrary prioritized policy, the legislators have to consider the anticipated derivation of *guilty-due-to-abortion* in a case in which both r_2 and r_4 are applicable. Then, a legal reasoning system can try all possible priorities between r_2 and r_4 to see which one leads to the anticipated derivation.

4 Exploring Contrary Prioritized Policy

This paper aims to explore how relations of normal logic programs and defeasible logic programs affect the contrary prioritized policy. To achieve this, we first explore a translation of a normal logic program to a defeasible logic program then we explore a translation of a defeasible logic program to a normal logic program in reverse.

4.1 Effect of Defeasible-to-Normal Logic Program Translation

The contrary prioritized policy is necessary for stratifying a normal logic program when translating a decisive defeasible theory into a normal logic program. The definition of a stratified normal logic program shows below.

Definition 5 (A stratified program). *A normal logic program T is stratified if there is a partition $T = T_0 \cup T_1 \cup \ldots \cup T_n$ (T_i and T_j disjoint for all $i \neq j$) such that*

- *if a proposition p is in a positive body of rule in T_i then a rule with p in the head is only contained within $T_0 \cup T_1 \cup \ldots T_j (j \leqslant i)$*
- *if a proposition p is in a negative body of rule in T_i then a rule with p in the head is only contained within $T_0 \cup T_1 \cup \ldots T_j (j < i)$*

One advantage of a stratified program is that it would have only one answer set [7]. This reflects the constraint that judges require a distinct and unambiguous interpretation of statutes.

A translation from a defeasible theory to a normal logic program is as follows [1, 5]. If there is a pair of non-prioritized rules in which heads are contrary to each other, then we cannot translate it into a stratified normal logic program. For each pair of prioritized rules in which heads are contrary to each other

$$r_1 : l_0 \Leftarrow l_{11}, \ldots, l_{1m}.$$
$$r_2 : \overline{l_0} \Leftarrow l_{21}, \ldots, l_{2n}.$$

where l_0, l_{ij} are literals, and $r_2 > r_1$, we can translate them into a normal logic program by introducing *not* e where e is a new proposition for each pair of such rules [3, 22]:

$$r_1' : l_0 \Leftarrow l_{11}, \ldots, l_{1m}, not\ e.$$
$$r_2' : e \Leftarrow l_{21}, \ldots, l_{2n}.$$

Then, we treat each left classical negation $\neg\, p$ as a new proposition. If R_n is a stratified normal logic program translated from a defeasible logic program R_d and a dependency graph of R_d is acyclic, we have that [1]:

- $(F, R_d, >) \vdash +\partial p$ if and only if $\langle F, R_n \rangle \vdash p$.
- $(F, R_d, >) \vdash -\partial p$ if and only if $\langle F, R_n \rangle \nvdash p$.

If a decisive defeasible theory can be translated into a stratified normal logic program, we can infer that such a defeasible theory is contrary prioritized by a partition defined in Definition 5 into a priority between pairs of rules in which heads are contrary to each other.

Example 4 demonstrates a translation of a defeasible logic program from Example 1 to a normal logic program. The result can be interpreted as *abortion is forbidden except the woman's life is endangered or the pregnancy is the result of sexual abuse.* This interpretation corresponds to the legal principle *exceptio firmat regulam* – general rules can apply unless exceptional situations occur.

Example 4. We can translate the defeasible logic program in Example 1 into a normal logic program as follows.

guilty-due-to-abortion ← abort, *not* exception1, *not* exception2.
exception1 ← abort, upon-consent, woman-life-endangered.
exception2 ← abort, upon-consent, pregnancy-from-sexual-abuse.

4.2 Effect of Normal-to-Defeasible Logic Program Translation

A translation from a normal logic program to a defeasible program is as follows [14]. For each rule of the form (1) in Section 2.1, we can translate them into a

set of defeasible rules [3]:

$$\begin{aligned} r_0 : &\quad h \Leftarrow b_1, \ldots, b_m. \\ r_1 : &\neg h \Leftarrow b_{m+1}. \\ &\vdots \\ r_k : &\neg h \Leftarrow b_{m+k}. \\ r_1 > r_0, \ldots, r_k > r_0 & \end{aligned}$$

However, some normal logic programs have implicit priorities. Example 5 demonstrates one normal logic program with an implicit priority.

Example 5. Considering this normal logic program

$$\begin{aligned} p \leftarrow a, not\ b. \\ p \leftarrow c. \end{aligned}$$

which can be translated into the following defeasible logic program

$$\begin{aligned} r_1 : &\quad p \Leftarrow a. \\ r_2 : &\neg p \Leftarrow b. \\ r_3 : &\quad p \Leftarrow c. \\ r_3 > r_2 > r_1 & \end{aligned}$$

Unlike Example 2, we have that r_3 implicitly overrides r_2 in the original program. Hence, there is an implicit priority $r_3 > r_2$. Furthermore, some normal logic programs cannot be translated into a contrary prioritized defeasible theory with the same priority for all cases.

Example 6. Considering this normal logic program

$$\begin{aligned} p \leftarrow a, not\ b. \\ p \leftarrow c, not\ d. \end{aligned}$$

which can be translated into the following defeasible logic program

$$\begin{aligned} r_1 : &\quad p \Leftarrow a. \qquad r_3 : \quad p \Leftarrow c. \\ r_2 : &\neg p \Leftarrow b. \qquad r_4 : \neg p \Leftarrow d. \\ &r_2 > r_1,\ r_4 > r_3 \end{aligned}$$

[3] Another translation is to simulate a negation as failure *not* b as $r_1 : b' \Leftarrow . \ r_2 : \neg b' \Leftarrow b.$ where $r2 > r1$ and b' is a new proposition corresponding to *not* b [2]. This translated theory is definitely contrary prioritized since those contraries occurring in the heads of rules are all new propositions.

From Example 6, we cannot say r_3 always overrides r_2 in general because in some cases, e.g. $\{a, b, c\}$, r_3 overrides r_2, but in some cases, e.g. $\{a, c, d\}$, r_1 overrides r_4 hence r_2 overrides r_3 due to transitivity. As a result, this theory does not have a general priority but a priority must vary from case to case in compliance with the contrary prioritized policy. This example is also related to a concept of *team defeat* [15] where the contrary prioritized policy is not required for avoiding ambiguity because a set of rules would be packed as a team to defeat other rules. For example, a defeasible theory with a case $\{a, b, c, d\}$ can derive $\neg p$ with a priority $\{r_2 > r_1, r_4 > r_3\}$ although the theory is not contrary prioritized.

Since a normal logic program has implicit priorities, the contrary prioritized policy suggests that we may overlook some implicit priorities that are constructed when we introduce a new rule in a normal logic program. Continuing from the normal logic program in Example 4, the following two revisions demonstrate all of the possible implicit priorities introduced by the new rule in the same manner as all of the possible priorities between r_2 and r_4 in Example 3. By using the relation of a normal logic program and a defeasible logic program, a legal reasoning system can take advantage to check intentions of these implicit priorities with a user. The system can ask the user whether some interesting cases should be distinguished (e.g. a case that makes r_2 and r_4 applicable in Example 3), then the system can revise a priority to reduce ambiguity in legal interpretations as the user anticipated.

Revision 1 (equivalent to $r_2 > r_4$)

guilty-due-to-abortion ← abort, *not* exception1, *not* exception2.
exception1 ← abort, upon-consent, woman-life-endangered.
exception2 ← abort, upon-consent, pregnancy-from-sexual-abuse, **not exception3**.
exception3 ← non-authorized-abort.

Revision 2 (equivalent to $r_4 > r_2$)

guilty-due-to-abortion ← abort, *not* exception1, *not* exception2.
exception1 ← abort, upon-consent, woman-life-endangered, **not exception3**.
exception2 ← abort, upon-consent, pregnancy-from-sexual-abuse, **not exception3**.
exception3 ← non-authorized-abort.

5 Discussion

In this paper, we investigate a defeasible logic program when there are two monotonic reasoning chains, one supports an anticipated conclusion but another supports its contrary, and a priority cannot resolve this conflict. This phenomenon

is usually referred to as *ambiguity*. This paper mainly investigates a contrary prioritized policy, preventing ambiguity by having a priority for every pair of rules in which heads are contrary to each other. This policy ensures that, with a transitive priority, we can always resolve such a conflict in a defeasible logic program with an acyclic dependency graph.

In this section, we compare the contrary prioritized policy to other policies dealing with ambiguity, especially *ambiguity blocking* and *ambiguity propagation* (see Section 3) which are two most common policies in defeasible logic. Although ambiguity blocking and ambiguity propagation have different behaviors, they have the same function in civil litigation, i.e. making a conclusion without adding new priorities[4] (for further discussion, see [4, 9, 30]). The effort of minimizing new priorities or exceptions is usually found in civil litigation as judges tend to limit themselves from legislative issues. This effort is usually referred to as *minimal revision* i.e. revising the theory as few as possible, only to justify the decision for the new case [6, 10, 23].

However, the contrary prioritized policy would be more appropriate for legislators than judges since the policy respects the coherentist revision (usually from legislators' perspective) rather than the minimal revision (usually from judges' perspective). From the coherentist perspective, a defeasible theory with ambiguous literals leads to varied and possibly unanticipated decisions since judges can choose any policies to deal with ambiguity. To prevent unanticipated decisions, the coherentist tends to avoid ambiguous or inconsistent conclusions in the first place (see [16]).

Example 3 is a good demonstration that the contrary prioritized policy respects the coherentist revision rather than the minimal revision. In compliance with the contrary prioritized policy, we need to consider a case in which a woman's life is endangered but she goes to a non-authorized clinic, where both r_2 and r_4 are applicable. This case is a prototypical case that did not go to the court so the revision is beyond the justification in the decision for the new case. However, the revised theory has no ambiguous literals hence it meets the goal of the coherentist. Hence, the policy serves as heuristics for revising a defeasible legal theory to prevent unanticipated consequences from the coherentist legislator's point of view.

6 Conclusion

In this paper, we explore a *contrary prioritized* policy, prioritizing every pair of rules whose heads are contrary to each other, hence it can reduce ambiguities in legal interpretations. We have demonstrated the policy by checking priorities of rules in which a head is contrary to a newly added rule and let the reasoning system try all possible priorities to see which one gives an anticipated result. Furthermore, we explore effects of relations between normal logic programs and

[4] This function is also related to a burden of proof, which is established for dealing with the situation where we cannot make a conclusion in civil litigation. There are many formalizations of the burden of proof, such as [8, 20, 27].

defeasible logic programs on the contrary prioritized policy. We found that the policy is required when translating a defeasible logic program to a stratified normal logic program and the priority is implicitly induced and may vary from case to case when translating a normal logic program to a defeasible logic program. From this study, we present a system to ask the user the anticipated conclusion of some prototypical cases to revise a priority in a defeasible theory so that the revised theory gives anticipated and unambiguous decisions, which is typically the goal of coherentist legislators.

Acknowledgements. The authors would like to thank all anonymous reviewers for insightful suggestions and comments. This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Antoniou, G.: Relating defeasible logic to extended logic programs. In: Hellenic Conference on Artificial Intelligence. pp. 54–64. Springer (2002)
2. Antoniou, G., Maher, M.J., Billington, D.: Defeasible logic versus logic programming without negation as failure. The Journal of Logic Programming **42**(1), 47–57 (2000)
3. Apt, K.R., Blair, H.A., Walker, A.: Towards a theory of declarative knowledge. In: Foundations of deductive databases and logic programming, pp. 89–148. Morgan Kaufmann, Burlington, MA,USA (1988)
4. Billington, D., Antoniou, G., Governatori, G., Maher, M.: An inclusion theorem for defeasible logics. ACM Transactions on Computational Logic (TOCL) **12**(1), 1–27 (2010)
5. Brewka, G.: On the relationship between defeasible logic and well-founded semantics. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 121–132. Springer (2001)
6. Fungwacharakorn, W., Tsushima, K., Satoh, K.: On semantics-based minimal revision for legal reasoning. In: Proceedings of the 18th international conference on Artificial intelligence and law - ICAIL '21. ACM, NY, USA (2021)
7. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, Kenneth (eds.) Proceedings of International Logic Programming Conference and Symposium. vol. 88, pp. 1070–1080. MIT Press, Cambridge, MA, USA (1988)
8. Governatori, G., Maher, M.J.: Annotated defeasible logic. Theory and Practice of Logic Programming **17**(5-6), 819–836 (2017)
9. Governatori, G., Maher, M.J., Antoniou, G., Billington, D.: Argumentation semantics for defeasible logic. Journal of Logic and Computation **14**(5), 675–702 (2004)
10. Governatori, G., Olivieri, F., Cristani, M., Scannapieco, S.: Revision of defeasible preferences. International Journal of Approximate Reasoning **104**, 205–230 (2019)
11. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing strong and weak permissions in defeasible logic. Journal of Philosophical Logic **42**(6), 799–829 (2013)

12. Governatori, G., Olivieri, F., Scannapieco, S., Cristani, M.: Superiority based revision of defeasible theories. In: International Workshop on Rules and Rule Markup Languages for the Semantic Web. pp. 104–118. Springer (2010)
13. Ito, S.: Basis of Ultimate Facts. Yuhikaku (2001)
14. Kakas, A.C., Mancarella, P., Dung, P.M.: The acceptability semantics for logic programs. In: ICLP. vol. 94, pp. 504–519 (1994)
15. Maher, M.J.: Propositional defeasible logic has linear complexity. Theory and Practice of Logic Programming **1**(6), 691–711 (2001)
16. Maranhão, J.S.: Conservative coherentist closure of legal systems. In: Logic in the Theory and Practice of Lawmaking, pp. 115–136. Springer (2015)
17. Merigoux, D., Chataing, N., Protzenko, J.: Catala: A Programming Language for the Law. In: International Conference on Functional Programming. pp. 1–29. Proceedings of the ACM on Programming Languages, ACM, Virtual, South Korea (Aug 2021)
18. Nute, D.: Defeasible logic. In: International Conference on Applications of Prolog. pp. 151–169. Springer (2001)
19. Prakken, H.: Logical tools for modelling legal argument: a study of defeasible reasoning in law. Kluwer Academic Publishers (1997)
20. Prakken, H., Sartor, G.: Formalising arguments about the burden of persuasion. In: Proceedings of the 11th international conference on artificial intelligence and law. pp. 97–106 (2007)
21. Prakken, H., Wyner, A., Bench-Capon, T., Atkinson, K.: A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. Journal of Logic and Computation **25**(5), 1141–1166 (2015)
22. Przymusinska, H., Przymusinski, T.: Semantic issues in deductive databases and logic programs. In: Formal techniques in artificial intelligence. Citeseer (1990)
23. Rotolo, A., Roversi, C.: Constitutive rules and coherence in legal argumentation: The case of extensive and restrictive interpretation. In: Legal Argumentation Theory: Cross-Disciplinary Perspectives, pp. 163–188. Springer Netherlands, Dordrecht, The Netherlands (2013)
24. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology. In: Onada, T., Bekki, D., McCready, E. (eds.) New Frontiers in Artificial Intelligence. pp. 153–164. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
25. Satoh, K., Kogawa, T., Okada, N., Omori, K., Omura, S., Tsuchiya, K.: On generality of PROLEG knowledge representation. In: Proceedings of the 6th International Workshop on Juris-informatics (JURISIN 2012), Miyazaki, Japan. pp. 115–128 (2012)
26. Satoh, K., Kubota, M., Nishigai, Y., Takano, C.: Translating the Japanese Presupposed Ultimate Fact Theory into logic programming. In: Proceedings of the 2009 Conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference. pp. 162–171. IOS Press, Amsterdam, The Netherlands (2009)
27. Satoh, K., Tojo, S., Suzuki, Y.: Formalizing a switch of burden of proof by logic programming. In: Proceedings of the First International Workshop on Juris-Informatics (JURISIN 2007). pp. 76–85 (2007)
28. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. Communications of the ACM **29**(5), 370–386 (Apr 1986)

29. Sherman, D.M.: A prolog model of the income tax act of Canada. In: Proceedings of the 1st international conference on Artificial intelligence and law. pp. 127–136. Association for Computing Machinery, New York, NY, USA (1987)
30. Stein, L.A.: Resolving ambiguity in nonmonotonic inheritance hierarchies. Artificial Intelligence **55**(2-3), 259–310 (1992)

Named Entity Recogntion on Judgments of High Courts in India

Rishabh Singhal, Sanyukta Tanwar, and Kshitiz Verma

JK Lakshmipat University, Jaipur, Rajasthan, India
{rishabhsinghal, sanyuktatanwar, kshitiz.verma}@jklu.edu.in

Abstract. Named entity recognition(NER) is the process of locating and identifying key nouns and proper nouns in a text. Legal data differs significantly from the texts in general corpus. There are some common words that have different meaning in legal domain and some technical terms which are not their in the general corpus. We first used the standard model of Spacy for NER on our data but it doesn't provide satisfactory result as we needed more precise details. After that we customized tags and tagged 20,000 words using BIO tagging format. We used total 28 tags in which 8 were of legal domain and 20 were general tags. Then we trained our data on two different named entity recognition models, namely, Bidirectional LSTM and BERT. We found that BERT model was not predicting good results in BIO format as it was unable to catch the near context word properly whereas Bidirectional LSTM outperformed BERT and was able to predict nearly all the tags with good accuracy.

Keywords: NER · BIO tagging · Legal data · India · Bidirectional LSTM · BERT.

1 Introduction

Natural language processing(NLP) using deep learning is being increasingly used in the legal domain, for legal text segmentation [17], legal subject categorization [18, 19], legal judgment prediction and analysis[2], legal information extraction[9], legal question answering[13, 14], etc. Modern neural NLP methods typically rely on pre-trained word embeddings or pre-trained language models. Word embedding is a way of representing text and documents. A numeric vector input that represents a word in a lower-dimensional space is known as Word Embedding or Word Vector. It permits words with comparable meanings to be represented in the same way. We need sequence labeling for preprocessing data in all of the above NLP applications. Sequence labeling can be accomplished in a variety of ways; one of the most common data preprocessing tasks is named entity recognition (NER).

NER includes recognizing key information in a text and categorizing it into a set of predetermined categories. An entity is essentially the object that is mentioned or referred to repeatedly throughout the text. As we can see in Fig. 1

Gopal Ramnath is a person entity and *Agra Cantt Railway Station* is a location entity. We can further classify an entity based on its beginning, inside and outside respectively. We apply the BIO/IOB tagging format (short for beginning, inside, and outside). The B- prefix before a tag indicates that it is at the start of a entity, while the I- prefix indicates that it is within a entity. A token with an O tag does not belong to any entity.

Fig. 1. Example explaining Named Entity Recognition.

Our focus is on the legal dataset where we have different judgment files in different formats of unstructured data. Also, the legal text has different characteristics compared to generic corpora, such as specialized vocabulary, peculiar syntax, semantics based on considerable domain-specific knowledge, and so on[5]. Some of the major differences are:

- Technical terms that are unfamiliar to the general public are used in legal English.
- In the legal text, there are other common words that have unique meanings.
- The word order used in several cases is noticeably different when compared to regular English.

In the legal world, a major task in proceeding with any case is to read and understand that case file as well the files with similar records. It is tough to process the legal documentations for automated tasks. Named entity recognition can be used to reduce this problem and provide an overview of the judgement. The goal of our research is to find an efficient method for Named Entity Recognition (NER). We used two models that are bidirectional LSTM and BERT. These models showed different behavior over legal text.

2 Literature Review

There are various rule-based NER systems that can provide 88 percent to 92 percent f-measure accuracy for English and include lexicalized grammar, gazetteer lists, and a list of trigger words [24]. The main drawbacks of rule-based approaches are that they necessitate a great deal of experience and grammatical knowledge of the target language or domain and that they are not transferrable to other languages or domains. To obtain high-level linguistic knowledge, ML-based approaches for NER make use of a huge amount of NE annotated training data.

The Hidden Markov Model was used, giving the improvement on F-measure [22], Maximum Entropy [12], and Conditional Random Field [20] are the most

commonly utilized ML approaches for the NER challenge. Different ML methods are also used in combination. To create a NER system, Srihari et al. (2000) integrate Maximum Entropy, Hidden Markov Model, and customized rules. For recognizing names, NER systems employ gazetteer lists[21]. Gazetteer lists are used in both the linguistic [24] and the ML-based approaches[21, 22].

Hand-crafted rules are used in the linguistic method, which necessitates the involvement of professional linguists. Some modern systems use machine learning to discover context patterns, reducing the amount of manual effort. The authors in [11] used a combination of grammatical and statistical methodologies to generate high-precision NE extraction patterns. They offered a method for learning lexical patterns for Indian 4 languages . In Greek legislation, researchers tested LSTM-based approaches for named entity recognition, demonstrating the utility of such techniques in foreign language contexts. In Resource description framework(RDF), they established and used a new language for representing textual references. They also looked at entity linking between textual references and entities from third-party datasets, as well as creating a new dataset for Greek geo-landmarks[3]. Similarly, [10] and [4] have worked on NER specific to legal domain for German and French judgments respectively.

