

**Proceedings of the 18th International Workshop
on Juris-Informatics
(JURISIN 2024)**

*in association with
the 16th JSAI International Symposia on AI (JSAI-isAI 2024)*

JURISIN 2024 Co-chairs

Ken Satoh, Center for Juris-Informatics, Japan

Nguyen Le Minh, Japan Advanced Institute of Science and Technology, Japan

May 28-29, 2024

Preface

This volume contains 13 papers which were selected for a presentation at the 18th International Workshop of Juris-Informatics (JURISIN 2024) held in Hamamatsu, Japan on May 28-29, 2024, but were not selected for publication in an LNAI volume for the 16th JSAI International Symposia on AI (JSAI-isAI 2024).

Juris informatics is an interdisciplinary discipline that studies various legal issues from an informatics perspective.

The international workshop on juris informatics, JURISIN, began in 2007 and has been held once a year with the support of the Japanese Society for Artificial Intelligence. Although only nine related topics were exemplified in the first JURISIN call for papers, including legal reasoning, argumentation agents, and legal ontology, in recent years, the development of artificial intelligence technology has greatly expanded the scope of problems to be solved, including the use of machine learning and the legal and social problems caused by artificial intelligence.

We received 30 submissions and each paper was reviewed by three reviewers, from which 12 papers were accepted for an LNAI volume and 14 papers were accepted for this volume but one paper was withdrawn. So 13 papers are included in this volume. In this volume, there are important research themes such as representation of legal knowledge, as well as research themes that have been the focus of much attention in recent years, such as NLP using machine learning. In addition to the general presentations, two invited talks were given by Prof. Mihoko Sumida (Hitotsubashi University, Japan) and Prof. Francesca Toni (Imperial College, UK).

Finally, we would like to express our deepest gratitude to those who submitted papers, to the PC members who reviewed the papers, and to the Japanese Society for Artificial Intelligence for providing the venue for this workshop.

May 28 and 29, 2024
Tokyo, Japan

Ken Satoh
Nguyen Le Minh

Table of Contents

Extracting Data from Patent Prosecution	1
<i>Jieh-Sheng Lee</i>	
Balancing Exploration and Exploitation in LLM using Soft RLLF for Enhanced Negation Understanding	14
<i>Ha-Thanh Nguyen and Ken Satoh</i>	
How to Establish Legally Binding Smart Contracts on Loan and Credit Agreements	28
<i>Sieh-Chuen Huang, Yun-Cheng Tsai and Hsuan-Lei Shao</i>	
On Debugging Structural Legal Rules in Natural Languages using Large Language Models	43
<i>Wachara Fungwacharakorn, May Myo Zin, Ha-Thanh Nguyen, Hideaki Takeda and Ken Satoh</i>	
Mind the Gap - The Rules of the Road for Humans and Machines	55
<i>Galileo Sartor, Adam Wyner, Giuseppe Contissa and Jacinto Davila</i>	
Forensic Analysis of Social Media IOS Apps via Reconstructing Timelines	71
<i>Oluwafisayo Theophilus and Hongmei Chi</i>	
Enhancing Legal Document Retrieval: A Multi-Phase Approach with Large Language Models	86
<i>Hai-Long Nguyen, Duc-Minh Nguyen, Tan-Minh Nguyen, Ha-Thanh Nguyen, Thi-Hai-Yen Vuong and Ken Satoh</i>	
An Empirical Evaluation of Using ChatGPT to Summarize Disputes for Recommending Similar Labor and Employment Cases in Chinese	101
<i>Po-Hsien Wu, Chao-Lin Liu and Wei-Jie Li</i>	
Towards General Requirements for Norm Representation Languages Using Competency Questions: The Case of Comparing Flint and ODRL	115
<i>Jeroen Breteler and Thom Van Gessel</i>	
Development of a Multilingual Model for Mapping Japanese and Foreign Laws	128
<i>Daichi Yamada and Makoto Nakamura</i>	
A hybrid approach for accessible rule-based reasoning through large language models	142
<i>Marco Billi, Alessandro Parenti, Giuseppe Pisano and Marco Sanchi</i>	
A Hybrid Approach to Legal Textual Entailment	154
<i>Cor Steging and Ludi van Leeuwen</i>	

Legal Yes/No Question Answering Through Text Embedding, Fine-Tuning, and Prompt Engineering	170
<i>Housam Babiker, Md Abed Rahman, Mi-Young Kim, Juliano Rabelo and Randy Goebel</i>	

Program Committee

Ryuta Arisaka	Kyoto University
Giuseppe Contissa	University of Bologna
Marina De Vos	University of Bath
Wachara Fungwacharakorn	National Institute of Informatics
Saptarshi Ghosh	Indian Institute of Technology Kharagpur
Randy Goebel	University of Alberta
Guido Governatori	Charles Sturt University
Tokuyasu Kakuta	Chuo University
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	University of Alberta
Nguyen Le Minh	Japan Advanced Institute of Science and Technology
Davide Liga	University of Luxembourg
Réka Markovich	University of Luxembourg
Makoto Nakamura	Niigata Institute of Technology
María Navas-Loro	Universidad Politécnica de Madrid
Ha-Thanh Nguyen	Center for Juris-Informatics
Katsumi Nitta	Tokyo Institute of Technology
Yasuhiro Ogawa	Nagoya City University
Adrian Paschke	Freie Universität Berlin
Juliano Rabelo	University of Alberta
Livio Robaldo	University of Swansea
Víctor Rodríguez Doncel	Universidad Politécnica de Madrid
Seiichiro Sakurai	Meiji Gakuin University
Ken Satoh	Center for Juris-Informatics
Jaromir Savelka	Carnegie Mellon University
Akira Shimazu	Japan Advanced Institute of Science and Technology
Satoshi Tojo	Asia University
Katsuhiko Toyama	Nagoya University
Vu Tran	The Institute of Statistical Mathematics
Bart Verheij	University of Groningen
Sabine Wehnert	Leibniz Institute for Educational Media — Georg Eckert Institute
Yueh-Hsuan Weng	Tohoku University
Hannes Westermann	University of Montreal
Hiroaki Yamada	Tokyo Institute of Technology
Masaharu Yoshioka	Hokkaido University
May Myo Zin	Center for Juris-Informatics
Thomas Ågotnes	University of Bergen

Extracting Data from Patent Prosecution

Jieh-Sheng Lee^[0000–0002–0990–6170]

National Yang Ming Chiao Tung University School of Law
No. 1001, Daxue Rd. East Dist., Hsinchu City 300093, Taiwan
jasonlee@nycu.edu.tw

Abstract. This manuscript demonstrates the extraction of structured information from patent prosecution, an area often overlooked. It features two examples of data extraction from office actions, which are documents issued by patent examiners as written opinions to address matters of patentability. The first example shows the extraction of data from tables in images, utilizing conventional image processing and OCR techniques. This method outperforms recent OCR-free approaches in the specific scenario presented in this manuscript. The second example focuses on identifying the relationships between patent claims and prior art references. In other legal fields, the approach demonstrated in the first example may offer a more effective method for processing image tables compared to recent approaches in visual document understanding. For the patent field, this manuscript highlights the opportunity to tap into a larger pool of data from the patent prosecution process. This direction establishes the groundwork for building larger training datasets and more effective language models in the future.

Keywords: Patent · Language Model · Natural Language Processing · Image Processing.

1 Introduction

The effectiveness of language models hinges critically on the breadth and quality of their training datasets. In the patent field, both granted patents and published applications offer a readily accessible and rich source of data. Utilizing these corpuses as the training data is foundational in the initial stages of building a patent-specific language model. However, for a more thorough coverage and to enhance the model’s capabilities, it is essential to include various intermediate documents that are part of the patent prosecution process. Patent prosecution is the process of obtaining a patent for an invention. Throughout the process, an “office action” (OA) is issued by a patent examiner as a written opinion to address matters of patentability. Various types of opinions are included in these office actions. For instance, in the United States, a “rejection” is the refusal to grant claims when the substance of the patent claims are deemed to be unallowable under 35 U.S.C. 101, 102, 103, 112, or other relevant provisions. If the issue with the claim pertains to its format rather than its substance, it is referred to as an “objection.” When a patent application is “allowed,” it means the examiner

has concluded that the invention is patentable and the application has met all procedural and substantive requirements.

Office action documents contain a wealth of structured information. In this manuscript, two instances of this structured content are showcased to illustrate methods of data extraction. The first instance involves extracting data from tables through image processing and Optical Character Recognition (OCR). The second instance is centered on pinpointing the relationships between patent claims and prior art references. The first example offers insights applicable to other legal fields, whereas the latter, being domain-specific, serves as a useful reference for the researchers in the patent field. In addition, office actions are written opinions containing patent examiner’s feedback in patent prosecution process. Office actions are “human feedback” regarding the matters of patentability. Consequently, in the context of Reinforcement Learning from Human Feedback (RLHF), extracting data from office actions lays the foundation for future implementation of RLHF in the patent field.

2 Related Work

The related work for this study falls into two categories. The first category is domain-agnostic and deals with PDF files, image processing, and OCR. The second category includes models and data sources that are specific to patents. In this first category, the task of extracting data from PDF files and images, especially when they include tables, presents a challenging problem. Tables can be intricate, often containing more than just text. Traditional methods have been evolving for years and are capable of addressing these tasks effectively to a considerable extent. For instance, various open-source Python libraries, such as *PyMuPDF* and *PDFMiner*, are capable of analyzing and manipulating PDF files [4]. For OCR, *Tesseract* [1] has been a widely recognized engine for years. Yet, the OCR field is still challenging and evolving. Recent developments in visual document understanding have introduced new methods that combine pixel-based techniques and Transformer-based models. These novel approaches strive to provide an end-to-end solution that operates without relying on traditional OCR methods.

Two significant examples of such solutions include Google Cloud’s Document AI [3] and Microsoft Azure AI’s Document Intelligence [8], which is powered by GPT-4 Turbo with Vision. According to the Google, Document AI is a document understanding platform that takes unstructured data from documents and transforms it into structured data, making it easier to understand, analyze, and consume. It uses natural language processing and computer vision to extract structured information from documents. According to Microsoft, Document Intelligence is an AI service that applies advanced machine learning to extract text, key-value pairs, tables, and structures from documents. The implementation described in this manuscript will be compared with these two solutions.

Concerning the second category of related work, the largest language model built specifically for the patent field is the PatentGPT-J-6B model, as detailed

in [5] and publicly released in [6]. Despite being exclusively pre-trained on patent data, it is worth noting that patent examiners’ opinions in office actions have not been utilized for generative language models. According to [7], the relative inaccessibility of office actions has prevented researchers from fully exploiting valuable information during patent prosecution. The authors in [7] aim to rectify the situation by using natural language processing and machine learning techniques to systematically extract information from office actions and construct a relational database of key data elements in [7].

In [7], the database covers 4.4 million office actions mailed during the 2008 to mid-2017 period from the United States Patent and Trademark Office (USPTO) to the applicants of 2.2 million unique patent applications. Based on [7], the USPTO released the Office Action Research Dataset for Patents [13] as the first dataset of its kind. However, after inspection in this research, it becomes evident that this dataset primarily offers categorized information, including various types of rejections, such as 101 Rejection (Subject Matter Eligibility, Statutory Double Patenting, Utility, etc.), 102 Rejection (Lack of Novelty), 103 Rejection (Obviousness), and 112 Rejection (Written Description, Indefinite Claims, etc.). The dataset lacks the pertinent prior art references and the patent examiner’s opinions explaining the reasons for rejecting a patent claim.

Aside from the Office Action dataset, the USPTO offers various additional research datasets as detailed in [17]. In addition, the USPTO provides other data sources, including the PatentsView website [16], which facilitates data visualization and analysis, the Bulk Data Storage System [12] serving as a repository for raw public bulk data, and the Patent Examination Data System [15], which allows users to search, display, and download multiple records related to patent applications, status, and transaction history. In the realm of academia, there is a newer dataset known as the Harvard USPTO Dataset (HUPD) [11]. This dataset provides a large-scale, well-structured, and multi-purpose corpus of utility patent applications filed to the USPTO between January 2004 and December 2014. However, it should be noted that none of these datasets or data sources are immediately suitable for application in RLHF or instruction tuning in the patent field.

3 Implementations

This section describes the details of two implementations in this manuscript. The first implementation focuses on image processing, particularly on utilizing OCR, which may hold relevance for other legal fields. The second implementation centers on text processing, tailored specifically to the domain of patents. In the majority of legal fields, image and text are the two most common modalities for machine learning applications.