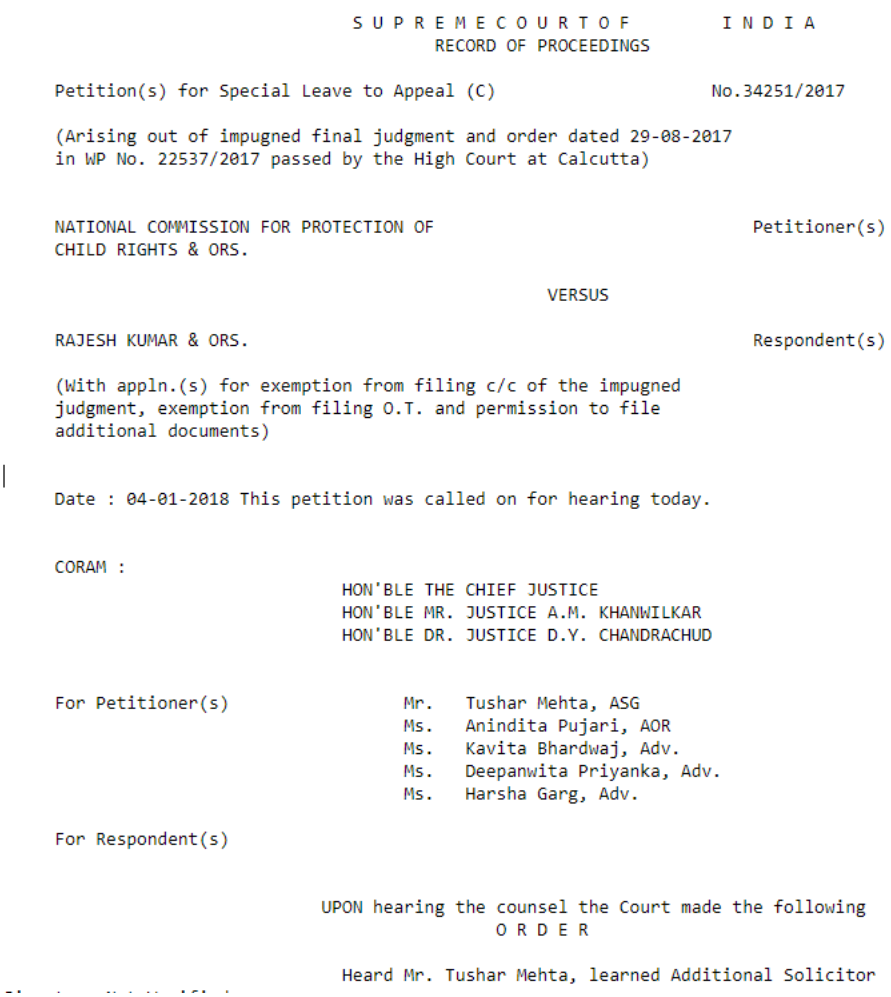
The author in [6] used Support Vector Machine(SVM) and Logistic regression in their work for extraction of contract elements. They also employed deep learning to improve tasks by adding a CRF layer on top of the BILSTM or stacking an extra LSTM on top of the BILSTM. When techniques are examined for their capacity to extract multi-token contract parts, the stacked BILSTM-LSTM misclassifies fewer tokens, while the BILSTM-CRF combination performs better. They also improved the querying platform for Greek legislation (Nomothesia) by integrating Name Entity Recognition and the Bi-LSTM Bi-LSTM LR neural model. BERT has already been trained for legal data [7]. However, we chose to train our because the problems faced are more than just the legal text. It is also about the formatting issues that our learning algorithms need to take care of.

3 The Research Problem

Our goal is to do NER on the judgments of Supreme Court of India and various High Courts in India. The failure of the standard models to do even basic NER on legal data promoted us to research the topic. We substantiate the failure of the standard model through the examples provided in the following subsections.

3.1 Standard libraries like Spacy don't work

Our research is on legal data which is different from the general corpus on languages.This problem can be explained by examples using Fig. 2 and Fig. 3. When we apply the standard model of Spacy for NER[1], it performs well over a general corpus, but when it is used for legal judgments(Shown in Fig. 2),it has majorly two issues. First one is lot of incorrect tagging. The model tags *Adv.*



S U P R E M E C O U R T O F I N D I A
RECORD OF PROCEEDINGS

Petition(s) for Special Leave to Appeal (C) No.34251/2017

(Arising out of impugned final judgment and order dated 29-08-2017
in WP No. 22537/2017 passed by the High Court at Calcutta)

NATIONAL COMMISSION FOR PROTECTION OF CHILD RIGHTS & ORS. Petitioner(s)

VERSUS

RAJESH KUMAR & ORS. Respondent(s)

(With appln.(s) for exemption from filing c/c of the impugned judgment, exemption from filing O.T. and permission to file additional documents)

Date : 04-01-2018 This petition was called on for hearing today.

CORAM :
HON'BLE THE CHIEF JUSTICE
HON'BLE MR. JUSTICE A.M. KHANWILKAR
HON'BLE DR. JUSTICE D.Y. CHANDRACHUD

For Petitioner(s) Mr. Tushar Mehta, ASG
Ms. Anindita Pujari, AOR
Ms. Kavita Bhardwaj, Adv.
Ms. Deepanwita Priyanka, Adv.
Ms. Harsha Garg, Adv.

For Respondent(s)

UPON hearing the counsel the Court made the following
O R D E R

Heard Mr. Tushar Mehta, learned Additional Solicitor

Fig. 2. Header of a Judgment of Supreme Court of India

```
['PROCEEDINGS\n\n     Petition(s', 'ORG']
['29', 'CARDINAL']
['WP', 'ORG']
['the High Court', 'ORG']
['VERSUS', 'ORG']
['RAJESH KUMAR & ORS', 'ORG']
['Respondent(s', 'ORG']
['O.T.', 'GPE']
['today', 'DATE']
["KHANWILKAR\n                                        HON'BLE DR", 'ORG']
['Tushar Mehta', 'PERSON']
['Anindita Pujari', 'PERSON']
['AOR', 'ORG']
['Ms.', 'ORG']
['Kavita Bhardwaj', 'PERSON']
['Deepanwita Priyanka, Adv', 'ORG']
['Harsha Garg', 'PERSON']
['Respondent(s', 'ORG']
```

Fig. 3. Output from Spacy model as applied to the judgment Fig. 2 for NER

and *HON'BLE* as *ORG*, but they refer to an advocate and a judge, respectively. *Section 13*, *Chapter 2*, and *Constitution* are all tagged as *LAW*, which is insufficient for the legal processing. We may need more specific tags on the Sections, Articles, etc. Another issue with the spacy model is data extraction. For example, in Figure 5, petitioner names are given like "Mr. Tushar Mehta ,ASG and Ms. Kavita Bhardwaj Adv." , the model is unable to extract these names due to the varying alignment, thereby leaving critical information out. So these examples justify how the standard model of Spacy for NER doesn't give good results in the legal context. Custom Spacy can solve the problem of tagging as we can customize tags but the problem of data extraction will not be solved.

3.2 Different documents may have different formats

Also, if we read the format of files to extract information, we'll have a lot of problems since we'll have to create separate models for different formats and we might not capture all of the information we need. For example, In a judicial judgment sections X, Y, and Z appear. The format reading is incapable of extracting all the sections which are in the text. It only reads the standard format and fails to capture complete information.

3.3 Entity Recognition in legal judgement

NER should enable us to read the files in different formats, even when data is in an unstructured format i.e all formats are not alike. Therefore, we need to develop our own NER model, we'll require manually tagged data to build a

custom NER model. When we discuss a specific domain, we may use specific tags. As a result, while discussing the legal domain, tags for judges, advocates, sections, and so on are required.

- **Person**: In the legal judgments name of persons which are stated in the text(e.g., Judge, Advocate, Respondent, Petitioner, etc.).
- **Organization**: In judgment sometimes petitioner or the respondent is a public or private organization (e.g., National Insurance Company, Rotomac Pens Limited, etc.).
- **Geopolitical Entity**: In legal terms we have some important locations (e.g. court, crime location, etc.)
- **Reference to the Law**: Any mention of Indian law is frowned upon (e.g., Sections, Articles, Chapters, Acts, etc.)

4 Methodology

We have different judgements from different courts of India that follow different formats. Firstly we do parts of speech(POS) tagging and remove the words which are not useful. Then we manually tagged the data in legal context which was required to train our models. BERT and Bi-LSTM are used to perform NER.

4.1 Dataset

Our dataset contains text files of Indian judicial judgments. The dataset contains judgments from several Indian courts, including the Supreme Court of India and the majority of the country's high courts. With approximate 6.1 millions files, we have data from 1947 to 2018. However, we realized that it is not trivial to generalize NER to legal dataset. Hence, we could only use 60 files to define the tags and build the model for NER on legal dataset and the results presented in this paper correspond to these 60 files.

4.2 Data Preprocessing

After reading the text files of the judgments. Our task is to preprocess it and tag the data as required for our model to train.

Tagging is a type of classification that is defined as the automatic assigning of a description to each token. The descriptor here is called a tag, and it can represent a number of things, including parts of speech, semantic information, and so on[16]. Part of speech(POS) tagging is a typical Natural Language Processing process that includes categorizing words in a text (corpus) into one of several parts of speech based on the definition and context of the word[23].

In POS tagging, there are two key steps: first, we must tokenize the word, and then we apply the nltk.pos tag to the tokenized words. So, after POS tagging, we only kept proper nouns, nouns, and numbers from the corpus and exported our data to a CSV file. The main element now is proper name tagging. For proper

name tagging we utilized BIO tagging i.e. "Begining-Inside-Outside" and we manually tagged 20000 words to get a total of 28 tags where 8 are specific to legal data which include section(*B-SEC, I-SEC*), article(*B-ART, I-ART*), judge(*B-JUG, I-JUG*), court(*B-CRT, I-CRT*) and remaining 20 were the general tags which include *B-STATE, B-COUNTRY, I-ORG, B-DAY, B-CITY, I-CITY, B-PLC, I-PLC, I-PER, B-TIM, I-TIM, B-ORG, B-PER, O, I-STATE, B-DATE, I-DATE, B-YEAR, B-MONTH, B-MONEY* .

4.3 Why BIO Tagging ?

The main problem is that if we tag the data with traditional format, the model may not be able to distinguish between a person and a judge. BIO format helps here. For example, *Hon'ble Manoj Gupta* is a judge's name, and *Hon'ble* appears before the judge's name in Indian judgments. A normal format model would tag *Manoj Gupta* as a person, but we need a specific tag, so we can use the BIO format to tag *Hon'ble* as *B-Judge* and *Manoj* and *Gupta* as *I-Judge*. When it comes to section numbers, if we have *Section 14* in our data, we require it to be *SEC NUM*. However, *14* might be classified as *DATE*. Also BIO format can solve many problems for example we have advocates from two side but it is difficult to understand for normally tagged data who is the advocate of applicant and who is for respondent. So BIO format can help us in identifying the relation. As a result, we need to create a model that can accurately determine the close context of words.

5 Model and Architecture

In this section, we discuss the neural network models that we have used to train out data. Bi-LSTM and BERT are two choices explored by us and we have shown that Bi-LSTM model outperforms BERT in our case.

5.1 Bidirectional LSTM

Bidirectional LSTM is just a combination of LSTM units in both forward and backward order to save the context of the previous and next LSTM unit.This trait of bidirectional LSTM helps in the preservation of near context of word. Long short term memory, or LSTM, was created to address the problem of vanishing gradients in recurrent neural networks. LSTM solves this problem in part by employing three gates: read gate, forget gate, and write gate. The forget and output gates determine whether fresh information should be kept or discarded. The model decision is made using the LSTM block's memory and the output gate's condition. The output is then fed again into the network as an input, resulting in a recurrent sequence. LSTM works effectively on long sequences because of this structure[15].
Our Bi-LSTM model is divided into the following three layers:

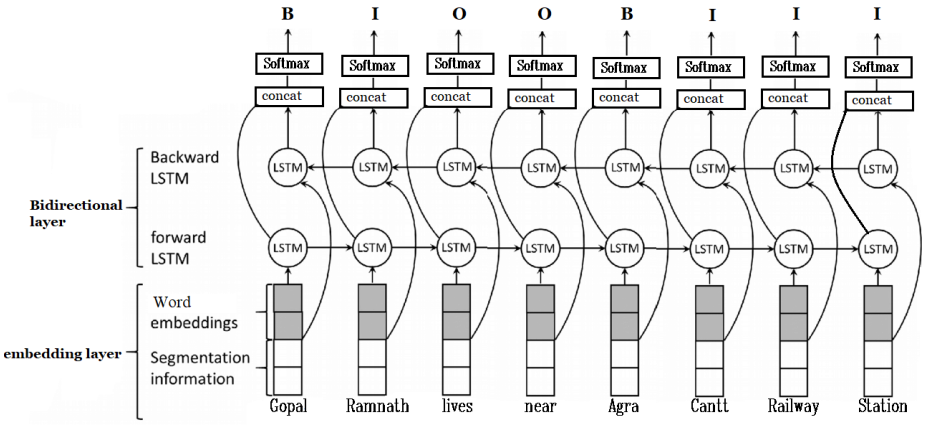


Fig. 4. Bidirectional LSTM diagram.

Embedding layer / Input layer :

This layer is responsible for creating input word embeddings for the Bi-LSTM model. It's done by defining the padded sequence's dimension.As the maximum length of a sentence in our data is 75 the padded sequences will have a maximum length of 75 words. After the network has been trained, the embedding layer will convert each token into an n-dimensional vector (word embeddings). We've set the value of n to 128. The dimensions of the embedding layer's output are (None, 75, 128). The batch sizes are represented by the first argument, 'None.' They display "None" if it is not specified, showing that the model can handle any batch size.

Bi-LSTM layer:

The output from the preceding embedding layer is used in this layer (75, 128). We used 75 LSTM units in this layer. We'll have forward as well as backward outputs because this is a bidirectional LSTM. Before passing it on to the next layer, it concatenates these outputs. As a result, double the number of outputs to the following layer. It becomes 256(128* 2) in our situation.

TimeDistributed Dense layer:

We're working with a Bi-LSTM architecture, which means we're anticipating output from each input sequence. As an example, in the sequence (a_1 b_1, a_2 b_2, $\cdots$, a_n b_n), a and b represent the sequence's inputs and outputs, respectively. Dense(fully-connected) operation across every output over every time step is possible with the TimeDistributeDense layers. This layer will result in a final output of dimension (75,28). The number of words for which tags have been

predicted is represented by 75 units, while the probability of each tag for a particular word is represented by a distribution of 28 units.

5.2 BERT

BERT is a Transformer-based language model that has been pre-trained on big corpora. A task-specific layer is built on top of BERT for each new task, and the model is trained together using task-specific data. For binary classification, multi-label classification, and case importance, we put a linear layer on top of BERT with a sigmoid, softmax, or no activation respectively. We truncate our case descriptions to BERT's maximum length, which affects its performance. This also highlights a significant shortcoming of BERT when it comes to processing large texts, which is a prevalent feature of legal text processing. Since BERT uses absolute position embeddings, it's best to pad the right side of the inputs rather than the left[8].

Masked language modelling (MLM) is used to train BERT. MLM gives BERT a sentence and for optimizing the weights within BERT to produce the identical text on the other side. However, we must mask a few tokens before giving BERT the input sentence because we give BERT a sentence and ask it to reproduce it.

Tokenization:

Tokenization is breaking down sentences into words, or tokens, while also removing specific characters, such as punctuation. BERT can conveniently implement it using the BertTokenizerFast class from a pre-trained BERT base model.We utilized the "bert-base-cased" model with "75 words " as the maximum length of an input sequence, post padding to extra "PAD" tokens at the beginning of sentences smaller than the maximum length of the sequence and set truncation to "TRUE" to restrict tokens that exceed the maximum length of the sequence.

([CLS]) and ([SEP]) tokens are appended to the beginning and end respectively of the tokenized sentence after each word in the input text is tokenized. These tokens serve as an input representation for classification tasks and are used to discriminate between two input sentences.

The length of the input sequence does not match the length of the original label after the tokenization process. The BERT tokenizer incorporates a wordpiece tokenizer, which splits a single word into one or more meaningful sub-words. After the tokenization procedure, there were two big issues. The first was the insertion of special tokens such as PAD, CLS, and SEP from BERT, and the second was the splitting of some tokens into sub-words.The "word ids" method from the tokenization was used to fix this problem because it does not give specific word ids to special tokens from BERT and only provides a word id to the first sub-word while assigning "-100" as an id to the rest of the sub-words.

The Token Embeddings layer will turn each wordpiece token into a 512-dimensional vector representation.

The three embedding layers in BERT are as follows:

Token Embedding layer:

This layer is responsible for determining the embedding for each word in the sequence. The model will learn an "emb-dim" dimensional vector for each word in the sequence during training. For each token, you can think of it as a vector look-up. The members of the vectors are considered as model parameters, and they, like any other weights, are optimized via back-propagation.

Segment Embedding layer:

This layer helps BERT to contextually distinguish each sentence in the input.If there are two input sentences, for example, *I enjoy learning NLP* and *I enjoy learning DBMS*. The model is simply fed the pair of input texts that have been concatenated. These sentences are represented in the model Segment embeddings layer by two vectors. All tokens belonging to input 1 are assigned to the first vector (index 0), whereas all tokens belonging to input 2 are assigned to the second vector (index 1).To produce final output of BERT, these vectors are transmitted through a linear output layer.

Position Embedding layer:

Transformers exist in this layer of BERT encoding of the sequential nature of their inputs from the Embedding layer. As a result,In *Happy new year* and *Good morning* both *Happy* and *Good* will have identical position embeddings because they are the first words in the given sentence.

Modeling and Training:

"BertForTokenClassification" class has been used to model the BERT model. This class is a model that encapsulates the BERT model and adds linear layers that operate as token-level classifiers on top of it. The number of unique tags defined in the dataset is set as the output of each token classifier.

We utilised "adam" as the opytimizer and trained our model across 50 epochs.

6 Result

We have trained our model on a specific legal dataset with 28 tags. Some tags like a person, location, and date, may stay common and can be changed according to the need. However, different domains may have different requirements like legal domain requires tags like section, article, court, etc.

As shown in the Table 1, we used various NER models on a legal dataset. It is evident that Spacy tags common data correctly, but when it comes to legal data, it fails. For example, it tagged "hon'ble" as "ORG," yet BERT and Bi-LSTM output reveals "hon'ble" as "B-JUG." This means that the BERT and Bi-LSTM models are more effectively trained on legal datasets.

When comparing the results of mentioned models, Bi-LSTM is the only model which was able to detect the judge name entity properly and when it is compared with BERT, its been displayed by results that Bi-LSTM was able to detect near

Table 1. Result of Different Models

Words	**Spacy**	**Bi-LSTM**	**BERT**
Hon'ble	ORG	B-JUG	B-JUG
Raj	PER	I-JUG	B-PER
Nath	PER	I-JUG	I-PER
2	DATE	B-DATE	B-DATE
January	DATE	I-DATE	O
,	O	O	O
2017	DATE	B-YEAR	O
Bench	O	O	O
:	O	O	O
Section	LAW	B-SEC	O
14	LAW	I-SEC	O

context by detecting "B" and "I" entity of date and judge tag because of its structural property of taking inputs from both the directions.

Table 2. Scores of Different Models

Model	**Precision**	**Recall**
Spacy	0.809	0.787
Bi-LSTM	0.92	0.92
BERT	0.882	0.882

When comparing Bi-LSTM to BERT, we discover that Bi-LSTM outperforms BERT in terms of accuracy and F1 score, as evidenced by our experimental results, Bi-LSTM model had an accuracy and F1score of 0.9943 and 0.92 respectively where as BERT model had an accuracy and F1 score of 0.9672 and 0.882 respectively. Precision and recall values are shown in the Table 2.

Fig. 5 shows the learning curve for training BERT on our data set. We see that the model is not learning after the first few epochs. We used dropout for regularizing the network with the probability of keeping a neuron to be 0.95. The optimizer used is Adam with learning rate 3×10^{-4} and the value of epsilon is 1×10^{-8}.

Similarly for Fig. 6, we used dropout for regularizing the network with the probability of keeping a neuron to be 0.8. The optimizer used is again Adam with learning rate 0.001 and epsilon value as 1×10^{-8}.

We see a good degree of overfitting in both the models which signify that our claim that Bi-LSTM works better than BERT may not hold as the degree of overfitting is a bit more for BERT compared to the Bi-LSTM model. However, we think that the reason for Bi-LSTMs to work better is that they capture short ranged dependencies much better. For example, Hon'ble is used just before the name of a judge, or in "Section 14", Section appears just before a number so our

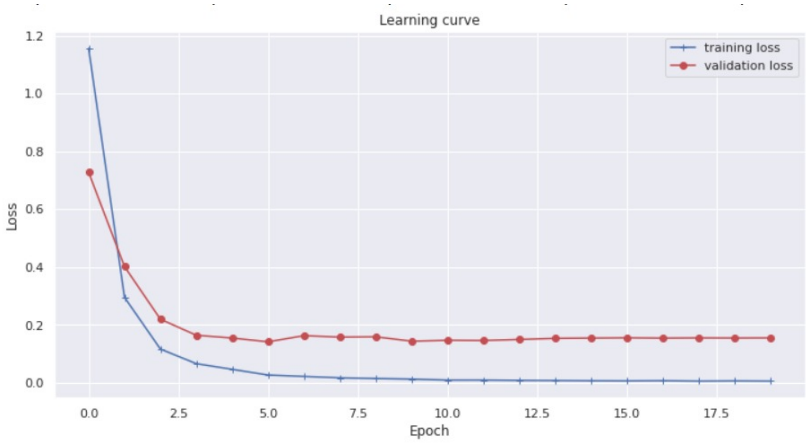


Fig. 5. BERT learning curve.