3.1 Image Processing

The example provided primarily utilizes *OpenCV* and *Tesseract*. The PDF document to process in this example, as shown in Fig. 1a, is an “Index of Claims” file

in office actions. The file provides an extensive record of claim statuses across various dates. Utilizing this file allows for a clear view of which claims were rejected, cancelled, allowed, or appealed on specific dates. Since the *OpenCV* library does not support PDF formats, the first step requires converting a PDF file into an image format, for example, using *PyMuPDF*. To identify tables within an image, the idea is to use the *findContours()* function in *OpenCV*. This function is capable of detecting all contours in the image. For instance, in step 1 of Fig. 1, the most extensive contour identified is the one that encloses all the tables. The second largest contour is the “CLAIMS” table, which is highlighted and surrounded by a gray border. This particular table is the target for further data extraction. The next step involves eliminating the unwanted cells within the table. In this example, these unwanted cells are larger than the target cells. Consequently, by continuously removing “the second largest contour” (background in black), the unwanted cells can be effectively discarded. This process is illustrated in steps 2, 3, and 4 of Fig. 1. Following step 4, the subsequent step will discard the “CLAIM” cell in the figure. This step is omitted here fore brevity.

Index of Claims		Application/Control No.	Applicant(s)/Patent Under Reexamination
		17/386,389	Seth et al.
		Examiner	Art Unit
		JAE U JEON	2193

☒ Rejected	☐ Cancelled	☐ Non-Elected	☐ Appeal
☐ Allowed	☐ Restricted	☐ Interference	☐ Objected

CLAIMS									
☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47									
CLAIM	DATE								
Final	Original	02/04/2023	06/06/2023	10/21/2023					
1	✓	✓	✓	✓					
2	✓	✓	✓	✓					
3	✓	✓	✓	✓					
4	✓	✓	✓	✓					
5	✓	✓	✓	✓					
6	✓	✓	✓	✓					
7	✓	✓	✓	✓					
8	✓	✓	✓	✓					
9	✓	✓	✓	✓					
10	✓	✓	✓	✓					
11	✓	✓	✓	✓					
12	✓	✓	✓	✓					
13	✓	✓	✓	✓					
14	✓	✓	✓	✓					
15	✓	✓	✓	✓					
16	✓	✓	✓	✓					
17	✓	✓	✓	✓					
18	✓	✓	✓	✓					
19	✓	✓	✓	✓					
20	✓	✓	✓	✓					

(a) Step 1

CLAIMS									
CLAIM	DATE								
Final	Original	02/04/2023	06/06/2023	10/21/2023					
1	✓	✓	✓	✓					
2	✓	✓	✓	✓					
3	✓	✓	✓	✓					
4	✓	✓	✓	✓					
5	✓	✓	✓	✓					
6	✓	✓	✓	✓					
7	✓	✓	✓	✓					
8	✓	✓	✓	✓					
9	✓	✓	✓	✓					
10	✓	✓	✓	✓					
11	✓	✓	✓	✓					
12	✓	✓	✓	✓					
13	✓	✓	✓	✓					
14	✓	✓	✓	✓					
15	✓	✓	✓	✓					
16	✓	✓	✓	✓					
17	✓	✓	✓	✓					
18	✓	✓	✓	✓					
19	✓	✓	✓	✓					
20	✓	✓	✓	✓					

(b) Step 2

CLAIM	DATE								
Final	Original	02/04/2023	06/06/2023	10/21/2023					
1	✓	✓	✓	✓					
2	✓	✓	✓	✓					
3	✓	✓	✓	✓					
4	✓	✓	✓	✓					
5	✓	✓	✓	✓					
6	✓	✓	✓	✓					
7	✓	✓	✓	✓					
8	✓	✓	✓	✓					
9	✓	✓	✓	✓					
10	✓	✓	✓	✓					
11	✓	✓	✓	✓					
12	✓	✓	✓	✓					
13	✓	✓	✓	✓					
14	✓	✓	✓	✓					
15	✓	✓	✓	✓					
16	✓	✓	✓	✓					
17	✓	✓	✓	✓					
18	✓	✓	✓	✓					
19	✓	✓	✓	✓					
20	✓	✓	✓	✓					

(c) Step 3

CLAIM	DATE								
Final	Original	02/04/2023	06/06/2023	10/21/2023					
1	✓	✓	✓	✓					
2	✓	✓	✓	✓					
3	✓	✓	✓	✓					
4	✓	✓	✓	✓					
5	✓	✓	✓	✓					
6	✓	✓	✓	✓					
7	✓	✓	✓	✓					
8	✓	✓	✓	✓					
9	✓	✓	✓	✓					
10	✓	✓	✓	✓					
11	✓	✓	✓	✓					
12	✓	✓	✓	✓					
13	✓	✓	✓	✓					
14	✓	✓	✓	✓					
15	✓	✓	✓	✓					
16	✓	✓	✓	✓					
17	✓	✓	✓	✓					
18	✓	✓	✓	✓					
19	✓	✓	✓	✓					
20	✓	✓	✓	✓					

(d) Step 4

Fig. 1: Identify the table and remove unwanted cells

In step 1 of Fig. 1, the eight statuses for claims and their corresponding symbols or characters are defined as follows: Rejected ($\checkmark$), Cancelled (-), Non-Elected (N), Appeal (A), Allowed (=), Restricted ($\div$), Interference (I), and Objected (O). For this implementation, images of these representative symbols or characters are manually extracted. This manual extraction is chosen due to its minimal effort compared to the coding involved. The extracted images are then saved and utilized for comparison with the contours that contain a symbol or character. Continuing with the use of the *findContours()* function in Fig. 2a, all cells are successfully identified. At this point, the complexity of the original image has been effectively reduced, allowing for accurate cell detection. Next, *Tesseract* is employed to correctly extract the text from column headers. For this specific table, a cell is designated as a header if its text corresponds to a date or is the word “Original.”

Once the column headers are pinpointed, it becomes straightforward to determine the cells directly beneath each header by checking their coordinates, width, and height. *Tesseract* can effectively recognize the numerical values under the “Original” column. However, for cells under the dated headers, *Tesseract* faces recognition issues. To address the issues, this implementation utilizes the previously extracted images representing claim statuses for comparison. The approach involves conducting a structural similarity comparison using the *structural_similarity* function from the *skimage* library. This method facilitates the identification of each cell in accordance with its corresponding symbol or character. Demonstrating this as a proof of concept, the recognitions are labelled as “Rj” (Rejected), “Cn” (Cancelled), “Ne” (Non-Elected), “Ap” (Appeal), “Aw” (Allowed), “Rs” (Restricted), “In” (Interference), or “Ob” (Objected). Fig. 2a displays the outcomes of these image processing steps, and the successful recognitions are highlighted in green.

For comparison purposes, the recognition results obtained from two commercial platforms are displayed in Fig. 2b (Google Document AI [3]) and Fig. 2c (Azure AI, Document Intelligence Studio [8]). It’s important to note that these outcomes are the default, out-of-the-box results, achieved without any fine-tuning or adjustments to parameters. Furthermore, these platforms are designed to be universal, equipped to handle a wide range of images, layouts, and complex tables. Thus, this comparison isn’t intended to be equitable. The key point highlighted here is that, by leveraging open-source tools and effectively separating the tables and cells in the image, it is feasible to achieve performance that surpasses that of commercial platforms. An additional example is shown in Fig. 3. The rationale behind these results is similar to that in Fig. 2, and is thus not included here to maintain brevity.

3.2 Text Processing

This section describes the details of implementing a parser to analyze office actions. The objective of the parser is to identify the relations among patent claims, prior art references, and patent examiner’s opinions. The input file to the parser is in XML format and downloaded from the “documents & transaction

[illegible]

(a) using OpenCV + Tesseract

Index of Claims 	Application/Control No. FT095-308 Examinee DAE UJ JEON	Applicant(s)/Patent Under Reexamination Sethi et al Attorney 2153
---	---	--

☒ Rejected	☐ Cancelled	☐ Non-Elected	☐ Appeal
☐ Allowed	☐ Restricted	☐ Interference	☐ Objected

CLAIMS

☐ Claims renumbered in the same order as presented by applicant
 ☐ CIPM
☐ EUB
☐ EPO

#	Original	304/2009	306/2009	306/2009	DATE			
1	0	0	0	0				
2	0	0	0	0				
3	0	0	0	0				
4	0	0	0	0				
5	0	0	0	0				
6	0	0	0	0				
7	0	0	0	0				
8	0	0	0	0				
9	0	0	0	0				
10	0	0	0	0				
11	0	0	0	0				
12	0	0	0	0				
13	0	0	0	0				
14	0	0	0	0				
15	0	0	0	0				
16	0	0	0	0				
17	0	0	0	0				
18	0	0	0	0				
19	0	0	0	0				
20	0	0	0	0				
21	0	0	0	0				
22	0	0	0	0				
23	0	0	0	0				
24	0	0	0	0				
25	0	0	0	0				
26	0	0	0	0				
27	0	0	0	0				
28	0	0	0	0				
29	0	0	0	0				
30	0	0	0	0				
31	0	0	0	0				
32	0	0	0	0				
33	0	0	0	0				
34	0	0	0	0				
35	0	0	0	0				
36	0	0	0	0				
37	0	0	0	0				
38	0	0	0	0				
39	0	0	0	0				
40	0	0	0	0				
41	0	0	0	0				
42	0	0	0	0				
43	0	0	0	0				
44	0	0	0	0				
45	0	0	0	0				
46	0	0	0	0				
47	0	0	0	0				
48	0	0	0	0				
49	0	0	0	0				
50	0	0	0	0				
51	0	0	0	0				
52	0	0	0	0				
53	0	0	0	0				
54	0	0	0	0				
55	0	0	0	0				
56	0	0	0	0				
57	0	0	0	0				
58	0	0	0	0				
59	0	0	0	0				
60	0	0	0	0				
61	0	0	0	0				
62	0	0	0	0				
63	0	0	0					

(b) Google Document AI

Index of Claims		Application/Control No. 17/386,389	Applicant(s)/Patent Under Reexamination Serhi et al.	
		Examiner JAE U JEON	Art Unit 2193	

☒ Rejected	- Cancelled	N Non-Elected	A Appeal
= Allowed	+ Restricted	I Interference	X Objected

CLAIMS					
☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ FL1.47					
CLAIM	Original	10/24/2003	06/29/2003	10/24/2003	DATE
1	✓	✓	✓		
2	✓	✓	✓		
3	✓	✓	✓		
4	✓	✓	✓		
5	✓	✓	✓		
6	✓	✓	✓		
7	✓	✓	✓		
8	✓	✓	✓		
9	✓	✓	✓		
10	✓	✓	✓		
11	✓	✓	✓		
12	✓	✓	✓		
13	✓	✓	✓		
14	✓	✓	✓		
15	✓	✓	✓		
16	✓	✓	✓		
17	✓	✓	✓		
18	✓	✓	✓		
19	✓	✓	✓		
20	✓	✓	✓		

(c) Azure AI

Fig. 2: Benchmark 1

history” web page at the USPTO Patent Center [14]. Based on the XML format, each paragraph is extracted in plain text for parsing. The first step of parsing is sentence segmentation. The second step is to identify the sentences containing “claim” or “claims” and the associated claim number(s). A text span like “claims 2-5” will be expanded as “claims 2, 3, 4, 5” for downstream steps. The third step is to identify prior art references as an entity recognition task. This step involves leveraging spaCy’s default named entity recognizer, filtering by rule-based matching, and making use of the “Notice of References Cited” file provided by the USPTO. spaCy [2] is a free open-source library for Natural Language Processing in Python. Without training a domain-specific recognizer for office actions, the default recognizer in spaCy using the “en_core_web_sm” model will generate false-positive results. Hence, an additional rule-based matching step is

Final	Original	06/22/2023								
	1	✓ Ri								
	2	✓ Ri								
	3	✓ Ri								
	4	○ Ob								
	5	○ Ob								
	6	✓ Ri								
	7	✓ Ri								
	8	✓ Ri								
	9	○ Ob								
	10	✓ Ri								
	11	○ Ob								
	12	○ Ob								
	13	○ Ob								
	14	○ Ob								
	15	○ Ob								
	16	○ Ob								
	17	✓ Ri								
	18	○ Ob								
	19	✓ Ri								
	20	○ Ob								
	21	○ Ob								
	22	○ Ob								
	23	○ Ob								
	24	○ Ob								
	25	○ Ob								
	26	○ Ob								
	27	= Aw								
	28	= Aw								