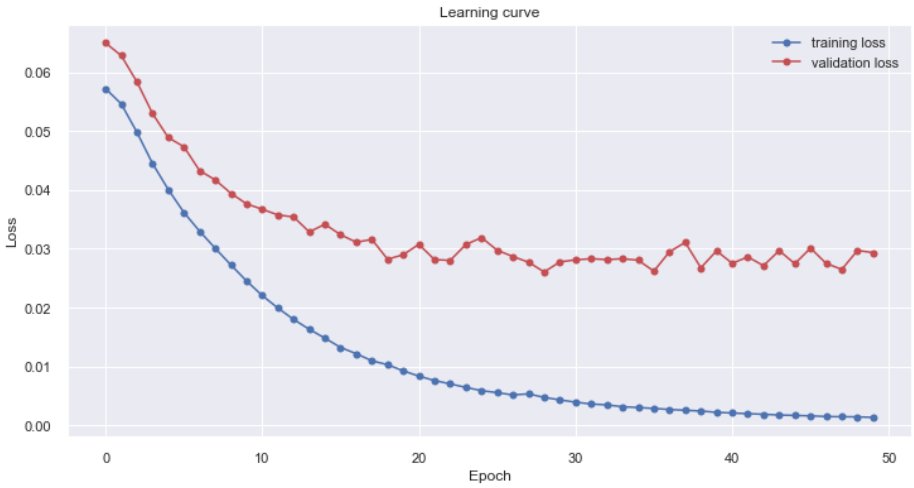


Fig. 6. BI-LSTM learning curve

dependencies are not very distant, they are rather very close. Hence, it is not unusual to claim that Bi-LSTMs may outperform BERT in such cases.

7 Conclusion

This research aims to perform NER on the legal judgments using bidirectional LSTM and BERT models. We used tags specific to legal data in tagging and trained the models. The experiment done on 60 judgments shows that our model improve the efficiency of NER on the legal judgments. Meanwhile, our model perfoms well on unstructured data where it signifies the uniqueness of our model. Furthermore, both the models have different outputs and with different accuracy. The main challenge we experienced was tagging the data because the legal domain differs significantly from the general corpus.

7.1 Future Work

We got the information but now we need to identify how that information is useful or related. For example if we want to find the Advocate of respondent and petitioner we need a relation or if we are able to captures all the sections we can not figure out which is relevant to the particular case. So we can work on this in future.As we have used both BERT and Bi-LSTM and Bi-LSTM works better so further we can add CRF layer in Bi-LSTM model for more efficient result in BIO-tagging.

8 Acknowledgment

Dr. Kshitiz Verma is supported by the Teachers Associateship for Research Excellence (TARE) fellowship (Sanction Number: TAR/2019/000437) provided by the Science and Engineering Research Board (SERB), Department of Science and Technology, India.

References

1. Entity recognizer. https://spacy.io/api/entityrecognizer
2. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science **2**, e93 (2016)
3. Angelidis, I., Chalkidis, I., Koubarakis, M.: Named entity recognition, linking and generation for greek legislation. In: JURIX. pp. 1–10 (2018)
4. Benesty, M.: Ner algo benchmark: spacy, flair, m-bert and camembert on anonymizing french commercial legal cases. https://towardsdatascience.com/benchmark-ner-algorithm-d4ab01b2d4c3 (2019)
5. Chalkidis, I.: Deep neural networks for information mining from legal texts. Ph.D. thesis, Οικονομικό Πανεπιστήμιο Αθηνών. Σχολή Επιστημών και Τεχνολογίας της ... (2021)

6. Chalkidis, I., Androutsopoulos, I.: A deep learning approach to contract element extraction. In: JURIX. pp. 155–164 (2017)
7. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: the muppets straight out of law school. CoRR **abs/2010.02559** (2020), https://arxiv.org/abs/2010.02559
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. Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., Wudali, R.: Named entity recognition and resolution in legal text. In: Semantic Processing of Legal Texts, pp. 27–43. Springer (2010)
10. E. Leitner, G.R., Moreno-Schneider, J.: Fine-grained named entity recognition in legal documents. SEMANTiCS, Lecture Notes in Computer Science, Vol.11702 (2019)
11. Ekbal, A., Bandyopadhyay, S.: Lexical pattern learning from corpus data for named entity recognition. In: Proceedings of International Conference on Natural Language Processing (ICON) (2007)
12. Grishman, R., Borthwick, A.: A maximum entropy approach to named entity recognition (1999)
13. Kim, M.Y., Xu, Y., Goebel, R.: A convolutional neural network in legal question answering. In: JURISIN Workshop (2015)
14. Kim, M.Y., Xu, Y., Lu, Y., Goebel, R.: Legal question answering using paraphrasing and entailment analysis
15. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
16. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering **34**(1), 50–70 (2020)
17. Loza Mencía, E.: Segmentation of legal documents. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law. pp. 88–97 (2009)
18. Loza Mencía, E., Fürnkranz, J.: Efficient multilabel classification algorithms for large-scale problems in the legal domain. In: Semantic Processing of Legal Texts, pp. 192–215. Springer (2010)
19. Nallapati, R., Manning, C.D.: Legal docket classification: Where machine learning stumbles. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 438–446 (2008)
20. Peng, F., Mccallum, A.: Accurate information extraction from research papers using conditional random fields. pp. 329–336 (01 2004)
21. Srihari, R.K.: A hybrid approach for named entity and sub-type tagging. In: Sixth Applied Natural Language Processing Conference. pp. 247–254 (2000)
22. Todorovic, B.T., Rancic, S.R., Markovic, I.M., Mulalic, E.H., Ilic, V.M.: Named entity recognition and classification using context hidden markov model. In: 2008 9th Symposium on Neural Network Applications in Electrical Engineering. pp. 43–46 (2008). https://doi.org/10.1109/NEUREL.2008.4685557
23. Voutilainen, A.: Part-of-speech tagging. The Oxford handbook of computational linguistics pp. 219–232 (2003)
24. Wakao, T., Gaizauskas, R., Wilks, Y.: Evaluation of an algorithm for the recognition and classification of proper names. In: Proceedings of the 16th Conference on Computational Linguistics Volume 1. p. 418–423. COLING '96, Association for Computational Linguistics, USA (1996). https://doi.org/10.3115/992628.992701, https://doi.org/10.3115/992628.992701

A Comparative Study of BERT and Legal-BERT for the Prediction of Indian Legal Case Judgements

Anukriti Bansal[1⋆], Aman Jain[1⋆], Himanshu Goyal[1], and Vikas Bajpai[1]

The LNM Institute of Information Technology, Jaipur, Rajasthan, India
anukriti.bansal@lnmiit.ac.in, amanj3335@gmail.com,
himanshu1234goyal@gmail.com, vikas.bajpai87@gmail.com

Abstract. Transformer-based models have gained immense popularity in the past few years and many state-of-the-art algorithms in natural image processing are based on transformers. BERT(Bidirectional Encoder Representation from Transformer) is also an attention based model which works very well on text classification. In order to use these models in Indian judiciary, incorporating domain knowledge is necessary. Legal-BERT was developed using the domain knowledge of US and EU legal documents. We obtain text embedding from both these models and use them for the prediction of Indian legal case judgements using several machine learning and deep learning algorithms. Both BERT and Legal-BERT have an input size limitation of 512 tokens. In this paper, we propose a solution for the limited token size of BERT models and perform a thorough comparison of BERT and Legal-BERT models to check their effectiveness in the Indian legal system. In all the algorithms, it has been observed that Legal-BERT outperforms basic BERT model and can be used in Indian legal context as well. The source code of the current work is made publicly available at https://github.com/stelios357/Predicting-Indian-Legal-Case-Judgements.

Keywords: BERT · Legal-BERT · Transformer model · Legal Judgement Prediction · Deep Learning

1 Introduction

Prediction of the judgement of legal court cases may help in the understanding of the judicial decision-making process. It can also assist lawyers and judges by suggesting them the possible outcome of any court case. This system will be very useful in countries like India where almost 27 millions cases are pending in courts, many of which are there for more than ten years [26, 29]. One of the reasons for such a huge backlog is very low judges to citizens ratio. In order to speed up the disposal of cases, intelligent and efficient systems for the prediction of outcomes, searching for similar cases from the past, and summarization of legal documents can be very helpful.

⋆ The authors have contributed equally.

Transformer-based language models are currently the state-of-the-art models for several natural language processing (NLP) tasks such as text classification, summarization, translation, question answering, etc [28] [21, 22]. Obtaining the text embedding is one of the most fundamental and significant steps in NLP tasks. BERT (Bidirectional Encoder Representation from Transformer) is very popular and effective context-based embedding model [13]. The embeddings obtained from BERT can be used as input to various machine learning and deep learning-based algorithms for various NLP tasks. The publicly available BERT models have been pre-trained mainly using Wikipedia pages and various books, therefore, they may fail to give good performance when used in specialized domains such as biomedical, clinical, scientific text, or legal [3,8,18,20]. Chalkidis et al [8] introduced a BERT model for legal domain and named it as Legal-BERT. They trained the model from scratch using a different set of hyper-parameters as compared to original BERT proposed by Devlin et al [13], on the EU and US legal text documents. Their model performed better on legal text classification and sequence tagging.

BERT and Legal-BERT both were trained on the datasets which are different from the documents of Indian judicial system. Additionally, both have the input size limitation of 512 tokens, which is very less in case of long legal case documents. To exploit the advantages of already trained language models, in this paper, we provide a unique way to handle the limitation of token size of these models and compare them for the prediction of judgements based on Indian legal documents. There are arguments and pleas in legal documents and the task is to predict the court's decision that whether the plea will be accepted or rejected. In this paper we use bert-base-uncased[1] and legal-bert-uncased[2] models and compare their text-embeddings to predict court's judgement, using various machine learning and deep learning algorithms.

First the complete legal document is tokenized into words and special tokens ([CLS] token at the starting of the text and [SEP] token at the end of each sentence) are added. An important constraint with the BERT and Legal-BERT is that they accept only 512 tokens as input. Malik et al [24] in their work have mentioned that for majority of Indian court files, last 512 tokens are sufficient for the prediction of judgement. We divide the whole legal document in the chunks of 512 tokens and use the last 5 chunks for our experiments. Text embeddings (vector of length 768) of each chunk is obtained as the output of BERT and Legal-BERT. These embeddings are given in different ways to various machine learning and deep learning model to predict the Indian court judgement that whether it will be accepted or rejected. From our experiments, we have observed that although Legal-BERT was trained on EU and US legal documents, but it perform well on Indian legal texts as well. Our research will help in choosing a better model for further research in Indian legal domain for various other NLP tasks as well.

[1] https://huggingface.co/bert-base-uncased
[2] https://huggingface.co/nlpaueb/legal-bert-base-uncased

2 Literature review

Several authors are working on finding solutions for different problems in the legal domain, such as measuring similarity between documents [25], document summarization [4], and legal judgement prediction [5, 32]. Here, we review some of the recent work on legal judgement prediction using classical machine learning and deep learning algorithms

2.1 Machine learning based prediction methods

Shaikh et al [27] have proposed a method to predict the outcome of murder related cases as Acquittal or Conviction. They have manually identified certain keywords and used used classical machine learning algorithms for the prediction. Lage-Freitas et al [19] used TF-IDF and machine learning-based algorithm for the prediction of Brazilian court decisions. In another approach, linear SVM was used by [2] to forecast the violations from facts in Human Rights cases in European Courts.

2.2 Deep learning based prediction methods

Chaphalkar et al [9] used multi-layer perceptron to predict the outcome of construction dispute claims. They determined the intrinsic factors for construction disputes related claims through case study approach and used MLP for the prediction of the outcome. Chalkidis et al [7] evaluated a variety of neural models for judgement prediction and proposed hierarchical BERT to surpass the limitation of token length of BERT. On CAIL2018 [30] dataset, Yang et al. [31] made an attempt to predict similar law articles and relevant information by using multi-perspective bi-feedback network. In another work to predict relevant similar law articles and the charges, Luo et al. [23] proposed an attention based model, which was trained and tested on Criminal Law dataset of China. For predicting the prison term, Chen et al. [10] used deep gating network.

2.3 Major contributions

Following are the major contributions of this paper:

1. We propose various novel ways to represent text encoding from the BERT model that also deals with the issue of limited token length of BERT. As observed in the results, the proposed feature representations give better results as compared to using 512 tokens of BERT and Legal-BERT.
2. A thorough comparison of BERT and Legal BERT is presented using various machine learning and deep learning architectures.
3. The comparison results can be used in further research in the use of transformer models in Indian legal domain.

The organization of the paper is as follows. In Section 3 we explain the proposed feature representation methods and various machine learning and deep learning algorithms that are used to predict court judgments. Details of the dataset and experimental evaluations of various algorithms are given in Section 4. Finally we conclude the paper in Section 5.

3 Proposed method for legal case decision prediction

3.1 Overview

In this work, we propose various novel ways to use BERT (and Legal-BERT) text embeddings as an input to various machine learning and deep learning algorithms for the prediction of legal case decision (whether the case will be accepted or rejected). First, we convert the legal text into a sequence of tokens (words), and this process is called tokenisation. After tokenisation, we add special tokens and send it as input to BERT/Legal-BERT. [CLS] token is added at the starting of the input text and [SEP] token is added at the end of each sentence of the text. BERT model then output an embedding vector of size 768 for each of the tokens. We use the embeddings of [CLS] token to represent the complete text.

As mentioned in Section 1, a major constraint with the BERT is that it only accepts 512 tokens as input. Malik et al [24] claimed that in most of the Indian legal case documents, text embedding of the last 512 tokens is sufficient for the prediction of the court's decision. To see the effect of capturing more information from the legal documents in the performance of machine learning and deep learning-based prediction algorithms, we divide the documents in the chunks of 510 tokens. We add [CLS] and [SEP] tokens at the starting and at the end of each chunk respectively. We use last 5 chunks of the document and use each of them as an input to the BERT. At the end, we have five vectors each of length 768 ([CLS] representations). Since length and the quality of input plays an important role in the performance of any machine learning and deep learning algorithm, we propose three ways to represent the information from these five vectors. These are explained in Section 3.2 Feature representation.

Next we explain various machine learning and deep learning models which are used for the prediction using the text embeddings from BERT and Legal-BERT. In addition to these models, we also use BERT/Legal-BERT as classifier.

3.2 Feature representation

We use four different feature representations for our experiments. In the first case we give only the last 512 tokens of legal case document as an input to BERT (and Legal-BERT), which give us the text embeddings in form of a vector of length 768. This vector acts as an input to different ML and DL algorithms. The features are represented as $bert_{512}$ and $legalBert_{512}$. The details of remaining three representations are given as follows:

Sequential representation We obtain text embeddings from the last 5 chunks (512 tokens each) of legal documents which give us five vectors of length 768 each. In this representation, we concatenate these five vectors to create a vector of length 3840. $bert_{seq}$ and $legalBert_{seq}$ are used to represent sequential features.

Averaging The length is the feature vector obtained after sequential representation is very large. Reducing the feature length may give better performance with machine learning algorithms. Inspired from the global average pooling layer of convolutional neural networks, here we take the average value of five vectors, which results in a vector of length 768. We represent these features as $bert_{avg}$ and $legalBert_{avg}$

Maximum In this case, the idea came from the max-pooling layer of convolutional neural networks. The maximum value at the same index of five vectors is chosen, which results in a 768 sized vector. The features are represented as $bert_{max}$ and $legalBert_{max}$

3.3 Classical machine learning models

The four types of feature representations mentioned in the above section are used as input to the various machine learning algorithms: Adaboost [14], gradient boosting [6, 15], logistic regression, random forest classifier [17], and support vector machine (SVM) [12].

3.4 Deep learning models

Classical deep learning models We experiment with two different deep learning models for the prediction of the judgement of Indian legal case documents. The first one is a multi-layer perceptron model consisting of five fully connected layers. Intermediate dropout layers are used to prevent the over-fitting. Another model we use for legal judgement prediction is the classifier that come along with the pre-trained BERT and Legal-BERT.

Sequential deep learning models The classical models mentioned above do not capture the sequence information of these five vectors or chunks ([CLS] representations of length 768 each). To preserve the sequence information, these vectors are used as input to sequence-based models. In this paper we have experimented with LSTM, GRU, Bidirectional LSTM (BiLSTM), and Bidirectional GRU (BiGRU). The network architecture for all these models is the same and consists of three LSTM/BiLSTM/GRU/BiGRU cells of sizes 100, 50, 20 respectively. To prevent over-fitting, we use intermediate dropout layers at the rate of 0.25, 0.20, and 0.1. We trained the models for with early stopping criteria and batch size 16.

4 Experimental results and discussion

All the experimental programs are coded using Keras [11] API of TensorFlow framework [1,16] . The hardware setup includes Nvidia DGX-1 server with Dual 20-core intel Xeon e5-2698 v4 2.2 Ghz processor, supported by 512 GB DDR4 RAM and 256 GB (8x32) GPU memory.

4.1 Dataset description

Our dataset consists of legal case files of Supreme Court of India that were scrapped from the website of indiankanoon.org [3]. Each case file is the entire case preceding of the Supreme Court of India in the English language. The data has been scrapped for the period 1947 to April 2020. These documents are very verbose and have a lot of grammatical and spelling errors since they are typed when a preceding is going on. Hence the pre-processing and the cleaning is done accordingly. Given a case, the appeal can either be accepted or rejected. The final decision of whether the plea has been accepted or rejected is found towards the end of the case preceding. This information about the final decision has been extracted and used to label the data: 1 for accepted and 0 for rejected. In certain case documents, there can be multiple pleas. In such scenario, if there is a single decision for all the pleas, the case file is annotated accordingly, otherwise, even if a single plea is accepted, we label the document as accepted. There are 42409 legal cases files in our dataset. Out of a total of 42409 case proceedings, the plea has been accepted in 17838 cases which makes 42.06% of the dataset, and in 24571 cases, the plea has been rejected, which makes 57.93% of the dataset. The cases in which the decision was neutral or which had no decisions are ignored and are not part of the dataset. To design algorithms for the prediction of court case judgement, i.e., whether the plea will be accepted or rejected, we split the dataset in training, validation, and test in the ratio 70:18:12.

4.2 Evaluation metrics

To evaluate the overall performance of different models in the prediction of legal case judgement, we use binary cross entropy as the loss function and F1 score as the basic evaluation metric.

Binary cross entropy The loss function is given as follows:

$$Log\ loss = \frac{1}{N}\sum_{i=1}^{N} -(y_i * log(p_i) + (1 - y_i) * log(1 - p_i)) \tag{1}$$

Here, p_i is the probability of class 1, and $(1 - p_i)$ is the probability of class 0.

[3] https://indiankanoon.org/

F1 score/F measure This measure helps in determining the correctness of the prediction. It requires the computation of precision and recall metrics. The precision (P) suggests how well the model is able to correctly predict the judgements (positive or negative). Recall (R), on the other hand determines how many predictions we correctly made while penalizing for the incorrect predictions. F1 score is computed as follows:

$$F1 = \frac{2.R.P}{R+P} \tag{2}$$

4.3 Results and Discussion

Prediction results using classical machine learning models As mentioned in Section 3.3, we have experimented with five classical machine learning models. We have four different types of feature representations (Section 3.2) and embeddings from BERT and Legal-BERT. Therefore, for each machine learning model there are eight results as shown in Table 1. In all the cases, Legal-BERT embeddings clearly outperforms the results using BERT-base embeddings. It can also be observed that a few machine learning models performs well for features obtained using last 512 tokens while others give better results on the proposed feature representations that consider more number of tokens (or chunks).

Prediction results using classical deep learning models The prediction results on different feature representations using MLP and BERT and Legal-BERT as classifiers is shown in Table 2. Here as well Legal-BERT performs better than BERT-Base for both deep learning models. Additionally, it can be observed that as we increase the number of tokens (five chunks of 512 tokens), results improve significantly. All three proposed feature representations (Section 3.2) give better results as compared to features obtained using 512 tokens. This show the applicability of our method in various NLP applications.

Prediction results using sequential deep learning models As mentioned in Section 3.4, we have experimented with four sequential deep learning algorithms. As can be seen from the Table 3, in all four cases text-embeddings obtained from Legal-BERT works better in prediction of judgements for Indian legal documents.

5 Conclusion

In this paper we proposed a solution for input token size limitation of BERT/Legal-BERT models and showed that the proposed method helps in capturing more contextual information and therefore gives better prediction accuracy. This can be used for various other NLP applications as well. BERT and Legal-BERT both were trained on the documents that are different from Indian legal texts.

Table 1. Prediction results using classical machine learning models

Feature Representation	Train set				Test set			
	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)
Support Vector Machine								
$bert_{512}$	66	66	65	65	63	63	63	63
$bert_{seq}$	71	72	70	70.37	55	63	59	62.17
$bert_{avg}$	63	63	63	63.14	61	62	61	61.51
$bert_{max}$	65	65	65	64.80	62	62	62	62.04
$\boldsymbol{legalBert_{512}}$	**72**	**72**	**72**	**72.22**	**67**	**68**	**67**	**67.75**
$legalBert_{seq}$	73	73	73	72.7	64	64	64	63.44
$legalBert_{avg}$	67	67	67	67.07	66	66	66	66.25
$legalBert_{max}$	70	71	70	70.55	66	66	66	66.22
Logistic Regression								
$bert_{512}$	63	63	63	63.16	62	62	62	62.26
$bert_{seq}$	68	68	68	67.79	64	64	64	63.83
$bert_{avg}$	63	63	62	62.74	61	61	61	61.18
$bert_{max}$	61	62	61	61.60	61	61	60	60.62
$legalBert_{512}$	67	68	67	67.89	61	61	60	60.62
$\boldsymbol{legalBert_{seq}}$	**74**	**74**	**74**	**74.08**	**68**	**69**	**68**	**68.74**
$legalBert_{avg}$	67	67	67	67.39	66	66	66	66.07
$legalBert_{max}$	64	65	64	64.74	64	64	64	64.06
Random Forest								
$bert_{512}$	66	66	65	65.94	60	60	60	60.54
$bert_{seq}$	62	60	57	57.55	60	59	55	55.88
$bert_{avg}$	64	64	64	63.87	58	59	58	58.35
$bert_{max}$	64	65	64	64.77	59	59	59	59.09
$\boldsymbol{legalBert_{512}}$	**66**	**66**	**66**	**66.60**	**61**	**61**	**61**	**62.35**
$legalBert_{seq}$	62	62	60	59.85	61	60	58	58.07
$legalBert_{avg}$	64	65	64	64.63	60	61	60	60.74
$legalBert_{max}$	65	65	65	65.19	60	61	60	60.62
Adaboost								
$bert_{512}$	99	99	99	99	77	69	69	73.11
$bert_{seq}$	99	99	99	99	74	70	70	72.18
$bert_{avg}$	99	99	99	99	73	69	69	71.62
$bert_{max}$	99	99	99	99	74	69	69	71.80
$\boldsymbol{legalBert_{512}}$	**99**	**99**	**99**	**99**	**79**	**71**	**72**	**75.04**
$legalBert_{seq}$	99	99	99	99	75	71	71	73.35
$legalBert_{avg}$	99	99	99	99	75	70	71	73.12
$legalBert_{max}$	99	99	99	99	75	71	71	73.20
Gradient Boost								
$bert_{512}$	96	96	96	96.22	71	71	71	71.88
$bert_{seq}$	97	97	97	96.91	71	71	71	71.44
$bert_{avg}$	98	96	96	96.31	70	70	70	70.83
$bert_{max}$	97	97	97	97.25	70	70	70	70.35
$legalBert_{512}$	96	97	96	76.56	73	73	73	73.82
$legalBert_{seq}$	98	98	98	97.88	73	73	73	73.33
$\boldsymbol{legalBert_{avg}}$	**97**	**97**	**97**	**96.89**	**74**	**74**	**74**	**74.22**
$legalBert_{max}$	97	97	97	97.31	72	72	72	72.94

Table 2. Prediction results using classical deep learning models

Feature Representation	**Train set**				**Test set**			
	Precision (%)	**Recall (%)**	**F1 (%)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Accuracy (%)**
Multilayer Perceptron (MLP)								
$bert_{512}$	60	64.1	62	62.39	55.4	66.1	60	59.96
$bert_{seq}$	70.7	71.3	71	70.59	69.9	62.2	64	63.99
$bert_{avg}$	62.7	63.3	63	63.42	60.3	62.1	61	61.12
$bert_{max}$	66.4	61.7	64	64.48	60.9	64.3	62	61.74
$legalBert_{512}$	64.6	54	57	57.39	59.9	54.9	57	57.35
$\boldsymbol{legalBert_{seq}}$	**95**	**97**	**96**	**96**	**71.3**	**73.3**	**72**	**71.87**
$legalBert_{avg}$	81.1	80.9	81	80.79	78.4	64.7	70	69.6
$legalBert_{max}$	75.7	72.3	74	74.01	67.6	64.6	66	65.99
BERT and Legal-BERT as classifier								
$bert_{512}$	88.8	91	90	89.9	64.9	59.1	62	61.9
$bert_{seq}$	92.5	94.1	93	93.3	74.4	69.9	72	72.1
$bert_{avg}$	88.9	91.3	90	90.1	63.8	66.4	65	65.1
$bert_{max}$	91.9	90.5	91	91.2	66	66.4	66	66.2
$legalBert_{512}$	92.3	95.9	94	94.1	69.2	65.3	67	67.2
$\boldsymbol{legalBert_{seq}}$	**97.7**	**95.5**	**97**	**96.6**	**76.7**	**81**	**79**	**78.8**
$legalBert_{avg}$	96.5	94.7	96	95.6	72.4	74.8	74	73.6
$legalBert_{max}$	97.5	94.1	96	95.8	73.7	74.7	74	74.2

Table 3. Prediction results using sequential deep learning models

Model	Text Embeddings	Train Accuracy (%)	Train Recall (%)	Train Precision (%)	Train F1 (%)	Test Accuracy (%)	Test Recall (%)	Test Precision (%)	Test F1 (%)
GRU	BERT	74.4	71.5	69.9	70.7	64.9	59.4	59.4	59.4
GRU	Legal-BERT	93.9	93.3	92.6	92.9	72.4	67.9	68.0	68.0
BiGRU	BERT	83.7	80.3	81.5	80.9	66.2	59.0	61.2	60.1
BiGRU	**Legal-BERT**	**95.9**	**92.5**	**98.0**	**95.1**	**72.8**	**62.5**	**71.0**	**71.0**
LSTM	BERT	71.8	55.6	72.5	62.9	65.7	47.9	63.6	54.6
LSTM	Legal-BERT	76.9	63.8	78.7		70.4	53.9	70.5	61.1
BiLSTM	BERT	70.6	58.3	68.8	63.1	66.1	62	60.4	61.3
BiLSTM	Legal-BERT	81.4	80.7	77.2	78.9	69.9	57.6	67.7	62.3

There is uncertainty in using Legal-BERT in Indian context by some of the researchers [24]. We presented a thorough comparison of BERT and Legal-BERT for the prediction of Indian court judgments using various machine learning and deep learning algorithms. In all the scenarios Legal-BERT clearly outperformed the basic BERT model. This establishes the applicability of Legal-BERT in Indian context as well. In future, we plan to pre-train and fine-tune transformer

model using Indian legal documents. Other transformer-based architectures and complete document embeddings will be explored as a future research direction.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). pp. 265–283 (2016)
2. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science **2**, e93 (2016)
3. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019)
4. Bhattacharya, P., Hiware, K., Rajgaria, S., Pochhi, N., Ghosh, K., Ghosh, S.: A comparative study of summarization algorithms applied to legal case judgments. In: European Conference on Information Retrieval. pp. 413–428. Springer (2019)
5. Bhilare, P., Parab, N., Soni, N., Thakur, B.: Predicting outcome of judicial cases and analysis using machine learning. International Research Journal of Engineering and Technology **6**(3), 326–330 (2019)
6. Breiman, L.: Arcing the edge. Tech. rep., Technical Report 486, Statistics Department, University of California at ... (1997)
7. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. arXiv preprint arXiv:1906.02059 (2019)
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Legal-bert: The muppets straight out of law school. arXiv preprint arXiv:2010.02559 (2020)
9. Chaphalkar, N., Iyer, K., Patil, S.K.: Prediction of outcome of construction dispute claims using multilayer perceptron neural network model. International Journal of Project Management **33**(8), 1827–1835 (2015)
10. Chen, H., Cai, D., Dai, W., Dai, Z., Ding, Y.: Charge-based prison term prediction with deep gating network. arXiv preprint arXiv:1908.11521 (2019)
11. Chollet, F., et al.: Keras (2015), `https://github.com/fchollet/keras`
12. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
13. 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)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)
15. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
16. Gulli, A., Pal, S.: Deep learning with Keras. Packt Publishing Ltd (2017)
17. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
18. Huang, K., Altosaar, J., Ranganath, R.: Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342 (2019)

19. Lage-Freitas, A., Allende-Cid, H., Santana, O., Oliveira-Lage, L.: Predicting brazilian court decisions. PeerJ Computer Science **8**, e904 (2022)
20. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
21. Liu, Y.: Fine-tune bert for extractive summarization. arXiv preprint arXiv:1903.10318 (2019)
22. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345 (2019)
23. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. arXiv preprint arXiv:1707.09168 (2017)
24. Malik, V., Sanjay, R., Nigam, S.K., Ghosh, K., Guha, S.K., Bhattacharya, A., Modi, A.: Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. arXiv preprint arXiv:2105.13562 (2021)
25. Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., Ghosh, S.: Measuring similarity among legal court case documents. In: Proceedings of the 10th annual ACM India compute conference. pp. 1–9 (2017)
26. Prakash, S.B.N.: E judiciary: A step towards modernization in indian legal system. Journal of Education & Social Policy **1**(1), 111–124 (2014)
27. Shaikh, R.A., Sahu, T.P., Anand, V.: Predicting outcomes of legal cases based on legal factors using classifiers. Procedia Computer Science **167**, 2393–2402 (2020)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Verma, K.: e-courts project: A giant leap by indian judiciary. Journal of Open Access to Law **6**(1) (2018)
30. Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al.: Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)
31. Yang, W., Jia, W., Zhou, X., Luo, Y.: Legal judgment prediction via multi-perspective bi-feedback network. arXiv preprint arXiv:1905.03969 (2019)
32. Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., Sun, M.: Iteratively questioning and answering for interpretable legal judgment prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 1250–1257 (2020)

Estimating Time to Clear Pendency of Cases in High Courts in India using Linear Regression

Kshitiz Verma[1], Anshu Musaddi[2], Ansh Mittal[2], and Anshul Jain[2]

[1] JK Lakshmipat University, Jaipur, India
[2] The LNM Institute of Information Technology, Jaipur, India.
kshitiz.verma@jklu.edu.in
{17ucs185,17ucs028,17ucs029}@lnmiit.ac.in

Abstract. Indian Judiciary is suffering from burden of millions of cases that are lying pending in its courts at all the levels. The High Court National Judicial Data Grid (HC-NJDG) indexes all the cases pending in the high courts and publishes the data publicly. In this paper, we analyze the data that we have collected from the HC-NJDG portal on 229 randomly chosen days between August 31, 2017 to March 22, 2020, including these dates. Thus, the data analyzed in the paper spans a period of more than two and a half years. We show that:

- the pending cases in most of the high courts is increasing linearly with time.
- the case load on judges in various high courts is very unevenly distributed, making judges of some high courts hundred times more loaded than others.
- for some high courts it may take even a hundred years to clear the pendency cases if proper measures are not taken.

We also suggest some policy changes that may help clear the pendency within a fixed time of either five or fifteen years.

Keywords: Indian Judiciary, Pending Cases, NJDG, Linear Regression

1 Introduction

Justice delayed is justice denied. Delays in a justice system don't just violate the fundamental, constitutional and human rights of a victim but they also have an adverse effects on the rights of the accused as well as those who are convicted. Recently, Supreme Court of India acquitted two persons, accused of a gang rape after 28 years of the incident [1]. In an another case, a litigant had to fight for more than four decades to retrieve possession [26]. Such unfortunate examples are not exceptions in India. They contribute towards the disrepute of the judicial system, lower the faith of the people in judiciary and also impact the economic growth. The judicial delays may potentially translate to a loss of 0.48% of the national Domestic Gross Product (GDP) of India [24]. Hence, the government has all reasons to implement policies that help removing the pendency. We define *pendency* and *delay* as they are defined in The Report No. 245 of Law Commission of India [3]:

1. *Pendency: All cases instituted but not disposed of, regardless of when the case was instituted.*

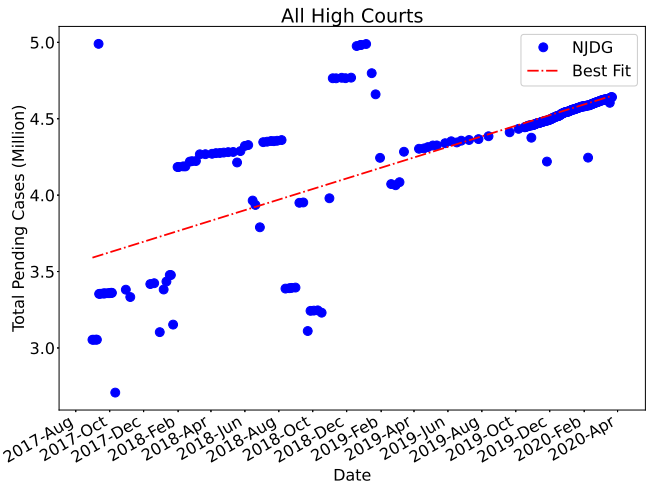


Fig. 1: Evolution of pending cases in all the High Courts of India.

2. *Delay: A case that has been in the court/judicial system for longer than the normal time that it should take for a case of that type to be disposed of.*
3. Nullifying pendency: To make pendency equal to zero.

Note that in our definition of pendency, even if a case was instituted just one day ago, it still counts as pendency. Hence, it is a very strict way of computing judicial backlogs. We will use this meaning of pendency throughout the paper.

Many attempts have been made to estimate the number of years required to clear the pendency of the cases. According to Justice V. V. Rao, a prediction made in 2010 [8], it would take 320 years to clear the pendency. A report of Delhi High Court stated that it would require 466 years to clear all the pendency in Delhi Courts [9]. On the other hand, a commitment to clear all the pending cases in five years was made by the then Union Law Minister in 2011 [15] and later by the then CJI Justice H.L. Dattu in 2015 [17]. Yet another commitment was made to make average disposal of cases as three years rather than then 15 years [23]. Law Commission of India published a report discussing delays and pendency in July 2014 [3]. However, they primarily addressed the issue in lower judiciary. Moreover, the statistics and the situation has changed drastically in last six years as even though the number of judges have increased, the pendency has not decreased. National Judicial Data Grid for High Courts (HC-NJDG) [20], provides data on pending cases in High Courts and was visioned to be a game changer [21].

In this paper, we exclusively study data from HC-NJDG and report results related to High Courts only. As of October 2020, there were more than 5 million cases pending in the High Courts of India [27]. We have chosen to study pendency in high courts only because they are generally well equipped with resources to implement the suggested measures effectively. Also, since the number of high courts is only 25 (the 25^{th} started on January 1, 2019), analysis and results are easy to interpret.

This study, to the best of our knowledge, is the first one to analyze NJDG data over such a long period of time. Most of the existing studies consider the data from only one day on NJDG. Hence, we differ from the other studies in this basic premises itself. We also also try to answer the question, "How reliable is single day analysis of NJDG?". We answer this as negative, i.e., the NJDG data collected on just one day may not be taken as reliable for any reasonable analysis. There have been instances when the data on NJDG was very erroneous and such days are not rare. For example, a recent article on the pendency statistics of Bombay High Court claimed that 4.64 lakh cases are pending [13]. Our finding is that throughout the data collection period, the Bombay High Court has updated the number of pending cases only once.

1.1 Results in the paper

Our work revolves around answering one central question. *How long will it take to clear all the pending cases in the High Courts?* We summarize some of our results below:

1. Increasing pendency: Pending cases are increasing in the high courts of India (Fig. 1). Hence, in the absence of clear policies and their implementation, pendency can never be cleared in most of the high courts.
2. Load on judges in different high courts: There is a huge difference in terms of average load of cases on judges of different high courts. E.g., a judge of Rajasthan High Court has almost 250 times more load than a judge in Sikkim High Court.
3. Years to clear the pendency: Assuming a linear increase in the number of cases as well as in the number of judges so that the high courts operate at their sanctioned strength of judges, then for most of the high courts pendency can be nullified (Fig. 5, 6). However, the number of years required to do so may vary significantly.
4. Proposed policies for clearing the pendency: Fig. 7 and Fig. 8 may help the government of India to take informed decisions on the number of judges that need to be increased. We find that the current sanctioned strength of the high courts is adequate to take care of the newly instituted cases. They are, however, insufficient for clearing the previous backlog of cases. Hence, the government may enact legislation to create some temporary positions in high courts, just to clear the pendency, without changing the actual current sanctioned strength.

1.2 Organization of the paper

The rest of the paper is organized as follows: Section 2 encompasses the scope of the work and the related studies. Section 3 discusses data collection and explains the graphs used in the paper. Section 4 is home to the most important result of the paper in which we estimate the time required to nullify the pendency in the high courts. Section 5 focuses on forming policies to ensure that the pendency decreases. Section 6 concludes the paper.

2 Scope and Related Work

Before we proceed any further to discuss the technical findings of the paper, we first understand the scope and the limitations of our work. There have been many news

reports, articles and studies on pendency in Indian courts [4] [18] [22] [30]. In this paper, instead of studying the pending cases in all the courts of India, we decided to limit ourselves to only the high courts. This limits the number of court complexes in our study to just 39 without decreasing the complexity of the problem as the high courts are more clogged with cases than subordinate courts. Moreover, the improvements suggested in this paper are relatively easier to implement in high courts than in lower courts because of better infrastructure and budget available. Thus, we would concretely know where things can be improved. Hence, throughout this paper, we have concentrated on the pendency in high courts rather than the subordinate courts.

In our study, we have not taken input from any real person. No judge, advocate, litigant or court staff was interviewed. This may have both positive and negative impact. Intervention of court staff and those who are involved in updating NJDG may have provided more insights to interpret our results. On the other side of it, their views might have biased our results. So we decided to leave it for future because we wanted our assessment to be purely technical and statistical based only on the observations made from the data that we have collected from NJDG.

A study by Alok Prasanna Kumar has used number of the District and Magistrate courts, collected from the National Judicial Data Grid as of 18 March, 2016 [28]. There are studies advocating to decrease the holidays available in judiciary [11]. There are studies conducted by the Department of Justice as well [10]. While this study is very comprehensive, the results reported are different from ours and the parameters considered for evaluation are different as well. Various studies including [29], have conducted research on e-Court policies. The importance of data analysis of judicial data and the role of computer science is also suggested in [5]. The Department of Justice also encourages research conducted on judicial reforms by means of funding [12]. Another rich source of information on pending cases are the annual reports published by the Supreme Court of India [25].

The most relevant work that can be compared with our work is Law Commission of India report of July 2014 [3]. The availability of data was a major concern for the authors of the report. They studied data on pendency at the end of years from 2002 to 2012. However, the primary focus of their analysis is the courts that are subordinate to the jurisdiction of high courts. Moreover, the pendency figures have more than doubled now compared to 2012. Hence an understanding of the rate of increase of cases is crucial in studying pendency. We make a clear distinction from the report by exclusively studing the high court data, collecting data over 229 days, and counting the contribution of each and every data point by using linear regression for the analysis.

Other relevant related work in this area is the Daksh report on the state of the Indian Judiciary [24]. Their approach, however, is very different from ours. They have conducted a ground level research by surveying and obtaining the first hand experience of the litigants and other stake holders. Our work, on the other hand, relies completely on the data provided by the National Judicial Data Grid for High Courts (HC-NJDG).

Hence, a lot of studies agree that the judicial throughput has to be increased. Either the number of vacations may be reduced or the number of judges may be increased.

The issue has been of utmost importance to all the Chief Justices of India [7] [6]. Hence, a lot of research needs to be conducted in the area so as to help solve a crucial problem faced by Indian Judiciary.

3 HC-NJDG Data Collection

High Court National Judicial Data Grid (HC-NJDG) was launched in July 2017 [20,21]. We started collecting data from the portal on August 31, 2017. The last data used in this paper was collected on March 22, 2020. We have collected data on 229 randomly chosen dates, spanning a period of more than two and a half years.

To provide a glimpse of the data, some of the statistics, as collected on August 20, 2018, are provided in Table 1. The portal has more statistics available but we have chosen to present only the ones that are relevant for this paper. The data related to the number of pending cases in all the high courts in India is shown. The table also presents data on the number of monthly disposed and filed cases.

Title	Civil	Criminal	Writs	Total
Pending Cases	1506780	769754	1114448	3390982
Cased Filed (monthly)	27663	42404	32009	102063
Cased Disposed (monthly)	28080	47368	36548	111996

Table 1: Some of the statistics available on the HC-NJDG portal [20].

Note that the last day for data collection for this paper was March 22, 2020, just before the nation-wide lockdown was announced in India due to Covid-19 pandemic. We hypothesize that data pre-lockdown and during the lockdown would be very different and hence not comparable for our work.

We have plotted many graphs in the paper. In order to maintain coherence and simplicity, and to have a reach to wider audience, we have restricted ourselves to only two kinds of graphs, as explained below.

Temporal data graphs (Dates on horizontal axis) The horizontal axis (also referred to as X-axis in the paper), consists of dates beginning August 31, 2017 to March 22, 2020 from left to right. The title of each graph is present on the top stating the name of the high court that plot corresponds to. If on some date data was not collected, then data is not shown against that date but the date is still present on the X-axis in the all cases. Fig. 1 is an example of this kind of graph.

Spatial data graphs (High Courts on horizontal axis) In these graphs, the data from the high courts is plotted. The horizontal axis, or X-axis, in these graphs have 24 points, each representing one of the 24 high courts. Y-axis plots the value of the considered parameter. If a high court does not have a valid data for that parameter, its name still appears on the X-axis but have no value on the Y-axis. The title of the graph is present at the top. Fig. 2 is an example of this kind of graph.

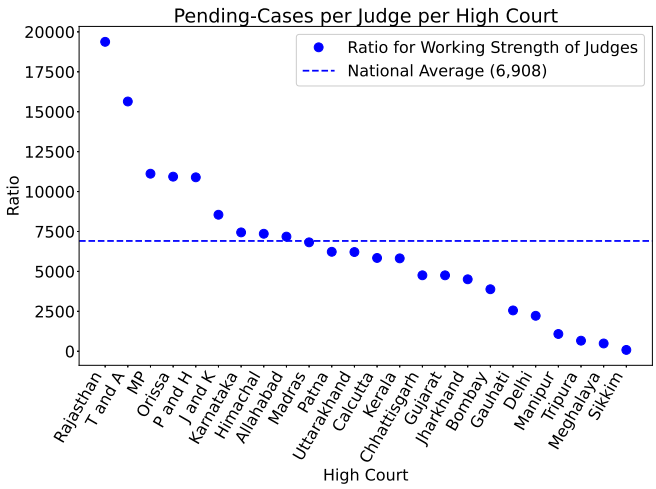


Fig. 2: Ratio of pending cases to the judges in High Courts, i.e., the average number of pending cases per judge per high court.