(a) using OpenCV + Tesseract

Index of Claims		Application/Control No. 17/859 868	Applicant(s)/Patent Under Reexamination Krahn et al.
		Examiner LISA LEA-EDMONDS	Art Unit 2847
☒ Rejected	☐ Cancelled	☐ Non-Elected	☐ Appeal
☒ Allowed	☐ Restricted	☐ Interference	☐ Objected

CLAIMS									
☐ Claims renumbered in the same order as presented by applicant									
Final	Original	06/22/2023							
	1	✓							
	2	✓							
	3	○							
	4	○							
	5	○							
	6	○							
	7	✓							
	8	✓							
	9	○							
	10	○							
	11	○							
	12	○							
	13	○							
	14	○							
	15	○							
	16	○							
	17	○							
	18	○							
	19	○							
	20	○							
	21	○							
	22	○							
	23	○							
	24	○							
	25	○							
	26	○							
	27	=							
	28	=							

(b) Google Document AI

Index of Claims		Application/Control No. 17/859 868	Applicant(s)/Patent Under Reexamination Krahn et al.
		Examiner LISA LEA-EDMONDS	Art Unit 2847
☒ Rejected	☐ Cancelled	☐ Non-Elected	☐ Appeal
☒ Allowed	☐ Restricted	☐ Interference	☐ Objected

CLAIMS									
☒ Claims renumbered in the same order as presented by applicant									
Final	Original	06/22/2023							
	1	✓							
	2	✓							
	3	○							
	4	○							
	5	○							
	6	○							
	7	✓							
	8	✓							
	9	○							
	10	○							
	11	○							
	12	○							
	13	○							
	14	○							
	15	○							
	16	○							
	17	○							
	18	○							
	19	○							
	20	○							
	21	○							
	22	○							
	23	○							
	24	○							
	25	○							
	26	○							
	27	=							
	28	=							

(c) Azure AI

Fig. 3: Benchmark 2

introduced to screen out entities that do not qualify as prior art references. A detail to consider in this step is that a prior art reference could appear in either the “Author” format or the “Author et al.” format. Both formats should be identified and treated as identical references when mapping claims and prior art references in the subsequent step.

To verify the correctness of the identified prior art references, two supplementary validation measures have been implemented: (1) Scrutinizing the contextual surroundings. For instance, it is frequently observed that a prior art reference “X” appears within sentences as “X teaches...” or “...in view of X...”. Additional surrounding text can be added for other instances when necessary in the

future. (2) Cross-referencing with the “Notice of References Cited” file, where the prior art references are listed. Up to this step, claim numbers and prior art references are identified. The next step involves revisiting all sentences to pinpoint those that contain both a claim number and a prior art reference. These identified sentences serve as anchor sentences, forming the basis for establishing connections between claims and prior art references. In typical office actions, it is customary to find the examiner’s opinions following an anchor sentence. For instance, an anchor sentence might serve as a section title for the subsequent paragraphs. As a proof of concept in this manuscript, the sentences situated between two anchor sentences are considered the examiner’s opinions. Taking US patent 10,452,978 as an example. Fig. 4 shows the result of the parser. The input file is the “Non-Final Rejection” issued on 02/21/2019 in XML format. The issue date of the “Notice of References Cited” is the same. The title of the patent is “Attention-based sequence transduction neural networks.” The patented invention is the novel neural network architecture Transformer in the “Attention Is All You Need” paper [18].

claim	Prior Art References				
	Luong	Cheng	Sukhbaatar	Lin	Daniluk
1	[81][96][98][101][107][127][129][130][132][149][156][180][181][189]	[81][96][98][101][107][127][129][130][132][149][156][180][181][189]	[130]	[180][189]	[189]
2	[130][132]	[130][132]	[130]		
3	[81][96]	[81][96]			
4	[81][98]	[81][98]			
5	[81][101][105][150]	[81][101][105][150]			
6	[81][105]	[81][105]			
7	[149][150]	[149][150]			
8	[149][156]	[149][156]			
9	[180][181][186][190]	[180][181][186][190]		[180][186][190]	
10	[189][190][199]	[189][190][199]		[189][190][199]	[189][199]
11	[189][199]	[189][199]		[189][199]	[189][199]
12	[180][186]	[180][186]		[180][186]	
13	[81][107][110]	[81][107][110]			
14	[81][110][112][118][140]	[81][110][112][118][140]			
15	[130][140]	[130][140]	[130]		
16	[81][112][116][162]	[81][112][116][162]			
17	[81][116]	[81][116]			
18	[149][162]	[149][162]			
19	[81][118][122][168][201]	[81][118][122][168][201]			
20	[189][201][216][218]	[189][201][216][218]		[189][216][218]	[189][216][218]
21	[189][216]	[189][216]		[189][216]	[189][216]
22	[189][218]	[189][218]		[189][218]	[189][218]
23	[149][168]	[149][168]			
24	[81][122][174][221]	[81][122][174][221]			
25	[189][221][236][238]	[189][221][236][238]		[189][236][238]	[189][236][238]
26	[189][236]	[189][236]		[189][236]	[189][236]
27	[189][238]	[189][238]		[189][238]	[189][238]
28	[149][174]	[149][174]			
29	[81][127]	[81][127]			
30	[81][129]	[81][129]			

Fig. 4: Claim v. Prior Art Reference

In Fig. 4, the first column shows the claim numbers of the patent. The subsequent columns show prior art references. Below these columns, the mapping shows the line number(s) indicating which sentence contains both the corresponding claim number and prior art reference. For example, regarding claim 1 and the first prior art reference Loung, “[101]” means that the 101th sentence in the input file is an anchor sentence. By clicking the line number, the result shows its following sentences until reaching the next anchor sentence, as displayed in Fig. 5. It is noted that claim 5 is also mentioned at line 101. Therefore, the hyperlink as “[101]” can be found in the intersection between claim 5 and the same prior art reference, as shown in Fig. 4. In brief, the parser facilitates easy navigation through claim numbers, prior art references, and patent examiner’s opinions.