3.1 Ratio of Pendency to Judges

The total pendency, in itself, does not provide any information until the number of judges in the respective high court is also taken into account. This subsection considers the ratio of *pending cases/number of judges* as a parameter for each high court.

Fig. 2 plots the ratio *pending cases/number of judges* for each high court. The number of pending cases is calculated by taking the pendency on the last day of data collected. The results are plotted in the descending order of the ratio so calculated. This graph provides the distribution of workload on each high court and judges thereof. The blue dots show the ratio *pending cases/working strength of judges* for the average working strength of each high court. For example, Rajasthan High Court has the maximum value of 19,374 pending cases for each judge whereas Sikkim High Court has the minimum ratio which is 78. Hence, statistically we can say that a judge in Rajasthan High Court has almost 250 times more load than a judge in Sikkim High Court. It can be seen that the situation is similar for most of the high courts. The mean of this ratio is 6908, i.e., the national average of the number of pending cases per judge. It signifies that on an average each sitting judge of the high courts in India needs to dispose 6908 cases to reduce pendency to zero, provided no more cases are filed. It also means that the judges in some of the high courts are insanely overburdened. In the interest of justice, urgent appointments are required so that the case load may be shared. Hence, the number of pending cases per judge is huge and the numbers are so high that it would not be unfair to state that they are simply beyond the capacity of the current number of working judges.

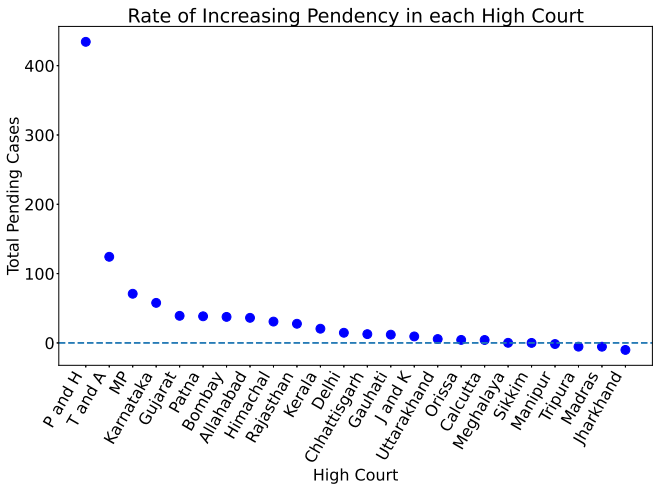


Fig. 3: Average rate of daily increase in the number of pending cases for each high court.

4 Estimating Time Required to Combat Pendency

Having discussed the trend of pending cases and the number of cases per judges in the high courts, we turn our attention to computing the time required to clear the pendency of the cases in high courts of India. Our attempt is the first – to the best of our knowledge – to be based on extremely rich statistical data to answer the question, "How long will it take to reduce the pendency in the high courts to zero?".

4.1 Rate of increase of pendency

Fig. 1 presents increase in the number of total pending cases from August 31, 2017 to March 22, 2020.

We observe that the number of pending cases in the high courts in India is increasing at a rate of approximately 1135 cases per day. It is basically the slope of the best fit line in Fig. 1. We have computed a similar best fit line for all the high courts. The slope for various high courts is taken as rate of increase of pendency. For most of the high courts it means that the pendency will never get over rather increase with time. Fig. 3 shows the rate of increasing pendency for each high court. It is quite expected that Rajasthan High Court, whose ratio of pending cases to judges is very high, has the highest rate of increase of pendency. A similar observation may be made for several other high courts whose ratio of pending cases to judges is very high.

4.2 Towards Computing Time to Combat Pendency

We use our analysis of NJDG data to find out answers to the following questions:

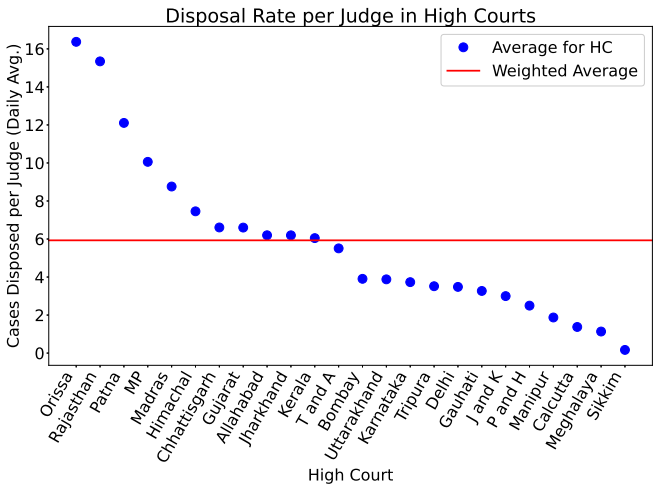


Fig. 4: Average case disposed per day per judge in High Courts. The average number of cases disposed across all the high courts is 5.93, i.e., around 6.

1. What is the rate of disposal of cases per day per judge in high courts? (Fig. 4)
2. If the number of judges in high courts increase linearly and reach their sanctioned strength in ten or twenty years from now, and the average disposal rate used for a judge is as provided in Fig. 4, then how many years are required to reduce the pendency of cases to zero? (Fig. 5)

Disposal related statistics are provided on NJDG portal on a monthly basis. Thus, we have divided the number by 30 to get the daily figure. In Fig. 4, we plot the number of cases disposed per judge per day for each high court. The national average is 5.93. This figure provides the average number of cases disposed by each high court judge in a day. We use these results to estimate the time required to nullify the pendency in different high courts in India.

We have enough information to compute the time required to nullify the pendency in high courts. We are assuming that the number of judges increase linearly every year. We define the following variables:

1. Assumed to be constant, disposal rate per judge per year, denoted as d, of a high court (extrapolated from Fig. 4),
2. Pendency p_t at the start of any given year t in a high court,
3. Working strength w_t of a high court during any given year t,
4. Yearly rate of increase (r_t) of pendency for a high court when average working strength is w_t (extrapolated from Fig. 3),

Then the following holds:

$$p_t = p_{t-1} + r_{t-1} \tag{1}$$

$$r_t = r_{t-1} - d \cdot (w_t - w_{t-1}) \tag{2}$$

where p_0 and r_0 are taken as the values of pendency on the last day of our data collection and from Fig. 3 respectively. The later values are updated according to Eq. 1 and 2 to compute t for which $p_t \leq 0$.

Fig. 5 shows the number of years required to nullify pendency in the high courts. It presents results assuming that the sanctioned strength of high courts are reached in ten years and twenty years respectively. We also assume that the rate of increase of the judges in both the cases is linear. We can see that there is a huge gap between the years taken to clear the pendency in the two cases. If we assume that the vacancy of judges in the high courts is to be filled in twenty years, then Himachal High Court and Madras High Court may take 150 and 113 years respectively. However, if we assume that the working strength of the high courts reach their sanctioned strength in ten years then the numbers for both the high courts mentioned above are 102 and 83 respectively, i.e., an improvement of 48 and 30 years respectively. On the other side of the spectrum we see Tripura High Court and Sikkim High Court that will take 2 years and 6 years respectively, irrespective of whether it takes ten or twenty years to fill the vacancy in these high courts. Thanks to the number of the pending cases, rate of decrease of pendency and the sufficient number of judges to handle that. Another extreme case is the Punjab and Haryana High Court. There is no plot against that high court in either case because the sanctioned strength, no matter whether reached in ten or twenty years, the rate of increase of pendency will still be positive rather than negative. We also see that the majority of the high courts will take more than twenty years if the sanctioned strength is reached in twenty years and more than 14 years if the sanctioned strength is reached in ten years. Hence, if only ten years are taken to fill the vacancy in high courts then substantially lesser number of years are required to clear the pendency. This is also reflected in the average number of years taken. For ten years to fill vacancy, on an average, it will take 25.3 years and for twenty years to fill vacancy, on an average it will take 35.35 years to clear the pendency. Hence, filling the vacancies in the high courts is a key to clearing the pendency.

5 Policy to Improve Access to Justice

In this section, we comment on some of the required fundamental changes in various customs and enactment of laws to increase the number of judges in the high courts. In order to reduce pendency in the future, many policy level changes have been proposed by various studies. We focus on the number of judges required in the high courts. We advocate filling of the vacancy of judges in the high courts [2] [16] and while filling the vacancies, age should be considered [19]. Young judges should be elevated so that the retirement rate of the judges in a high court decrease. We:

1. argue the impact of increasing the efficiency of judges on pendency in high courts Fig. 6. This may be increased by providing more staff and ICT infrastructure.
2. reason for the proposed sanctioned number of judges in high courts depending on the targets set to reduce pendency to zero.

Fig. 6 presents the number of years to nullify pendency if the disposal rate of those high courts is increased to 5.93 for which it is lesser than that. In other words, we

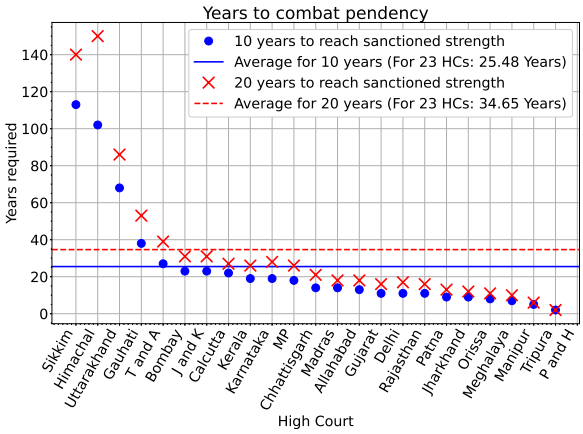


Fig. 5: Years required to nullify pendency if the working strength of each high court is assumed to reach its sanctioned strength in ten and twenty years. The rate of disposal of cases per day per judge for each high court is taken as reported in Fig. 4.

hypothetically increase the number of cases disposed per judge per day to 5.93, if it is lesser than that, unchanged otherwise. This figure represents the number of years corresponding to a very ambitious case in which we assume the minimum disposal of cases per high court judge per day. We see that there is a significant improvement in the average number of years required to nullify the pendency. The average has come down from 25.3 years to 20.61 years if it takes ten years to reach to the sanctioned strength and from 35.35 years to 29.48 years if it takes twenty years to reach the sanctioned strength. In this work, we have not tried to estimate the optimal number of cases a judge may dispose on an average in a day, without compromising on the quality of justice. However, if we are given such a number then we are in a position to deduce its impact on the pendency of cases in the high courts.

Fig. 7 shows the number of judges required if the pendency in high courts is to be nullified in five or fifteen years. We also assume that the number of judges as well as the pendency increase linearly in high courts and that the proposed sanctioned strength reaches in five or fifteen years. To provide a comparison, we have also plotted the current working strength of judges in the high court. Thus, we insist that the only way to substantially reduce the pendency in the high courts is to increase the number of judges. All the other factors like introduction of technology, etc will have a role to play but the scarcity of the judges and supporting staff is the primary reason for pendency. The numbers are very high compared to the current working strength of the high courts and it is in sharp contrast with the fact that the number of judges have not changed much during the data collection period. The rate of appointment of judges is roughly canceled by the rate of retirement, leaving the average number of judges unchanged in the high courts. Elevation of younger judges may help solve this problem. Since the number of judges required is very high compared to the current working strength of judges, the

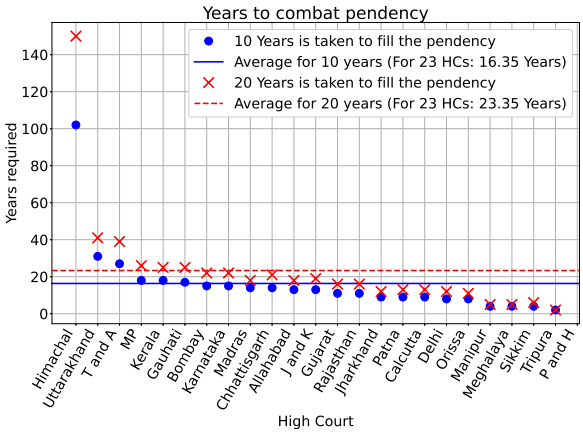


Fig. 6: Years required if the minimum rate of disposal of cases per day per judge is fixed at a minimum of 5.93 in each high court.

whole purpose of this graph is to provide an estimate on how aggressively the judges should be elevated to the high courts. However, such large number of judges may not be required once the pendency is cleared.

Fig. 8 provides an insight on the number of judges required if the rate of increase of pendency is to be made zero, i.e., the pending number of cases should neither increase nor decrease. This provides a good sign for most of the high courts as the number of judges required to make the rate of increase equal to zero is less than the vacancy in that particular high court according to the current sanctioned strength. Note that only Punjab and Haryana High Court has the required number greater than the vacancy. This means that once the pendency is taken care of, the current sanctioned strength is capable of handling the volume of fresh cases that are instituted in the high courts. A similar finding is also reported in [14].

From the above discussion, we can see that the the government will have to be a bit innovative to be able to aggressively clear the pendency. So without increasing the sanctioned strength of the permanent judges in the high courts, government may, through appropriate legislation, increase the number of judges in high courts by elevating judges purely for clearing the pendency. Once pendency is cleared, the current sanctioned strength of the high courts is sufficient to take care of the newly instituted cases.

A due analysis of the cost and the infrastructure required has to be done which is beyond the scope of the current paper.

6 Conclusion

The problem of pending cases in India has taken an unimaginable form. In high courts alone, close to 4.5 million cases are pending of which around 20% are pending for more

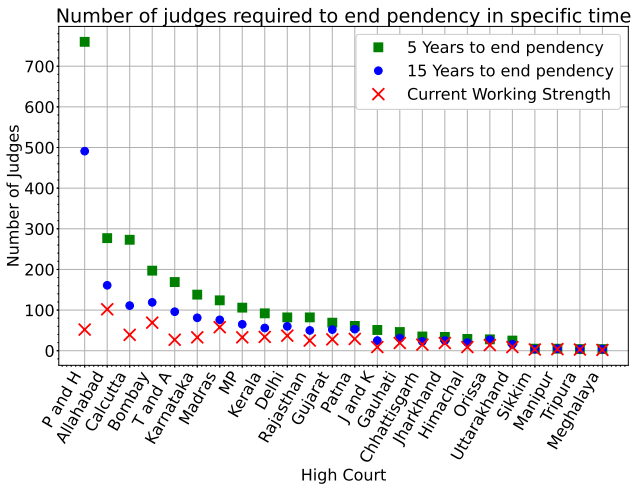


Fig. 7: Number of judges required in each high court to clear the pending cases in five and fifteen years respectively.

than 10 years. We use the data collected from HC-NJDG portal for a period of more than two and a half years to study the trends in the pendency in the high courts. We realize that the pending cases are increasing for almost every high court. We use linear regression to capture the rate of increase of pendency in the high courts. We also make use of the data on disposed cases from the HC-NJDG portal to compute the number of cases disposed by each high court judge per day and use these statistics to estimate the number of years required to clear the pendency in the high courts. The number of pending cases is well beyond the capacity of the number of judges currently working in the high courts. Hence, the number of judges should be increased by taking necessary legislative measures.

The energy and efforts put in e-Courts project, NJDG in particular, must continue for few more years, if not decades, to see the real impact. Hence, more aid of ICT in judiciary must be sought to reduce the pendency of millions of cases and the use of artificial intelligence should be more than just welcome.

7 Acknowledgment

Dr. Kshitiz Verma is supported by the Teachers Associateship for Research Excellence (TARE) fellowship (Sanction Number: TAR/2019/000437) provided by the Science and Engineering Research Board (SERB), Department of Science and Technology, India.

References

1. After 28 years, Supreme Court acquits 2 of gang rape. https://timesofindia.indiatimes.com/india/

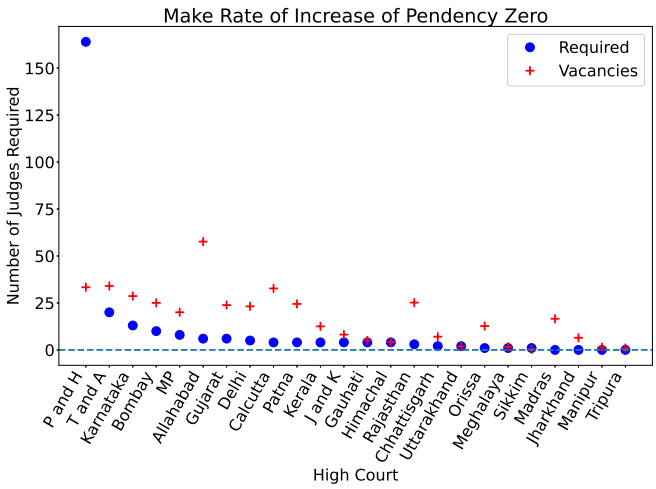


Fig. 8: Number of judges required to make the rate of increase of pendency to zero, i.e., the pendency neither increases nor decreases.

after-28-years-supreme-court-acquits-2-of-gang-rape/articleshow/65615121.cms
2. Appointment of Judges and Judicial Reforms: Need of the hour. https://www.theleaflet.in/appointment-of-judges-and-judicial-reforms-need-of-the-hour/
3. Arrears and Backlog: Creating Additional Judicial (wo)manpower. http://lawcommissionofindia.nic.in/reports/Report_No.245.pdf
4. Arrears, Arrears Everywhere: How Case Pendency Continues to Plague Indian Judiciary. https://www.news18.com/news/india/arrears-arrears-everywhere-how-case-pendency-continues-to-plague-indian-judiciary-1622129.html
5. Can Big Data & Analytics clean up India's Judicial Mess?, https://swarajyamag.com/analysis/can-big-data-analytics-clean-up-indias-judicial-mess
6. CJI Bobde calls for speedy resolution of cases. https://www.thehindu.com/news/national/cji-bobde-calls-for-speedy-resolution-of-cases/article30646807.ece
7. CJI-designate Ranjan Gogoi says he has a plan to tackle judicial backlog. https://www.hindustantimes.com/india-news/cji-designate-ranjan-gogoi-says-he-has-a-plan-to-tackle-judicial-backlog/story-PRqWd4J7mxUjRbZ8BFFatI.html
8. Courts Will Take 320 Years to Clear Backlog Cases: Justice Rao. http://timesofindia.indiatimes.com/india/Courts-will-take-320-years-to-clear-backlog-cases-Justice-Rao/articleshow/5651782.cms
9. Delhi HC to take 466 years to clear all criminal cases. https://www.news18.com/news/india/delhi-hc-to-take-466-years-to-clear-all-criminal-cases-308458.html
10. Evaluation study of e-courts integrated mission mode project. http://doj.gov.in/sites/default/files/Report-of-Evaluation-eCourts.pdf

11. The fundamental reason why Indian courts have a huge backlog of cases. https://www.cnbctv18.com/legal/the-fundamental-reason-why-indian-courts-have-a-huge-backlog-of-cases-5045901.htm
12. Guidelines of the Scheme for Action Research and Studies on Judicial Reforms. http://doj.gov.in/sites/default/files/Action%20Research.pdf
13. In Bombay High Court, 4.64 lakh cases pending. https://indianexpress.com/article/india/in-bombay-high-court-4-64-lakh-cases-pending-5380496/
14. Judicial backlogs can become history. https://www.theleaflet.in/judicial-backlogs-can-become-history/
15. Judicial delay may become a thing of the past. https://www.thehindu.com/opinion/lead/Judicial-delay-may-become-a-thing-of-the-past/article13556341.ece
16. Judicial manpower. https://doj.gov.in/sites/default/files/Judicial-manpower.pdf
17. Judiciary alone can't tackle pendency: HL Dattu. https://www.hindustantimes.com/india/judiciary-alone-can-t-tackle-pendency-hl-dattu/story-oXHQ7wCdJDBVbJjMrNyeNJ.html
18. Lower courts have over 22 lakh cases pending for over 10 years: Report. https://www.ndtv.com/india-news/lower-courts-have-over-22-lakh-cases-pending-for-over-10-years-report-1918439
19. MoP: Age criterion for elevation to High Courts from Bar? Collegium says 45 to 55. https://www.barandbench.com/news/age-criterion-elevation-high-court-bar
20. The National Judicial Data Grid for High Courts. http://njdg.ecourts.gov.in/hcnjdg_public/main.php
21. "NJDG for High Courts is Going to be a Game Changer", Justice Madan Lokur. https://barandbench.com/njdg-high-courts-game-changer-justice-madan-lokur/
22. Pending cases in J&K courts pile up to 147000. https://www.greaterkashmir.com/news/kashmir/pending-cases-in-j-k-courts-pile-up-to-147000/297359.html
23. Reforms could see disposal of cases in three years. https://www.thehindu.com/news/national/reforms-could-see-disposal-of-cases-in-three-years/article2129739.ece
24. State of the Indian Judiciary - a Report by Daksh. http://dakshindia.org/state-of-the-judiciary-report/
25. Supreme Court Annual Reports. https://main.sci.gov.in/publication
26. Unfortunate That Landlord Had To Litigate For Over Four Decades To Retrieve Possession. https://www.livelaw.in/unfortunate-landlord-litigate-four-decades-retrieve-possession-sc-read-judgment/
27. What does data on pendency of cases in Indian courts tell us? https://www.theleaflet.in/what-does-data-on-pendency-of-cases-in-indian-courts-tell-us/
28. Kumar, A.P.: A comparative analysis of courts across India. http://www.commoncause.in/publication_details.php?id=450
29. Shalini Seetharam, S.C.: e-Courts in India: From Policy Formulation to Implementation. https://vidhilegalpolicy.in/wp-content/uploads/2019/05/eCourtsinIndia_Vidhi.pdf
30. Verma, K.: e-Courts Project: A Giant Leap by Indian Judiciary. Journal of Open Access to Law **6**(1) (2018)

Legal aspects of the use of continuous-learning models in Telemedicine*

Chiara Gallese[1 [0000−0001−8194−0261]]

Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands; School of Engineering, Carlo Cattaneo University - LIUC, Varese, Italy c.g.gallese.nobile@tue.nl; cgallese@liuc.it

Abstract. The rapid expansion of the use of telemedicine in clinical practice and the increasing use of Artificial Intelligence (AI) based systems has raised many privacy issues and concerns among legal scholars. Due to the sensitive nature of the data involved, particular attention should be paid to the legal aspects of those systems. This article explores the legal implication of the use of AI in the field of telemedicine, especially when continuous learning and automated decision making systems are involved; in fact, providing personalized medicine through continuous learning systems may represent an additional risk. Particular attention is paid to vulnerable groups, such as children, the elderly, and severely ill patients, due to both the digital divide and the difficulty of expressing free consent.

Keywords: privacy · AI · Telemedicine · GDPR · continuous learning.

1 Telemedicine: a definition

The term "telemedicine" was coined in the 1970s by Thomas Bird and was defined by Strehle and Shabde as "healing at a distance" [25]. A number of official definitions have been added over time, such as: "the provision of healthcare services, through use of ICT, in situations where the health professional and the patient (or two health professionals) are not in the same location. It involves secure transmission of medical data and information, through text, sound, images or other forms needed for the prevention, diagnosis, treatment and follow-up of patients. Telemedicine encompasses a wide variety of services. Those most often mentioned in peer-reviews are teleradiology, telepathology, teledermatology, teleconsultation, telemonitoring, telesurgery and teleophthalmology. Other potential services include call centres/online information centres for patients, remote consultation/e-visits or videoconferences between health professionals. Health information portals, electronic health record systems, electronic transmission of prescriptions or referrals (e-prescription, e-referrals) are not regarded as telemedicine services for the purpose of this Communication." [8], which has been used as a reference for national implementation (such as the definition provided by the Italian Ministry in 2012 [19]).

* Funded by the REMIDE project, Carlo Cattaneo University - LIUC

This new way of providing health care services is not only useful to optimize processes by making them more efficient, and it is not intended to replace traditional in-patient medicine [4], but it also serves to provide the patient with better follow-up, a greater chance of prevention and greater comfort, especially in the case of disabled or particularly frail patients. In fact, compared to traditional medicine, the devices used to monitor the patient from home allow patients not to travel as often to the hospital or to the doctor's office, remaining comfortably at their residence (which may be their own home, but also an hotel, if they fall ill during holidays or business trips). This circumstance is particularly important during the pandemic, as it helps limit the chance of spreading the Sars-cov-2 infection or to get an hospital-acquired infection. It can be said that it was precisely the Covid-19 emergency that gave a boost to the use of medicine, as it sought to ensure continuity of health care even in a context where travel has long been limited.

This incredible opportunity, however, may create some risks, and many legal issues of different nature may arise. In this paper we will focus in particular on the legal issues connected to the use of Artificial Intelligence (AI) in telemedicine, with particular attention to continuous learning systems.

2 Legal framework regarding Telemedicine in Europe

The rapid expansion of the use of telemedicine in clinical practice has prompted, for some time now, the European Union to address the implications of the use of new technologies on patients, the development of the e-Health market, the creation of an European Health Data Space, and the impact that this may have on the health services of Member States. Therefore, over the years, several soft law instruments have been issued, such as guidelines, recommendations, and other tools, analyzed in detail by Botrugno [3].

In addition, the European Regulation on Medical Devices (Regulation(EU) 2017/745 , MDR), fully applicable as of May 26, 2021, disciplines all telemedicine devices used to make a diagnosis and deliver care at a distance.

In Italy as well, the matter is mainly left to soft law, in particular the Guidelines dating back to 2012 [19]. In addition to this, in 2017 Law no. 219 regulated the matter of informed consent and Anticipated Treatment Arrangements in case of possible and future inability to self-determine, a topic that, as will be seen, in the case of telemedicine delivered through intelligent systems takes on particular importance. It has been pointed out, in fact, that if the emotional needs of the patient were to be considered an integral part of the treatment, telemedicine could be qualified only as an integrative service to the traditional ones [5].

To the current regulatory framework, well examined by Campagna [5], will soon be added the new European Regulation on AI (AI Act), approved by the European Commission during 2021. This instrument will bring a very relevant change on AI-based telemedicine devices, as it will impose very stringent requirements for them to be used in medical practice, clinical trials and scientific research. Medical devices, in fact, are classified by the Regulation as "high risk".

The new regulation will complement GDPR and MDR, providing for additional guarantees for users (both patients and health care professionals).

3 Artificial Intelligence in Telemedicine

With the progress of technology, many of AI techniques have been applied to telemedicine, with the aim of improving its results, since, in several areas (such as image recognition), the performance of AI has now surpassed that of humans. However, healthcare professionals are still reticent to use these techniques in clinical practice: data from the research conducted in 2020 on Italian physicians by the Digital Innovation in Healthcare Observatory of the Politecnico di Milano on Connected Care in the Covid-19 emergency showed that only 9% of them used them before the Covid emergency and only 6% work in a facility that introduced (or enhanced) them during the pandemic. Despite this, 60% of medical specialists believe AI techniques can play a key role in emergency situations, 52% believe they help personalize care, 51% believe they help make care more effective, and 50% believe they help reduce the likelihood of clinical errors. Those results are similar to those of other surveys conducted in different countries, where concerns about liability were also raised [23,6]. However, surveys shows that the general public has a great distrust over AI models employed in medicine [29].

Trends in the development of AI use in telemedicine can be classified into four different groups [20]:

- patient monitoring,
- health information technology,
- intelligent diagnostic assistance, and
- collaboration in information analysis.

The branches of medicine in which these techniques are most frequently used are diabetes care, cardiology, ophthalmology, oncology, epidemiology, and dermatology. During the pandemic, telemedicine has been used, among other things, to help the management of the patients who were suspected to be infected and to provide assistance for chronic diseases [30].

Typically, patients wear removable devices such as *smart watches* or sensors, or use an app on their tablets; however, more invasive methods can also be used, such as pills that can be swallowed, cameras placed inside the home, and sensors that monitor actions such as opening medication packages. Low-intrusive techniques can also be employed, such as the use of an app on the cellphone, especially in the case of younger patients, who may engage in telemedicine through a game [10], or older patients [22], who need to be stimulated in engaging with the AI systems. Devices using the more complex techniques also adapt to the individual patient, continuing to learn throughout the duration of use.

This creates a number of ethical and legal problems, on which doctrine and jurisprudence are trying to conduct an in-depth reflection, in order to suggest guidelines for healthcare professionals and researchers who develop these devices. Very often, in fact, the regulatory framework - as in the case of personal data

protection - is complicated even for jurists, a circumstance that represents an obstacle to effective patient protection.

The complex regulatory framework is further complicated by the fragmentation of the AI discipline within the European Union, which the new and ambitious AI Act intends to remedy. Today, however, there remain, and will remain even after the entry into force of this legal instrument, many points peculiar to each Member State in terms of protection of personal data, professional liability of the doctor, informed consent, product liability and criminal liability.

4 Continuous learning and personalized medicine

One of the most popular model to provide personalized medicine is the so-called *continuous learning*. Physicians want to provide the patient with the best possible care, which also means, in some cases, trying to adjust health care services to a specific patient's needs, due to the differences between ethnicities, genders, habits, family anamnesis, psychological state, etc. This goal can effectively be achieved with the help of AI through systems that learn through use by the patient and adapt to their characteristics over time.

From a legal perspective, particular attention should be paid to models based on machine learning, i.e., *machine learning* (especially *deep learning*) and also to those that are based on a *black-box* approach, since in these cases the programmers create the model and provide relevant examples, but they do not know the end result because the model is designed to learn on its own (before it is marketed, during the training phase, but also after it is marketed). This often means that it is not possible to know the reason behind the output given by those models, so it is important to explore and understand how they behave in different scenarios [13]. This problem is not limited to telemedicine, but it is related to all AI applications; however, the use of this type of models in the context of telemedicine poses major risks not only because of the category of data involved, but also due to their potential impact on fundamental rights of patients. These systems may be more dangerous in case of telemedicine than they are in case of in presence medicine especially because the patient is not closely monitored by the doctor, being physically away. We will explore these themes in the following.

We could consider, as an example, the case of a smart watch employed to monitor diabetes in children, which detect blood levels of sugar, measure physiological values (such as the heartbeat), reminds to eat and exercise, and suggest the correct amount of medicine to be assumed by the patient, communicating the output to the doctor. On the basis of this output, the doctor is able to set appointments, testing, and medical evaluations.

In the case of continuous learning, one of the main issues is that neither the programmer nor the physician who will use the AI system can know *a priori* how it will behave and what it will learn from the interaction with the patient, since it is a system that evolves over time.

In fact, four elements should be taken into consideration. The first is that the quality of the training reflects the quality of the samples: if the final user provides the system with biased samples, or samples characterized by poor quality, the behavior of the AI will eventually change to reflect the extended learning set. One notable example was the case of the Tay chatbot [7] [1]. In the case of personalized telemedicine, many examples are provided directly by the patient through the use of the system. Clearly, the patient is not an expert, and is unable to understand the implications of certain choices. For example, a child may lend the smart watch to classmates or to younger siblings, thus providing inexact measurements to the system. The consequence of an inaccurate observation may lead the system to believe that the child has lower level of sugar in the blood, and then suggest to take less medicine than needed.

Secondly, machine learning methods are prone to overfitting, i.e., they tend to lose their capability to generalize. This circumstance – well known to machine learning practitioners – is generally detected by observing the relationship between the training error and the test error: when an improvement of the error on the training set leads to a worse error on the test dataset, the network is overfitting and the training process should be stopped. The latter technique – one of the most widespread and effective to prevent overfitting – is known as early stopping and is clearly not applicable in the case of systems that are supposed to be both sold "market ready" (i.e., the learning process was halted before overfitting) and able to learn outside the factory. In our example, the smart watch may lose its capability to generalize after being used on the patient for a while, both because of an incorrect use or because in that specific time frame the number of specific types of observations were unbalanced (e.g., the patient has been ill for a while, and the heartbeat has not been normal for a long time).

Thirdly, these problems are exacerbated in the case of *lifelong continual learning* [21], i.e., machine learning methods that continuously receive new instances to be used to refine the behavior of the system. As a matter of fact, these methods are challenging because the new samples are often unbalanced (e.g., some categories are more represented than others, due to their probability to be met in the "real world"), a circumstance that can strongly affect the quality of the learning and impact the future behavior of that AI. In the case of the smart watch, it is possible that the specific characteristics of the patients lead to an over-representation of some physiological values, leading to incorrect outcomes.

Finally, the problem is basically not solvable in the extreme case of *generalized class incremental learning* [18], in which the machine learning method receives new instances that can, in principle, belong to new classes/cases never considered before: in this peculiar situation, the algorithm must be able to reconfigure

[1] The case of Tay, the Microsoft's Twitter bot based on AI, that became racist and nazist, is emblematic, as it shows how a model may deviate from the modelist's intention because of the interaction with users after the release to the public: `https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter` accessed 29 November 2020.

its internal functioning (e.g., in the case of deep neural networks, adapt the architecture, change the topology, alter the number of neurons, and re-calibrate all parameters) which clearly prevents any realistic possibility of predicting the future behavior of the system. In our example, the patient may have unique characteristics that lead to unusual physiological levels that were not present in the training data sets. In this scenario, it may be dangerous for the patient, as the system suggestions may be wrong and lead to assume an incorrect amount of medicine, and the doctor is not present to correct the error.

The proposal for the AI Act only superficially tackles the issue of continuous learning at article 15: "[...] High-risk AI systems that continue to learn after being placed on the market or put into service shall be developed in such a way to ensure that possibly biased outputs due to outputs used as an input for future operations ('feedback loops') are duly addressed with appropriate mitigation measures [...]". Recital 78 adds that "In order to ensure that providers of high-risk AI systems can take into account the experience on the use of high-risk AI systems for improving their systems and the design and development process or can take any possible corrective action in a timely manner, all providers should have a post-market monitoring system in place. This system is also key to ensure that the possible risks emerging from AI systems which continue to 'learn' after being placed on the market or put into service can be more efficiently and timely addressed. In this context, providers should also be required to have a system in place to report to the relevant authorities any serious incidents or any breaches to national and Union law protecting fundamental rights resulting from the use of their AI systems". This provision is extremely vague and it does not clarify what possible mitigation measures or correcting actions may be considered adequate. The practical implementation of this paragraph will be difficult due to its opacity and it will leave the interpretation open to judicial discretion. The provision of a "post-market monitoring system" is a measure that may be helpful but scarcely effective when referred to personalized medicine, as producers are not able to constantly monitor every single patient.

Continuous learning poses a major legal problem in multiple respects [15]. For example, it challenges the traditional liability paradigm: is it possible to adapt a strict liability regime to a situation where neither the programmer nor the manufacturer can predict the behavior of the AI? And from a technical point of view, is it safe to use on patients a product that, even if trained by the manufacturer, cannot be fully explained? Some have theorized a so-called *responsibility gap* [16], while others have opposed this view [26]. In these cases, it becomes really difficult to assume professional liability on the part of the physician.

However, many privacy issues arise as well. For example, because the health care professional has no control over the way in which those systems process the patients' health data, they are not able to fully comply with the transparency obligations. We will explore the privacy issues in the next section.

5 Privacy issues in Telemedicine

One of the most relevant aspects in the field of telemedicine and AI is definitely the protection of personal data and, in particular, the European Regulation No. 679/2016 (GDPR). Under the GDPR, health data, along with a few others, are considered "special categories of data" (Art. 9) and are therefore subject to increased legal protection.

Telemedicine, as well as traditional medicine, involves the processing of special categories of personal data, that is health data. Patients' data employed by AI models in the health care sector can rarely be fully anonymous; most often, they are considered pseudonymized, as the hospital or other health care facility is able to match it back with patients' names and other personal details. Even when telemedicine devices are not based on AI, the identity of the patients is most of the time known to the health care professionals, as the goal of the system is to provide health care services to a specific individual. GDPR and other privacy law rules (e.g., national implementation acts, sectoral internal laws, etc.), thus, will apply.

Because of the ways telemedicine is necessarily carried out, there are a number of privacy issues that will always be present. The most obvious one is security, as interactions at a distance imply that a connection must be established (between patients and doctors, but also between different professionals).

Security measure prescribed by article 32 GDPR must be in place: it must be assured that the connection is secure, that the identity of patients and doctors is ascertained, that all persons involved are adequately trained, and that data are correctly stored and deleted when no longer needed.

Especially in cases involving older patients, who are generally not familiar with new technologies, the risk of data breaches is high. IoT devices may be lost, passwords may be cracked by malicious attackers, and data may be deleted by mistake. The fact that the device is in the patient's hand means that the health care professionals and their IT experts are not always present to check the system. Before introducing these devices, therefore, patients should be adequately trained.

However, recent attacks towards hospitals have highlighted that organizational measures implemented by hospitals are not always at the state-of-the-art, mainly due to the lack of proper training of employees, who have a low level of cyber-security awareness [9]. In providing telemedicine services, hospitals and other health care facilities need to make sure that they are able to fulfill the requirements of art. 32 GDPR, which includes training their employees regarding basic cyber-security measures.

If health care professional are not able to train themselves regarding security, it is hardly possible that they are able to monitor how patients interact with the system. Because they are not technical experts, they may also be unable to detect anomalies in the system, such as an incorrect training of the neural network. In this context, it is necessary that devices are regularly checked, updated, and re-calibrated by AI experts.

Additional safeguards with regards to special categories of data are required by the AI Act in the last paragraph of article 10: it is possible to process them to the extent that it is strictly necessary for the purposes of ensuring bias monitoring, detection and correction in relation to high-risk AI systems, but appropriate safeguards are necessary (such as technical limitations on the re-use of data, and the use of state-of-the-art technical measures, including pseudonymisation or encryption). The concept is further explained in Recital 44 [2].

When employing AI system to deliver telemedicine services, privacy by design and by default principles should be carefully implemented in the system, taking into account the data minimization and storage limitation principles as well. To facilitate this, Recital 44 is accounting for an additional specific purpose that may allow the data processing of health data, that is the "bias monitoring, detection and correction", in order to create a fair and trustworthy AI system.

This can be difficult to implement in the case of continuous learning: how can accuracy, up-to-dateness, and data minimization be ensured, if it is not possible to predict how the system will behave?

Another general issue that may arise in telemedicine is the transfer of data abroad. The distance between the patient and the doctor may involve cross-border consultation, subject to different jurisdictions. In addition, the device employed to deliver the service may rely to cloud solutions that may have their servers outside EU. A careful assessment is needed before employing those systems, and a Data Protection Impact Assessment (DPIA) may be needed. Appropriate contractual agreement may also be needed between the hospital and the processors providing the telemedicine devices.

In the case of conflict between different jurisdictions and between different legal systems, it may be difficult to reach a contractual agreement regarding responsibilities and liability of a system that is unpredictable.

5.1 Article 22 GDPR and human oversight

The use of AI-based devices in telemedicine often falls under the definition of automated decision-making (ADM) and profiling (Art. 22), for example in the case of automatic scanning of medical imaging to provide a diagnosis.

Along with the principle of transparency, which guarantees data subjects the right to be informed about how their data will be used and the consequences of processing on them, the GDPR also guarantees an additional right with regard to ADM and profiling: the right to an explanation. This means that "The controller should find simple ways to tell the data subject about the rationale behind, or the criteria relied on in reaching the decision without necessarily always attempting a complex explanation of the algorithms used or disclosure of the full algorithm. The information provided should, however, be meaningful to the data subject" [2].

[2] "In order to protect the right of others from the discrimination that might result from the bias in AI systems, the providers should be able to process also special categories of personal data, as a matter of substantial public interest, in order to ensure the bias monitoring, detection and correction in relation to high-risk AI systems".

In this context, *black box* models, and in particular those based on continuous learning, cannot meet this requirement, as it is not possible to justify to the patient why the model has given a certain output.

The guidelines note that "Complexity is no excuse for failing to provide information to the data subject. Recital 58 states that the principle of transparency is of particular relevance in situations where the proliferation of actors and the technological complexity of practice makes it difficult for the data subject to know and understand whether, by whom and for what purpose personal data relating to him are being collected, such as in the case of online advertising" [2]. This is particularly important in the case of telemedicine, as health data are involved, and the patient is not closely monitored by a doctor. The consequences of a mistake may even be lethal, and this is one of the reasons that led the European Commission to classify medical devices as "high risk systems"[3].

Data controllers, which may be, for example, the hospital or the producer of the telemedicine device, must provide information about the processing and how ADM and profiling might affect data subjects. In Convention 108+, Article 77 provides that "Data subjects should be entitled to know the reasoning underlying the processing of data, including the consequences of such a reasoning, which led to any resulting conclusions, in particular in cases involving the use of algorithms for automated decision making including profiling. For instance in the case of credit scoring, they should be entitled to know the logic underpinning the processing of their data and resulting in a "yes" or "no" decision, and not simply information on the decision itself. Having an understanding of these elements contributes to the effective exercise of other essential safeguards such as the right to object and the right to complain to a competent authority".

Data subjects, in addition, have the right to express their views on ADM and profiling, the right to have the decision affecting them made by a human being, and the right to challenge the decision. In the case of AI-based telemedicine, it may be extremely difficult to manage the remote health care process in such a way that the physician oversees each and every decision made by the system and at the same time ensure that only the physician makes the decision, as this may negate the benefit of using an automated decision. Even if the doctors managed to be constantly present, in the case of continuous learning they still won't be able to tell the patient how the system may behave with them. This will probably be possible in some cases, but in the future, if these devices become widespread and a high degree of integration of telemedicine into the health service on the ground is achieved, it will be essential to keep a close watch on the aspects just outlined.

In this scenario, it is possible to argue that, for the sake of transparency and fairness, interpretable AI models should be used as a standard in telemedicine and black boxes should be employed only in cases in which the health care professional is able to closely oversee the output of the system and make a larger

[3] To analyze the new legal framework envisaged by the AI Act proposal with regards to medical device is out of the scope of this article; however, it will be explored in a different work.

clinical assessment, based on additional elements other than the AI output. Also, continuous learning should be limited only to decisions which are not able to harm the patient in case of errors, and should be constantly monitored by a health care professional.

5.2 Informed Consent

When the legal basis that allows the processing of the patient's health data is consent, which is often the case for telemedicine devices, then GDPR requires it to be both *explicit* and *informed*, other than freely given, specific, and unambiguous. Other than the usual elements that should be communicated to the data subject according to the applicable law, additional information should be provided about how the virtual consultation is performed. This information includes rights under data protection regulations, the possibility of errors in the system, contact protocols during tele-consultations, prescribing policies, and coordination of care with other professionals [17].

Additional transparency requirements are found in the AI Act: article 13 lists the information that should be provided to the users, who also needs to be properly instructed regarding the AI system: "High-risk AI systems shall be accompanied by instructions for use in an appropriate digital format or otherwise that include concise, complete, correct and clear information that is relevant, accessible and comprehensible to users". A detailed technical documentation is required by Article 11 and 18, which also add the requirement of keeping it up to date before the release of the system.

The guidelines on ADM note that "Data controllers seeking to rely on consent as a basis for profiling will need to demonstrate that data subjects understand exactly what they are consenting to, and remember that consent is not always an appropriate basis for processing. In all cases, data subjects should have enough relevant information about the intended use and consequences of the processing to ensure that any consent provided represents an informed choice." [2]. It might be argued that this is inherently not possible in the case of continuous learning, as even the doctor or the modellist won't be able to predict the consequences of using an unpredictable model.