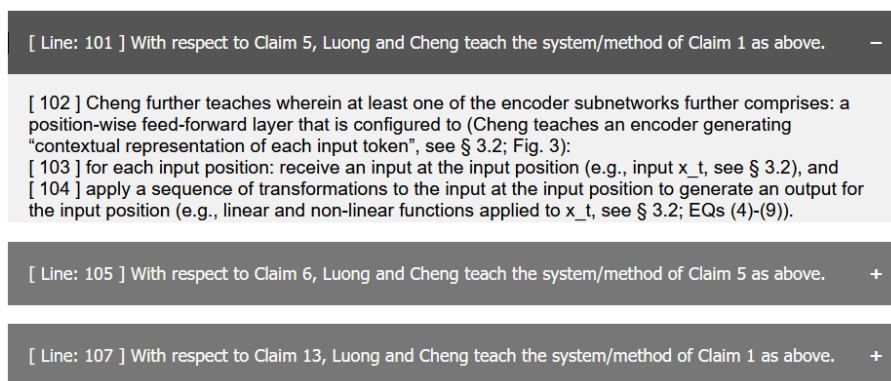


Fig. 5: Anchor Sentence 101

4 Discussion

4.1 Image Processing

The recent advancements in visual document understanding are notable, offering the capability to convert unstructured data from documents into structured data. This innovation addresses challenges previously deemed unsolvable. However, it’s important to note that in the patent domain, and likely in most other legal fields, structured data in documents is more prevalent than unstructured data. Traditional OCR and image processing tools and methods still hold significant value. What’s needed is algorithmic thinking to effectively leverage these existing tools for extracting content from structured data. In contexts like the patent field or other legal fields, the advanced OCR-free approach that combines pixel-based methods with neural network models may become excessively complex if model fine-tuning is required. Alternatively, without domain-specific

fine-tuning, this approach might lead to potential inaccuracies in certain edge cases, as the OCR-free model is designed to generalize accross a wide range of scenarios. The implementations detailed in this manuscript have demonstrated this concept. While the OCR-free and machine learning approach certainly holds value for handling complex structured data, traditional image processing and OCR techniques can still play a crucial role. They can simplify the structure by removing unnecessary elements and extracting the essential ones. Additionally, in situations where end-to-end models require further fine-tuning with substantial training data, these traditional methods can be employed for data labeling or data synthesis, thus minimizing manual effort and costs. In summary, a careful balance needs to be struck between the novel OCR-free approach and the conventional image processing methods.

4.2 Text Processing

Two specific improvements can be implemented in future research: (1) training a transformer-based name entity recognizer specific to office actions, (2) leveraging the hierarchical structure of office action files. Regarding (1), it is worth investigating whether the performance of the task of named entity recognition can be enhanced by training a state-of-the-art transformer model. The prior art references identified in this manuscript are based on the “en_core_web_sm” model in spaCy. The model is a small English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, syntax and entities. As described in section 3.2, the entities identified in this way may not correspond exactly to the prior art references and a rule-based filtering is required. While it is possible to expand the rule-based approach to accommodate a broader range of cases, training a transformer model is generally superior to manual feature engineering. Transformer-based models have demonstrated their state-of-the-art capabilities across numerous domains. When it comes to transformer model training, the labeled data produced through the rule-based approach can serve as the seed dataset. After training, the model becomes capable of labeling prior art references in new office action files to a certain extent. By verifying these new labels with manual corrections, additional accurately labeled data can be incorporated into the training dataset. As this dataset expands, the model can then be fine-tuned to predict labels in more office action files. Through this iterative curation, both the size of the training datasets and the accuracy of named entity recognition can progressively improve.

Regarding point (2) mentioned in the preceding paragraph, the hierarchical composition of office action files encompasses various elements, including sections, titles, paragraphs, sentences, patent law statutes, MPEP (Manual of Patent Examining Procedure) rules, text formatting (such as underlining or bold text), and the delineation of which part of a claim corresponds to a specific section of a prior art reference. As detailed in section 3.2, anchor sentences are determined using a rule-based approach. The sentences located between two anchor sentences are regarded as the examiner’s opinions. This rule-based approach may not cover all scenarios due to the flexible, unstructured nature of office action

file content. Hence, the rationale behind choosing a transformer-based approach in (1) holds true in this context as well. It is worth investigating the potential of training a new model to identify anchor sentences and examiner’s opinions. Likewise, the labeled data produced by the rule-based approach, in addition to curation, can serve as the seed dataset for training. Also, this rationale can be extended to encompass other hierarchical elements within office action files. It is noted that certain office action records may contain XML, DOCX, and PDF formats simultaneously, and various file formats can offer extra structural elements to utilize.

5 Looking Forward

This manuscript provides a new source at the data layer for implementing RLHF and instruction tuning in the future. RLHF is an iterative approach in which a model is trained to perform tasks using feedback provided by humans. A language model in the context of RLHF is to generate its output based on a reward model or function. Instruction tuning, which leverages a language model, aims to teach the model to understand and respond to human instructions effectively. In the patent field, issuing and responding to office actions is a back-and-forth process between patent examiners and patent applicants. During this iterative process, it is common for patent applicants to make revisions to their claims. Once all rejections and objections have been resolved, a notice of allowance is issued as an office action to inform that the claims in the patent application can be patented.

As patent prosecution involves an iterative process with substantial human feedback, this manuscript finds inspiration in OpenAI’s ChatGPT [9] and its predecessor InstructGPT [10]. These models have shown the effectiveness of RLHF and instruction tuning. ChatGPT’s success, in particular, highlights the potential of applying these approaches to the patent field using office action documents. The progress made in this manuscript will serve as a groundwork for future implementation. For example, from the perspective of human feedback, the various claim statuses, including “rejected”, “objected”, and “allowed” in office actions can be considered as labeled preferences in the reward model of RLHF. From the perspective of instruction tuning, the claims both before and after revisions, along with the patent examiner’s opinions in office actions and cited prior art references, can collectively form the training dataset for instruction tuning. This manuscript demonstrates the feasibility of extracting text from tables in office actions and identifying the connections between patent claims and prior art references. The techniques developed here will enable the inclusion of more human feedback and instructions in patent prosecution. Consequently, going forward, the training dataset can be substantially expanded in both scope and detail. By leveraging office action documents, generative language models in the patent field can effectively learn from the interactions and insights of patent examiners and applicants, utilizing approaches from RLHF and instruction tuning.

6 Conclusion

Office actions, which are intermediate documents in the patent prosecution process, contain an abundance of structured information that is often underutilized. This manuscript presents two examples demonstrating how data can be extracted from the structured content of office actions. The first example illustrates the extraction of data from image tables using image processing and OCR techniques. The second example focuses on identifying the relationships between patent claims and prior art references. The methods applied in the first example have potential applications in other legal fields, whereas the second example offers a relevant task for researchers specializing in the patent domain. This manuscript holds dual significance. Firstly, it demonstrates that structured data within image tables can be successfully recognized and extracted utilizing conventional OCR and image processing tools. In certain scenarios, this method can outperform recent OCR-free approaches in visual document understanding. Secondly, it highlights the potential to utilize a greater amount of data from patent prosecution processes. By capitalizing on this potential, there's an opportunity to create a more extensive training dataset in the future. This is essential for the development of more effective language models that are specifically designed for the patent field.

Acknowledgements The research reported in this manuscript has been funded by the National Science and Technology Council (NSTC) in Taiwan (Project ID: 112-2221-E-A49-117). Additionally, the author expresses deep gratitude to the Research Solutions GCP Credits Program (Google) for providing GPU resources. The contribution of these generous resources has made this research endeavor possible.

References

1. Tesseract open source ocr engine. <https://github.com/tesseract-ocr/tesseract>
2. ExplosionAI: spaCy. <https://spacy.io/>
3. Google: Document ai. <https://cloud.google.com/document-ai>
4. Lamichhane, P.: Pdf handling with python. <https://github.com/prajwollamichhane11/PDF-Handling-With-Python>
5. Lee, J.S.: Evaluating generative patent language models. *World Patent Information* **72**, 102173 (2023). <https://doi.org/https://doi.org/10.1016/j.wpi.2023.102173>, <https://www.sciencedirect.com/science/article/pii/S0172219023000030>
6. Lee, J.S.: Patentgpt-j-6b. <https://huggingface.co/patent/PatentGPT-J-6B/tree/main> (2023)
7. Lu, Q., Myers, A., Beliveau, S.: Uspto patent prosecution research data: Unlocking office action traits. <https://ssrn.com/abstract=3024621> (2017)
8. Microsoft: Document intelligence studio. <https://documentintelligence.ai.azure.com/studio>
9. OpenAI: Introducing ChatGPT. <https://openai.com/blog/chatgpt> (Nov 2022)
10. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L.,

- Simens, M., Askill, A., Welinder, P., Christiano, P.F., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) *Advances in Neural Information Processing Systems*. vol. 35, pp. 27730–27744. Curran Associates, Inc. (2022)
11. Suzgun, M., Melas-Kyriazi, L., Sarkar, S.K., Kominers, S.D., Shieber, S.M.: The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications (2022), <https://arxiv.org/abs/2207.04043>
 12. USPTO: Bulk data storage system. <https://bulkdata.uspto.gov/>
 13. USPTO: Office action research dataset for patents. <https://www.uspto.gov/ip-policy/economic-research/research-datasets/office-action-research-dataset-patents>
 14. USPTO: Patent Center. <https://patentcenter.uspto.gov>
 15. USPTO: Patent examination data system. <https://ped.uspto.gov/peds>
 16. USPTO: PatentsView. <http://www.patentsview.org/>
 17. USPTO: Research datasets. <https://www.uspto.gov/ip-policy/economic-research/research-datasets>
 18. 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 the 31st International Conference on Neural Information Processing Systems*. p. 6000–6010. NIPS’17, Curran Associates Inc., Red Hook, NY, USA (2017)

Balancing Exploration and Exploitation in LLM using Soft RLLF for Enhanced Negation Understanding

Ha-Thanh Nguyen and Ken Satoh

National Institute of Informatics, Tokyo, Japan
{nguyenhathanh,ksatoh}@nii.ac.jp

Abstract. Finetuning approaches in NLP often focus on exploitation rather than exploration, which may lead to suboptimal models. Given the vast search space of natural language, this limited exploration can restrict their performance in complex, high-stakes domains, where accurate negation understanding and logical reasoning abilities are crucial. To address this issue, we leverage Reinforcement Learning from Logical Feedback (RLLF) to create an effective balance between exploration and exploitation in LLMs. Our approach employs an appropriate benchmark dataset for training and evaluation, highlighting the importance of exploration in enhancing negation understanding capabilities. We compare the performance of our RLLF-enhanced LLMs with baseline models trained without RLLF, demonstrating the value of this balanced approach. Furthermore, we showcase the potential of our method in legal AI applications by employing transfer learning and evaluating its impact on negation understanding. Our experimental results exhibit the effectiveness of balancing exploration and exploitation with RLLF in improving LLMs' negation capabilities. This has implications for the development of more accurate, reliable, and logically consistent language models in high-stakes domains.

Keywords: LLM, Soft RLLF, Negation Understanding

1 Introduction

Negation, a fundamental aspect of communication in natural language, remains a challenging concept for state-of-the-art pre-trained language models like GPTs [15,1,14,13]. Outstanding in various tasks, these models often exhibit shortcomings when grappling with the intricacies of negation, an issue that becomes particularly pronounced in the context of legal AI, where precision in understanding is non-negotiable [9,7,3,8,6,12,10]. The complexity of negation requires a deep understanding and accurate computational treatment, which is critical in high-stakes domains such as law.

As the field of natural language processing (NLP) evolves, the treatment and comprehension of negation in textual data stand as a testament to a model's sophistication and practical application, especially in legal settings. Addressing

this, initiatives have surfaced to enhance language models’ capabilities through various means, including the augmentation of models with negation-focused datasets like MoNLI [3], CondaQA [16], Jina [4] and unlikelihood training objectives [8]. While these initiatives have shown progress in enhancing language models’ capabilities in handling negation, fully addressing the nuanced challenges presented by negation remains an ongoing task.