The information received by the patient needs to be concise, transparent, intelligible and in an easily accessible form, using clear and plain language, depending on the audience, adapted to the age, mental ability, and education level of the data subjects.

In any case, it is only possible to use ADM and profiling with special categories of data, such as medical data, if there is explicit consent from the data subject or if it is permitted by law for substantial reasons of public interest, provided there are adequate safeguards in place. It is not enough to use public interest as a legal basis, it must also be considered substantial. It can be argued, for example, that a use aimed at combating Covid-19 falls within the "substantial public interest" grounds.

Informed consent, both from a legal and ethical point of view, becomes a central element of AI systems employed in telemedicine.

5.3 Vulnerable groups: an open issue

Telemedicine using intelligent systems raises more concerns when particularly vulnerable subjects are involved, such as oncological patients, patients with cognitive problems - for example, elderly people suffering from senile dementia or Alzheimer's disease-, children, neurodiverse patients [24], or people who do not speak the language used by the doctor.

First, a major issue is the ability to provide consent that is informed, free, unambiguous, and specific (Art. 7 GDPR). The doctor-patient relationship is itself an unbalanced one, where the parties are not on the same level, and where the patient is in a vulnerable situation; it may be particularly difficult to obtain consent that meets all the requirements of the law from a fragile individual, such as a cancer patient, already prostrated by the disease, might be. If we add to this the right to obtain an explanation, it becomes clear that several problematic points may arise: in telemedicine, the doctor is far from the patient, and human contact is completely lacking. Explanations given remotely may be unclear or misunderstood, as the patient is unable to clearly see the signals of nonverbal communication.

Since these are AI systems, whose functioning, even in the case of explainable and interpretable AI, is often obscure even to experts, the risks of misunderstanding become significant and difficult to eliminate, unless a live consultation is also provided. When the system behavior is not predictable, it becomes even more difficult to explain the consequences of using the device to a vulnerable patient.

Also to be kept in mind is the *digital divide*, that is, the fact that not all patients have the same degree of digital literacy. This circumstance takes on significant weight in the case of telemedicine services provided in the public sphere, since there would be discrimination between users who are able to use the service and users who, for various reasons, are not. This risk is also underlined by the European Commission, which based its opinion, among other things, on the OECD report "Health at a Glance: Europe - 2018", according to which there is a risk of discrimination between users who are able to use the service and those who are not. The Commission notes that there is a direct relationship between the level of education and the number of searches for health information on the web; in fact, similar disparities in the use of digital solutions for health promotion and disease prevention are also likely, and there is a risk that digital tools such as apps, wearable technology, and online forums will not benefit those who need them most, potentially widening health-related inequalities [11].

The divide may even be exacerbated by continuous learning systems, as those who are better at using the telemedicine devices are more likely to get a more precise output. This risk should be evaluated by health care professionals when delivering tele-health services.

5.4 The balance between privacy and protection from harm at distance

As we discussed above, one of the major problems of continuous learning in telemedicine is that the model "evolves" while the patient is physically distant

from the doctor (and from the IT support) and at the same time the system is usually designed to be less intrusive as possible.

This leads to the necessity for a balance between constantly monitoring patients to ensure their safety and preserving their privacy, without making them feel uncomfortable with the technology. Therefore, four elements become relevant: technical measures, organizational measures, psychological factors, and legal requirements.

From a technical point of view, a number of different techniques have been studied to solve some of the privacy issues discussed above, such as problems related to identity management, using blockchain [1,12,28], encryption [14], or federated learning under edge computing [27]. However, some of the problems cannot be solved only using technology: organizational measures play a crucial role in achieving the balance between the protection of privacy and the benefits of surveillance technology as a countermeasure against unexpected malfunction of the system.

In fact, as a first measure, the harm caused by malfunctions of the system can be mitigated by scheduling appropriate maintenance interventions delivered by IT experts and medical doctors, who can test the system, evaluate its performance, and assess the patient's health. These measures can be enacted keeping the burden on patients at the very minimum (e.g., combining the maintenance to the regular in-person appointments at the hospital). However, a second measure is equally important: training doctors and patients on the correct use of telemedicine technology is crucial both to preserve privacy and to mitigate the risk of malfunctioning and incorrect use. Once doctors are correctly trained, for example, they may be able to recognize peculiar and unique features of patients that could lead the self-learning system to generate errors.

From a psychological point of view, vulnerable patients may need additional support with i) understanding the functioning of the system in order to be able to use it correctly and express a free informed consent; ii) accepting the system in their life without feeling a disruption of everyday activities iii) learn how to express their concerns and discomfort (including physical pain) at a distance. A trained therapist, working together with doctors and IT experts, could be appointed to help patients in overcoming those issues. On the other hand, doctors should make sure that patients are able to understand their instructions and they should regularly check if their consent is truly informed.

Privacy law is not an obstacle against the enacting the above-mentioned measures, as GDPR provides for appropriate means to protect patients' fundamental rights. As an additional organizational measure, hospitals should appoint a privacy expert to monitor the use of telemedicine devices and to guide all the stakeholders.

6 Conclusion

In this brief contribution we have seen how telemedicine carried out through continuous learning intelligent systems is a useful tool to ensure better health

care to the patient, but that its diffusion may also bring new challenges to the jurist and to health care professionals.

New legal problems, such as the regulation of systems that adapt to the patient or the protection of vulnerable subjects, and their ability to understand how the AI system works, in order to be able to express a free consent, will have to be addressed and dissected by the doctrine in the near future.

Health care facilities intending to employ AI systems to deliver telemedicine services should train their employees and the patients to handle the system safely and securely, in order to avoid data breaches and an incorrect use of the devices. An efficient way to achieve this goal is to have dedicated teams of IT experts, doctors, and trained therapists.

Although the European Union has tried to give a legal response to the development of AI techniques, there are still many unresolved issues. It is hoped, therefore, that the issue can be adequately explored before telemedicine through AI techniques becomes a widespread and common practice on the EU territory.

References

1. Ahmad, R.W., Salah, K., Jayaraman, R., Yaqoob, I., Ellahham, S., Omar, M.: The role of blockchain technology in telehealth and telemedicine. International journal of medical informatics **148**, 104399 (2021)
2. Article 29 Working Party: Guidelines on Automated individual decision-making and Profiling (2017)
3. Botrugno, C.: Un diritto per la telemedicina: analisi di un complesso normativo in formazione. Politica del diritto **45**(4), 639–668 (2014)
4. Burrai, F., Gambella, M., Scarpa, A.: L'erogazione di prestazioni sanitarie in telemedicina. Giornale di Clinica Nefrologica e Dialisi **33**, 3–6 (2021)
5. Campagna, M.: Linee guida per la telemedicina: considerazioni alla luce dell'emergenza covid-19. Corti supreme e salute **3**, 11 (2020)
6. Castagno, S., Khalifa, M.: Perceptions of artificial intelligence among healthcare staff: a qualitative survey study. Frontiers in artificial intelligence p. 84 (2020)
7. Davis, E.: AI Amusements: The Tragic Tale of Tay the Chatbot. AI Matters **2**(4), 20–24 (Dec 2016). https://doi.org/10.1145/3008665.3008674, `https://doi.org/10.1145/3008665.3008674`
8. European Commission: Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions on telemedicine for the benefit of patients, healthcare systems and society (2008)
9. Gioulekas, F., Stamatiadis, E., Tzikas, A., Gounaris, K., Georgiadou, A., Michalitsi-Psarrou, A., Doukas, G., Kontoulis, M., Nikoloudakis, Y., Marin, S., et al.: A cybersecurity culture survey targeting healthcare critical infrastructures. In: Healthcare. vol. 10, p. 327. Multidisciplinary Digital Publishing Institute (2022)
10. Giunti, G., Ciancaglini, A., Otero, C., Baum, A., de Quirós, F.G.B.: The use of a gamified platform to empower and increase patient engagement in diabetes mellitus adolescents. In: Amia (2014)
11. Hashiguchi Cravo Oliveira, T.: Bringing health care to the patient: An overview of the use of telemedicine in oecd countries. OECD, Directorate for Employment, Labour and Social Affairs, Health Committee (2020)

12. Jain, N., Gupta, V., Dass, P.: Blockchain: A novel paradigm for secured data transmission in telemedicine. In: Wearable Telemedicine Technology for the Healthcare Industry, pp. 33–52. Elsevier (2022)
13. Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J.: Faithful and customizable explanations of black box models. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 131–138 (2019)
14. Ma, M., Fan, S., Feng, D.: Multi-user certificateless public key encryption with conjunctive keyword search for cloud-based telemedicine. Journal of Information Security and Applications **55**, 102652 (2020)
15. Marchant, G.E., Lindor, R.A.: The coming collision between autonomous vehicles and the liability system. Santa Clara Law Review **52**, 1321 (2012)
16. Matthias, A.: The responsibility gap: Ascribing responsibility for the actions of learning automata. Ethics and information technology **6**(3), 175–183 (2004)
17. Membrado, C.G., Barrios, V., Cosín-Sales, J., Gámez, J.M.: Telemedicina, ética y derecho en tiempos de covid-19. una mirada hacia el futuro. Revista Clinica Espanola (2021)
18. Mi, F., Kong, L., Lin, T., Yu, K., Faltings, B.: Generalized class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 240–241 (2020)
19. Ministero della Salute: Telemedicina - Linee di indirizzo nazionali (2012)
20. Pacis, D.M.M., Subido Jr, E.D., Bugtai, N.T.: Trends in telemedicine utilizing artificial intelligence. In: AIP conference proceedings. No. 1, AIP Publishing LLC (2018)
21. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019)
22. Schatten, M., Protrka, R., et al.: Conceptual architecture of a cognitive agent for telemedicine based on gamification. In: Central European Conference on Information and Intelligent Systems. pp. 3–10 (2021)
23. Scheetz, J., Rothschild, P., McGuinness, M., Hadoux, X., Soyer, H.P., Janda, M., Condon, J.J., Oakden-Rayner, L., Palmer, L.J., Keel, S., et al.: A survey of clinicians on the use of artificial intelligence in ophthalmology, dermatology, radiology and radiation oncology. Scientific reports **11**(1), 1–10 (2021)
24. Shaw, S.C., Davis, L.J., Doherty, M.: Considering autistic patients in the era of telemedicine: the need for an adaptable, equitable, and compassionate approach. BJGP open **6**(1) (2022)
25. Strehle, E., Shabde, N.: One hundred years of telemedicine: does this new technology have a place in paediatrics? Archives of disease in childhood **91**(12), 956–959 (2006)
26. Tigard, D.W.: There is no techno-responsibility gap. Philosophy & Technology pp. 1–19 (2020)
27. Wang, R., Lai, J., Zhang, Z., Li, X., Vijayakumar, P., Karuppiah, M.: Privacy-preserving federated learning for internet of medical things under edge computing. IEEE Journal of Biomedical and Health Informatics (2022)
28. Wang, W., Wang, L., Zhang, P., Xu, S., Fu, K., Song, L., Hu, S.: A privacy protection scheme for telemedicine diagnosis based on double blockchain. Journal of Information Security and Applications **61**, 102845 (2021)
29. Yakar, D., Ongena, Y.P., Kwee, T.C., Haan, M.: Do people favor artificial intelligence over physicians? a survey among the general population and their view on artificial intelligence in medicine. Value in Health (2021)
30. Ye, J., et al.: The role of health technology and informatics in a global public health emergency: practices and implications from the covid-19 pandemic. JMIR medical informatics **8**(7), e19866 (2020)

Argumentation Schemes for Blockchain Deanonymization

Dominic Deuber, Jan Gruber[0000−0003−1862−2900],
Merlin Humml[0000−0002−2251−8519], Viktoria Ronge, and Nicole Scheler

Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany
{firstname.lastname}@fau.de

Abstract Cryptocurrency forensics became standard tools for law enforcement. Their basic idea is to deanonymise cryptocurrency transactions with the aim of identifying the people behind such transactions. Cryptocurrency deanonymisation techniques are often based on premises that largely remain implicit, especially in legal practice. This implicitness can have far-reaching consequences for the rights of those affected. Argumentation schemes could remedy this untenable situation by rendering underlying premises transparent and aiding in critically evaluating the probative value of any results obtained by cryptocurrency deanonymisation techniques. In the argumentation theory and AI community, argumentation schemes are influential as they state implicit premises for different types of arguments. Through their critical questions they aid the argumentation participants in critically evaluating arguments. We specialise the notion of argumentation schemes to legal reasoning about cryptocurrency deanonymisation and demonstrate the applicability of the resulting schemes by means of an example case. We envision that the use of our schemes in legal practice can solidify evidential value of blockchain investigations as well as uncover and help to address uncertainty in underlying premises and thus ultimately contribute to protecting the rights of those affected by cryptocurrency forensics.

Keywords: Argumentation · Legal Reasoning · Blockchain Analysis.

1 Introduction

"Follow the money" is arguably the central investigation strategy for any profit-driven offence [32]. Beyond its obvious usefulness in white-collar crime, analysing flows of incriminated money is crucial to understand the business models and inner workings of organised crime groups, the hierarchy of the involved entities, and finally to identify the group's members. However, the fight against money laundering is challenging and criminals utilising virtual currencies as early adopters aggravate the situation even further. While law enforcement agencies need to expend many resources to follow complex transnational flows of fiat currencies, blockchain-based investigations impose even further challenges. These challenges arise from the fact that cryptocurrencies are generally pseudonymous,

with some even being anonymous. Bitcoin [16] is arguably the most famous and widespread cryptocurrency – both for legal economic purposes and criminal activities [6]. Already in the early days of Bitcoin it was shown that the currency is not anonymous, because it is possible to link multiple pseudonyms belonging to the same person [1, 13, 20]. However, also supposedly anonymous cryptocurrencies like Monero [14] or Zcash [33], have been target of deanonymisation attacks [10, 15]. What all attacks on Bitcoin, Monero, and Zcash have in common is that they are based on partly uncertain assumptions [5]. The reliability of these assumptions determines the quality of the results of an attack. In legal practice, those assumptions are critical for inferring the evidential value of the deanonymisation of a perpetrator. However, no standard practice for deriving and discussing reliability of those analysis results has been proposed yet. Therefore, we contribute argumentation schemes for assessing the reliability of investigations on the Bitcoin blockchain.

1.1 Related Work

Argumentation schemes [31] as a way to classify arguments by their underlying principles of convincingness have been influential both in the argumentation theory as well as the artificial intelligence community [11]. They present the various types of arguments as informal deduction rules together with accompanying *critical questions* to aid in evaluating arguments of the respective type.

Given that expert testimonies as well as the court process itself is a form of argumentation, it is not surprising that argumentation schemes were applied to legal processes [2]. Walton [30] gives a detailed overview of the applicability of many argumentation schemes to representing and analysing legal processes. There have also been more formal – and even automated – approaches to legal reasoning based on argumentation theory [19, 2]. However, our goal is not to automate parts of the legal process, but rather to aid in evaluating statements about blockchain deanonymisation.

Postulating application tailored argumentation schemes to capture specialised forms of argument is not a new idea. Parsons et al. [17] introduces schemes to reason about trust in entities to specialise arguments building on statements. Reasoning about treatment choices in order to aid doctors in their decision making and producing automated patient specific recommendations is an area from the medical field where specific argument schemes were developed [25, 26].

On the legal side, evidence must be critically evaluated as investigative measures justified by unreliable results potentially impinge upon fundamental rights of the suspects [21]. Particularly in the case of cryptocurrency investigations, it is important to have a method to judge the reliability of evidence. Fröwis et al. [7] provide key requirements that must be satisfied to safeguard the evidential value of cryptocurrency investigations; one of them being reliability. They suggest specific measures to achieve reliability, such as sharing any information necessary to assess reliability, without discussing how they can be implemented in practice. As a step in that direction Deuber, Ronge and Rückert [5] provide a taxonomy for the different assumptions underlying deanonymisation attacks on cryptocurrency

users. The practical application of their taxonomy, however, is only briefly discussed.

1.2 Contribution

In legal practice, the lack of a profound framework means that there is no standard way to reason about the reliability of findings from blockchain-based investigations. Less reliable findings might cause two issues: First, results with low reliability might not establish the degree of suspicion required by investigative measures and thus result in unlawful actions. In the worst case, any evidence obtained from unlawful investigations might be inadmissible in court depending on the exclusionary rules of the respective jurisdictions. Second, even if it might be admissible, low reliability corresponds with low evidential value and thus the evidence might not be sufficient for a conviction. Given that any findings and the blockchain investigations themselves are highly abstract for most parties involved, there needs to be a common ground between technical analysts, investigators, and other legal practitioners for assessing these findings.

Our contributions are argumentation schemes to assess heuristics employed in investigations based on the Bitcoin blockchain to deanonymise criminal users. The schemes render the taxonomy proposed by Deuber, Ronge and Rückert [5] broadly accessible and easy to use in practice. By presenting the implicit and explicit premises of those heuristics, our argumentation schemes enable all parties involved in the legal process to assess evidential value in a systematic manner. Thus, the schemes have potential to render blockchain-based analyses in Bitcoin more comprehensible and the findings more reliable and conclusive.

2 Preliminaries

2.1 Bitcoin (BTC)

Bitcoin [16] is a decentralised transaction ledger that is maintained in a peer-to-peer network and designed to be append-only. The transactions are organised in blocks, which is why the ledger is also referred to as blockchain. Using a consensus mechanism, the network agrees on which blocks, i.e. particularly transactions, should extend the ledger. The network nodes participating in this consensus mechanism are called *miners*.

Transactions consist of a list of inputs and outputs. An output usually states an amount of Bitcoin (v BTC) and the hash h_{pk} of a public key pk, which is also referred to as address a. An input is a reference to an output of another transaction which is uniquely described by the hash of that other transaction tx_{hash} and the position of the output in the transaction's list of outputs out_{id}. An example of a transaction with one input and two outputs is given in Fig. 1. Usually, transactions have several in- and outputs. To spend the first output of this transaction with an amount of v_1 Bitcoin, it is required to provide a public

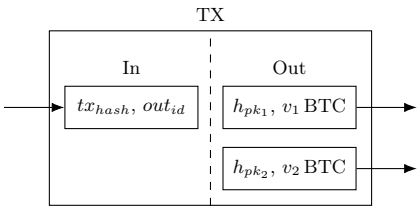


Figure 1: Bitcoin transaction

key pk' whose hash equals h_{pk_1} and a signature that verifies under pk'. The input amount of a transaction is always consumed entirely. Thus, the second output of the transaction might be a so-called *change* output. A change output pays back to the sender(s) the difference between its input amounts and the amount that the recipient(s) should receive.

Wallets in Bitcoin can be seen as a collection of several addresses which belong to the same entity. On a technical level, a wallet is often referred to as software that generates and stores the private keys corresponding to different addresses and allows creating new addresses and issuing transactions. By only inspecting transactions on the blockchain, it is not immediately obvious which addresses belong to the same wallet.

CoinJoin transactions are a special type of transactions that tries to add anonymity to Bitcoin. The idea is to combine inputs from multiple entities while at the same time having equally valued outputs [12]. In Bitcoin, the concept of having transactions with inputs from multiple users to hinder linking, is called *mixing*.

2.2 Bitcoin Investigations

Research has shown early on that Bitcoin is not anonymous but pseudonymous as it is possible to cluster addresses that are likely to be controlled by the same entity, referred to as *address clustering*. The most important address-clustering heuristics are the *multi-input heuristic* [1, 13, 20] and the *change-address heuristic* [13, 1, 10]. The multi-input heuristic states that all inputs of a transaction are controlled by the same entity, which can already be found in Bitcoin's whitepaper [16]. The multi-input heuristic should not be applied to CoinJoin transactions as they are issued by multiple entities by design. The change-address heuristics utilise that change often accrues in Bitcoin (see Section 2.1).

The main objective of blockchain investigations is *re-identification*, that is to determine the natural or legal person who controls an address cluster. This is especially relevant for law enforcement trying to identify persons connected to flows of incriminated virtual currencies. By tracing such transactions and conducting address clustering, they might identify a single relevant address cluster.

As addresses typically do not contain any personal identifiable information, the investigation requires re-identification. To facilitate this, address clusters are usually connected with off-chain information, a process also referred to as *attribution tagging*. As its name implies, the tagged information in attribution tagging can be used to identify the actual entity. The most important attribution information in practice is arguably that an address cluster is related to some cryptocurrency exchange as law enforcement might request the respective customer data from this exchange.

2.3 Legal Background

Many states committed themselves to the fight against cybercrime by ratifying the Convention on Cybercrime [3]. This commitment includes establishing cybercrime offences under domestic law as well as providing investigative measures to enable the prosecution of such offences, while simultaneously protecting fundamental human rights and liberties. The actual balance between the interests of law enforcement and human rights are dictated by the domestic laws of the ratifying states. However, the legal issues discussed in this section are not specific to a particular jurisdiction or legal system. This is illustrated by using the US as an example of a common-law jurisdiction and Germany as an example of a civil-law jurisdiction; both states having ratified the convention.

The starting point for our discussion is the following example case of a typical blockchain-based investigation:

Example. Investigators seized a darknet marketplace and recovered a local Bitcoin wallet that was presumably used to pay the website's operator. The investigators then use blockchain analysis to discover the wallet which was used by the operator to receive payments. While the discovered operator wallet is a local wallet, the operator is suspected to use another wallet at a cryptocurrency exchange that employs a Know-Your-Customer (KYC) policy, to convert Bitcoin into fiat currency. To prevent that the exchange wallet can be linked to the incriminated local wallet, the operator mixes the funds prior to the transfer. The investigators still manage to establish a link between the incriminated local wallet and the exchange wallet through blockchain analysis. In a next step, the investigators issue a request for the disclosure of customer data to the exchange, which collected them to comply with anti-money-laundering laws, to find the natural person that controls the incriminated local wallet. After having identified this suspected operator, the investigators conduct electronic surveillance and execute a search of premises.

In summary, the investigative measures used in the example above were the blockchain analysis, a request for the disclosure of customer data, electronic surveillance, and a search of premises. In general, such investigative measures have in common that they require a specific degree of suspicion in order to protect the rights of the targeted person.

Under German law an *initial suspicion* is sufficient to justify a blockchain analysis (according to Sections 161, 163 German Code of Criminal Procedure

(GCCP), [24, 8]) or a request for the disclosure of customer data (according to Section 100j GCCP). An initial suspicion must be based on a conclusive and established factual basis (factual quality). Due to lax requirements, these measures may be directed not only against the suspected person but also other third parties that might be somehow connected [9, 23]. When it comes to electronic surveillance pursuant to Section 100a GCCP or a search of premises pursuant to Section 102 GCCP there are stronger requirements. Beyond the mere 'possibility' of the commission of a crime, in these cases the suspicion of the crime must be specific and individualised (so-called *qualified initial suspicion*) as well as 'probable' [22, 18]. These measures have to be directed only against the accused person [22] and may only involve other persons who are directly connected to the accused person or involved in the crime (see Sections 100a (3) and 103 GCCP).

Under US law, especially the requirements for the analysis of blockchain data and a request for the disclosure of customer data differ significantly from German law. However, this does not affect the legal issues raised by blockchain analyses as we will point out below. Both, blockchain analyses and the request for the disclosure of customer data are not subject to the probable cause requirement of the Fourth Amendment given that the *third-party doctrine* applies [29]. However, electronic surveillance and search of premises are subject to the Fourth Amendment and therefore require *probable cause* as degree of suspicion. The Fourth Amendment demands the suspicion to be particularised with respect to the person under surveillance, being searched, or specific things to be seized.

The most important legal issue concerning blockchain analysis in practice is whether or not the findings of the analysis can establish the required degree of suspicion for subsequent investigative measures. Therefore, the deviating requirements for blockchain analysis or a request for the disclosure of customer data under US law do not matter as at least subsequent measures, such as searches of premises, require similar degrees of suspicion. In contrast to German law, however, the issue only arises later in the investigation.

To illustrate the legal issue, we return to the example of the darknet marketplace operator. Here, blockchain analysis was used to link an incriminated wallet to an exchange service. Next, disclosure of customer data was requested from the exchange. Imagine that solely based on the linkage of the wallets, further investigative measures are conducted against the natural person identified by the data. If those measures are electronic surveillance or searches of premises, the required suspicion must be particularised against the person targeted by the measures, both, under German and US law. Blockchain analysis might fail to establish this particularised suspicion, if it is unreliable. Imagine that the analysis is based on the multi-input heuristic but the heuristic is applied to CoinJoin transactions. In this case, the analysis would definitely yield false positives as CoinJoin transactions are issued by multiple entities by design. False positives might render the individualisation insufficient and thus the respective investigative measure unlawful.

To summarise, certain invasive and targeted investigative measures require a degree of suspicion that is individualised with respect to the target of these

measures. Blockchain analysis based on uncertain assumptions might lead to unreliable findings that are not sufficient to establish the individualisation and thus the required degree of suspicion for subsequent investigative measures. If investigative measures are conducted without the necessary degree of suspicion they are unlawful and thus might render obtained evidence inadmissible, depending on the exclusionary rules of the respective jurisdiction.

2.4 Argumentation Schemes

Argumentation schemes classify arguments by their warrant in the sense of Toulmin [27] – i.e. by their principle of convincingness. They are presented as informal presumptive deduction rules inferring plausible truth of a conclusion from truth of multiple premises [31]. For example, the *Argument from Abductive Inference* is tailored towards reconstructing the cause E for a set F of observed findings.

Premise:	F is a finding or given set of facts.
Premise:	E is a satisfactory explanation of F.
Premise:	No alternative explanation E' given so far is as satisfactory as E.
Conclusion:	Therefore, E is plausible as hypothesis.

Scheme 1: Argument from Abductive Inference [31]

In addition to the deduction rule representing the informal shape of the argument, an argumentation scheme specifies *critical questions (CQs)* as ways to attack an argument based on the scheme. The critical questions aid both the producer and the receiver of arguments by suggesting relevant statements to present or ask about. There are usually critical questions attacking the individual premises or the conclusion of the argument as well as ones attacking the applicability of the scheme. Consider for example the CQs of the Argument from Abductive Inference:

1. How satisfactory is E as an explanation of F, apart from the alternative explanations available so far in the dialogue?
2. How much better an explanation is E than the alternative explanations available so far in the dialogue?
3. How far has the dialogue progressed? If the dialogue is an inquiry, how thorough has the investigation of the case been?
4. Would it be better to continue the dialogue further, instead of drawing a conclusion at this point?

Scheme 1: Critical questions of Argument from Abductive Inference

CQs 1 and 2 are direct attacks on truth of premises of the rule. CQs 3 and 4 are specific attacks based on the idea that there could be other explanations not yet put forth due to the temporal nature of argumentative dialogues.

By making premises and possible flaws of an argument explicit, argumentation schemes aid critical discussion of expert statements by legal decision-makers and other practitioners without the need for deep understanding of the underlying topic. For judging the reliability of a claim from blockchain analysis it is particularly helpful to have transparency with regards to the underlying assumptions as they have to be judged on a case-by-case basis [5]. This added transparency can also increase the evidential value of such findings if the reliability of dependent information is sufficiently well established.

3 Our Argumentation Schemes

In criminal investigations, blockchain analyses are typically conducted to establish a link between an entity and a criminal offence through involved cryptocurrency addresses. For this purpose, we present a custom argumentation scheme to argue involvement of an entity in an offence from control of an address that is connected to that offence (see Scheme 2).

Premise:	Address A is connected to offence O
Premise:	Entity E controls address A
Conclusion:	Entity E is connected to offence O

1. Which circumstantial evidence indicates that entity E controls address A?
2. Could it be that at the time of offence O someone else controlled address A instead of entity E?
3. How was address A connected to offence O that E's involvement is indicated?
4. Are there other indicators that E is connect to offence O?

Scheme 2: Suspicion through Address Control

We do not need a custom argumentation scheme to represent linking an entity with an address by requesting data from a cryptocurrency exchange as this is covered by *Argument from Position to Know* [31]. This standard scheme covers this case as exchanges typically collect personal information of its customers as part of KYC policies and are thereby in a position to know who the customer using an account is.

To establish a link between addresses, there are software tools implementing various heuristics, such as the multi-input heuristic or change heuristics, which are arguably used by investigators [5]. We pose the *Cluster from Software* scheme to represent arguments based on such a software tool to establish the link between addresses and thereby forming clusters.

Premise: Software S establishes a link between address A_1 and address A_2
Premise: Software S is reliable
Premise: Entity E controls address A_1

Conclusion: Entity E controls address A_2

1. How does software S establish the link?
2. How reliable is software S? Why is software S considered reliable?
3. Could this link be also established without the use of software S, e.g. by using a different software, human-reasoning with the multi-input heuristic, or other non-blackbox methods?
4. What evidence exists for entity E controlling A_1?
5. Are there other indicators that E might control A_2?

Scheme 3: Cluster from Software

Naturally, it is not enough for a software tool to establish the link without further evidence backing that claim. Therefore, an investigator would back the findings of the software by manual analysis in case the software does not disclose the reasons for linking addresses. To represent the claims from manual analysis we present two exemplary schemes capturing the use of the multi-input (see Scheme 4) and the change-address heuristic (see Scheme 5), respectively.

Premise: Transaction T has multiple input addresses
Premise: Entity E controls some input addresses of T

Conclusion: Entity E controls all input addresses of T

1. Could T be a CoinJoin transaction?
2. Could it be that another entity F shares secret keys with E and thereby can control other or all inputs of T?
3. Which input addresses of transaction T does entity E control? What evidence is there for E controlling these addresses?
4. Are there other indicators that E might control other input addresses of T?

Scheme 4: Cluster from Multi-Input

For brevity, the argumentation schemes presented in this section only cover the most common Bitcoin blockchain analysis heuristics used in practice and especially do not cover non-blockchain specific reasoning. For the latter we can make use of the vast array of preexisting schemes [31]. Together, these schemes can be applied to represent reasoning about Bitcoin blockchain investigations in practice as we will show in Section 4.

Premise:	Transaction T has multiple output addresses
Premise:	Output address C is a *change* address of transaction T
Premise:	Entity E controls all input addresses of T
Conclusion:	Entity E also controls *change* address C

1. Could T just have multiple distinct benefactors? Could the change for example be donated to a supported unrelated entity?
2. What evidence is there suggesting that client software was used which generates a fresh change address for every new transaction?
3. Are there other indicators that E controls address C?

Scheme 5: Cluster by Change-Address

4 Application in the Wall Street Market (WSM) Case

In order to illustrate our approach, we present the argumentation behind the investigative results of the proceedings against one of the administrators of the infamous WSM. WSM was one of the largest darknet marketplaces on which illegal narcotics, financial data, hacking software as well as counterfeit goods were traded between approximately 2016 and its seizure in 2019 [4]. Besides technical surveillance measures, blockchain-based investigations of Bitcoin transactions conducted by the US Postal Service (USPS) were a decisive factor for the identification of the administrators operating the marketplace [28]. The publicly available criminal complaint states that the USPS employed proprietary software of an undisclosed company to conduct its blockchain analyses [28]. Furthermore, neither the exact methods employed during the analyses nor the involved Bitcoin addresses were specified, instead the final results – meaning actual investigative findings in the form of off-chain information – were presented on their own.

To prove the correctness, it is merely stated that the software was found to be reliable based on numerous unrelated investigations [28]. This might either suggest the software was utilised as a black box or that the details were (intentionally) not published and kept secret to protect the technical means for tactical reasons. However, in order to understand where further argumentation would have been possible to corroborate the findings of the analysis and convince the legal decision maker of their rightness, we try to infer from the criminal complaint which analysis methods the software might have employed.

The blockchain analyses of the USPS constituted the initial lead that enabled the involved law enforcement agencies to identify 'TheOne' who acted as one of the administrators of the platform [28]. 'TheOne' is believed to be 'Klaus-Martin Frost', one of the three defendants, mainly based on the following two findings:

The investigators could establish a link between the administrator 'TheOne' from WSM and the user 'dudebuy' from *Hansa Market* by analysing data seized from both platforms. They found that 'TheOne' used the same PGP public key as 'dudebuy' did at the previously operated and meanwhile seized darknet marketplace Hansa Market. As a PGP key pair is a highly individual piece of data

used to prove one's identity and encrypt communications, it has to be inferred that those two monikers belong to the same real world entity. As 'dudebuy' used a wallet $W2$ as his *refund wallet* on Hansa Market, the investigators found an entry point to perform financial investigations concerning this perpetrator seeming to operate now as 'TheOne'.

Here, the investigators could establish suspicion using the *Suspicion through Address Control* scheme and infer that the owner of wallet $W2$ seems to be the targeted administrator of the ongoing investigations regarding WSM. This conclusion could be assessed by the evaluation of the critical questions of the scheme. CQ 1 regarding circumstantial evidence leads to a high degree of confidence, as the investigators resorted to seized user data including an identical PGP public key. While CQ 2 does not seem to be of relevance to the investigators at this point in time, CQ 3 reveals an intermediary involvement of the address in the offence in question.

Being confident that the owner of $W2$ is the target, the USPS revealed that other wallets that appeared in the investigations, namely $W1$ and $W4$ were funded by transactions originating from $W2$. As this analysis step is basically rather typical payment flow analysis, which is also employed in traditional money laundering investigations concerning fiat currencies, it is dispensable to assess it with a newly formulated argumentation scheme. For example, *Argument from Sign* or *Argument from Abductive Inference* would be a good fit here [31]. Those newly uncovered wallets, in turn, were identified to be the true origin of several payments to various services, which were conducted via a bitcoin payment processing company (BPPC). Prior to these payments, the corresponding funds were supposedly mixed via a commercial mixing service, whose flow of transactions could be 'de-mixed' by the USPS' analysts [28].

Given the fact that no further information regarding the de-mixing is presented in the criminal complaint, we deliberately assume, that some sort of software established the link, so that the *Cluster from Software* scheme should be employed to be able to judge the evidential value of this result. This scheme revolves around the mechanism for link establishment (CQ 1) the reliability of the tool itself (CQ 2), human comprehensibility (CQ 2) and additional evidence available (CQs 4 and 5). Here, the most important critical question to pose might be CQ 3, i.e. whether the link could be established by comprehensible reasoning of a human analyst, since the following requests for the disclosure of customer data were based on this result, it must be considered crucial evidence in this early phase of investigations.

By obtaining user records from the BPPC regarding the payment from $W1$, investigators uncovered an e-mail address, which could be linked to the before mentioned suspect because it was actually used alongside with his real world identity Klaus-Martin Frost. In addition to that, they uncovered that $W4$ served as the suspected source for payments for two accounts at a video gaming company, which were also linked to the suspect as the records obtained by a subpoena suggest. Furthermore, a second link could be established from another wallet $W5$ that is considered to be used to pay for a third account linked to the suspect at the

gaming company in a similar manner and was found to be funded by a different wallet that could also be associated with WSM's administrators at a later point in time. While this correlation accumulates the reliability, each respective request for the disclosure of customer data might be assessed by employing the *Argument from Position to Know* scheme [31].

In summary, the USPS's blockchain analyses included the following broader steps: *identification of wallets*, *detection of payments between wallets*, *de-mixing* and the *association of wallets with off-chain information* mainly from other darknet marketplaces as well as service providers. While the investigators later found various pieces of evidences in the course of the following investigative actions, these steps were central for the case in order to find a starting point into targeted investigations. We showed that their reliability could be effectively assessed by the utilisation of our argumentation schemes.

5 Conclusion

After having demonstrated the usage of several argumentation schemes for blockchain-based investigations, we conclude by presenting use-cases in which the schemes will be especially beneficial and by pointing out directions for future work.

As our argumentation schemes allow to reason about the findings of blockchain-based investigations, we see potential use-cases wherever such findings have to be communicated to and assessed by persons involved in respective criminal proceedings. By utilising the schemes, an analyst can clearly articulate the employed heuristics, their individual strengths, and potential weaknesses. This increases the comprehensibility of such analyses and court proceedings for the decision makers and also eases the documentation for later verification by an expert witness. Clear articulation is key to determine the quality of blockchain-based findings, especially, if they are not or only weakly supported by other evidence. On the one hand, applying an argumentation scheme and utilising its critical questions, enables law enforcement agencies as well as the preliminary judge to reason about the eventual perpetration of the identified person and therefore establish a certain degree of suspicion to justify further investigative measures. On the other hand, the rights of suspects can be protected by ensuring that the results obtained from blockchain investigations are of quality, can be understood, independently checked for plausibility by the parties to the proceedings, and are actually able to establish the relevant suspicion required by law.

As a result, we consider the application of argumentation schemes in the context of blockchain-based investigations a supportive mechanism for making sense of the intangible crime scene and highly abstract commission of cybercriminal offences. Our schemes can be a helpful tool for investigators and prosecutors that strive to identify perpetrators as well as for legal decision makers to answer the question of guilt. Finally, the schemes are a step forward in the direction of harmonising effectiveness and explainability of high-tech investigations.

Extending this work can be done in multiple directions. Further schemes for other blockchain analysis heuristics or other cybercriminal investigations could be created as indicated already in Section 3. In addition to that, the critical questions of our schemes could be refined to comprise more specific sub-questions as done for *Argument from Expert Opinion* in Walton, Reed and Macagno [31] to capture more expert knowledge.

Acknowledgements This work was supported by DFG (German Research Foundation) as part of the Research and Training Group 2475 "Cybercrime and Forensic Computing" (grant number 393541319/GRK2475/1-2019). Merlin Humml was also supported by DFG project RAND (grant number 377333057). The authors also wish to thank Marie-Helen Maras for fruitful discussions.

References

1. Androulaki, E., *et al.*: Evaluating User Privacy in Bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, pp. 34–51. Springer, Heidelberg (2013). DOI: 10.1007/978-3-642-39884-1_4
2. Atkinson, K., and Bench-Capon, T.J.M.: Argumentation schemes in AI and Law. Argument & Computation 12(3), 417–434 (2021). DOI: 10.3233/AAC-200543
3. Council of Europe: Convention on Cybercrime. COM(2020) 605 final, (2001)
4. Department of Justice - Office of Public Affairs: Three Germans Who Allegedly Operated Dark Web Marketplace with Over 1 Million Users Face U.S. Narcotics and Money Laundering Charges, Department of Justice. (2019). https://www.justice.gov/opa/pr/three-germans-who-allegedly-operated-dark-web-marketplace-over-1-million-users-face-us (visited on 07/12/2020)
5. Deuber, D., Ronge, V., and Rückert, C.: SoK: Assumptions underlying Cryptocurrency Deanonymizations - A Taxonomy for Scientific Experts and Legal Practitioners. Proc. Priv. Enhancing Technol. (In press)
6. European Union Agency for Law Enforcement Cooperation: IOCTA 2021: internet organised crime threat assessment 2021, (2021). https://data.europa.eu/doi/10.2813/113799. DOI: 10.2813/113799
7. Fröwis, M., *et al.*: Safeguarding the evidential value of forensic cryptocurrency investigations. Digit. Investig. 33, 200902 (2020). DOI: 10.1016/j.fsidi.2019.200902
8. Grzywotz, J., Köhler, O.M., and Rückert, C.: Cybercrime mit Bitcoins - Straftaten mit virtuellen Währungen, deren Verfolgung und Prävention. StV 11, 753–759 (2016)
9. Hauschild: Münchener Kommentar zur StPO, § 152. Hans Kudlich (2014)
10. Kappos, G., *et al.*: An Empirical Analysis of Anonymity in Zcash. In: Enck, W., and Felt, A.P. (eds.) USENIX Security 2018, pp. 463–477. USENIX Association (2018)
11. Macagno, F.: Argumentation schemes in AI: A literature review. Introduction to the special issue. Argument Comput. 12(3), 287–302 (2021). DOI: 10.3233/AAC-210020
12. Maxwell, G.: CoinJoin: Bitcoin privacy for the real world, (2013). https://bitcointalk.org/index.php?topic=279249 (visited on 12/08/2019)
13. Meiklejohn, S., *et al.*: A fistful of bitcoins: characterizing payments among men with no names. In: Internet Measurement Conference, pp. 127–140 (2013)

14. Monero, https://www.getmonero.org/ (visited on 12/08/2019)
15. Möser, M., *et al.*: An Empirical Analysis of Traceability in the Monero Blockchain. PoPETs 2018(3), 143–163 (2018). DOI: 10.1515/popets-2018-0025
16. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, (2008)
17. Parsons, S., *et al.*: Argument schemes for reasoning about trust. Argument & Computation 5(2-3), 160–190 (2014). DOI: 10.1080/19462166.2014.913075
18. Peters: Münchener Kommentar zur StPO, § 102. Hans Kudlich (2016)
19. Prakken, H.: Historical Overview of Formal Argumentation. In: Handbook of Formal Argumentation Vol. 1 (2017)
20. Reid, F., and Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Security and privacy in social networks, pp. 197–223. Springer, Heidelberg (2013)
21. Rückert, C.: Cryptocurrencies and fundamental rights. J. Cybersecur. 5(1), 1–12 (2019). DOI: 10.1093/cybsec/tyz004
22. Rückert, C.: Münchener Kommentar zur StPO, § 100a. Hans Kudlich (2022)
23. Rückert, C.: Münchener Kommentar zur StPO, § 100j. Hans Kudlich (2022)
24. Safferling, C., and Rückert, C.: Telekommunikationsüberwachung bei Bitcoins - Heimliche Datenauswertung bei virtuellen Währungen gem. §100a StPO. MMR (2015)
25. Sanchez Graillet, O., and Cimiano, P.: Argumentation Schemes for Clinical Interventions. Towards an Evidence-Aggregation System for Medical Recommendations. In: Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing HEALTHINFO 2019 (2019)
26. Sassoon, I., *et al.*: Argumentation schemes for clinical decision support. Argument Comput. 12(3), 329–355 (2021). DOI: 10.3233/AAC-200550
27. Toulmin, S.: The Uses of Argument. Cambridge University Press (1958)
28. United States Discrict Court for the Central District of California: Criminal Complaint - United States of America v. Tibo Lousee, Klaus-Martin Frost, and Jonathan Kalla - Case No. 19MJ1843, (2019). https://www.justice.gov/opa/press-release/file/1159706/download (visited on 07/12/2020)
29. *United States v. Gratkowski*, 964 F.3d 307 (5th Cir. 2020),
30. Walton, D.: Legal Reasoning and Argumentation. In: Handbook of Legal Reasoning and Argumentation, pp. 47–75. Springer Netherlands (2018). DOI: 10.1007/978-90-481-9452-0_3
31. Walton, D., Reed, C., and Macagno, F.: Argumentation Schemes. Cambridge University Press (2008). DOI: 10.1017/CBO9780511802034
32. Wechsler, W.F.: Follow the money. Foreign Affairs 80, 40 (2001)
33. Zcash, https://z.cash/ (visited on 12/08/2019)

Author Index

ISBN 978-4-915905-95-7