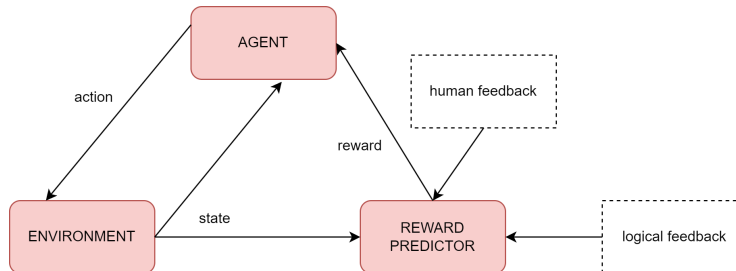


Fig. 1: RLLF is the idea of allowing feedback for reinforcement learning to come not only from the user but also from the accuracy in the chain of logical reasoning. [11]

Advances in language model development have seen the introduction of reinforcement learning from human feedback (RLHF) [14], which, while effective in aligning models with human intent, simultaneously exposes the limitations of human bias and the necessity for logical coherence [5]. In this context, the legal domain emerges as a pivotal use case that requires a blend of factual accuracy and stringent logical reasoning — competencies that are indispensable for reliable decision-making. In contrast to the reliance on human feedback in RLHF, Reinforcement Learning from Logical Feedback (RLLF) [11] allows the reward predictor to be trained using logical feedback in addition to human evaluations. This method aims to enhance LLMs’ logical reasoning capabilities while minimizing the impact of human biases.

In this paper, we propose and experiment with a method called Soft RLLF, which uses logical signals from natural language rather than logic programming languages. This choice is a practical decision for experimentation, as it considers the challenges associated with closed models that do not allow reinforcement learning and the high costs of working with models containing billions of parameters. Moreover, it takes into account the feasibility of working with smaller models in a laboratory setting, acknowledging that higher levels of logic programming capabilities may be more suited for larger commercial models. The experimental goal of this paper is not to create a state-of-the-art (SOTA) performance model but to demonstrate the effectiveness of the proposal and provide insights into its potential benefits. By using Soft RLLF, we aim to improve model performance through increased exploration of the vast search space of natural language.

2 Background

Generative Pre-trained Transformer 2 (GPT-2) is an advanced language model developed by OpenAI [15]. As the successor of GPT-1, GPT-2 marks a significant leap in language modeling by increasing the parameter count tenfold and enhancing the training dataset size proportionally [19]. Unlike previous models that employed recurrence-based or convolution-based architectures, GPT-2 utilizes a transformer-based approach which relies on attention mechanisms [18]. This allows the model to focus selectively on relevant segments of input text through effective parallelization, outperforming existing benchmarks. Despite its achievements, GPT-2 also exhibits limitations, particularly when generating longer texts, which can become repetitive and nonsensical [15].

Throughout the development of the GPTs, the automated understanding of language has evolved considerably. GPT-2 [15] demonstrated impressive zero-shot learning capabilities on language modeling tasks. GPT-3 [1], with its increased model size, achieved strong performance on numerous NLP tasks, including translation and question-answering. The GPT-3.5 model (a.k.a. ChatGPT) [14] introduced a method labeled as reinforcement learning from aggregate human feedback (RLHF) to better align the model with user intent. However, this approach has led to sacrificing the model’s logical abilities in exchange for user satisfaction.

Table 1 summarizes the performance of the GPT models on xNot360 dataset in the zero-shot setting. The *eXploring Negation Over Text with 360 samples* (xNot360) dataset [12] was developed to explicitly assess the negation detection abilities of language models, such as GPT-2, GPT-3, GPT-3.5, and GPT-4. The dataset was designed to take into account a wide range of sentence pairs containing diverse negation structures and various language contexts. By creating a more challenging benchmark for negation detection, xNot360 aims to further understand the limitations of current state-of-the-art pre-trained language models and offer valuable insights into potential improvements in natural language understanding.

In the table, we can see that GPT-4 outperforms the other models in both major metrics, achieving an accuracy of 0.7833, an F1-score of 0.7706, a precision of 0.8187, and a recall of 0.7278. On the other hand, GPT-3.5 shows a significant performance dip compared to its counterparts, with an accuracy of 0.4306, an F1-score of 0.2705, a precision of 0.3762, and a recall of 0.2111. GPT-3 exhibits moderate performance, with an accuracy of 0.6056, an F1-score of 0.6913, a precision of 0.5679, and a recall of 0.8833. Lastly, GPT-2 has the lowest accuracy (0.5000) but achieves the highest recall (1.0000). This demonstrates that the model classified all samples as positive, ensuring that no positive samples were omitted [12].

To provide a visual representation of the performance variations among the four GPT models, Figure 2 plots a performance chart that illustrates the fluctuating nature of their accuracy, F1-score, precision, and recall metrics. The chart exhibits a sinusoidal-like pattern, indicating the disparities in performance across the models.

Model	Accuracy	F1-score	Precision	Recall
GPT-2	0.5000	0.6667	0.5000	1.0000
GPT-3	0.6056	0.6913	0.5679	0.8833
GPT-3.5	0.4306	0.2705	0.3762	0.2111
GPT-4	0.7833	0.7706	0.8187	0.7278

Table 1: Performance comparison of GPT models on the xNot360 dataset.[12]

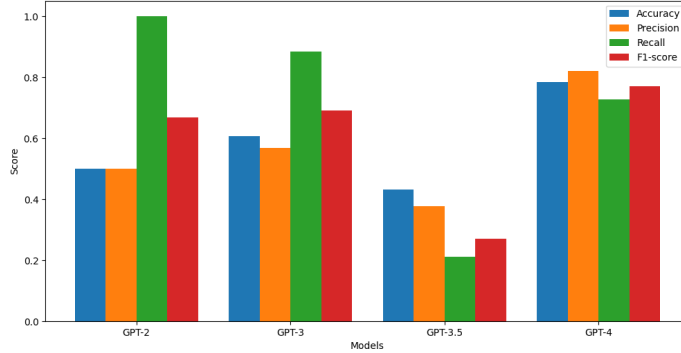


Fig. 2: Performance chart of GPT models on the xNot360 dataset. The chart displays a sinusoidal-like pattern, highlighting the differences in performance among the models.[12]

As shown in Figure 2, the performance of the GPT models varies considerably. GPT-4 exhibits the highest performance across major metrics, while GPT-3.5 demonstrates a noticeable dip in performance. GPT-3, on the other hand, shows moderate performance and GPT-2 has the lowest accuracy but the highest recall. From GPT-3.5, RLHF is applied. This observation suggests that optimizing parameters to align with user preferences may compromise the model’s logical reasoning abilities, especially for models with limited size. With larger models like GPT-4, this issue is partially addressed, but there remains substantial room for improvement in terms of performance.

In high-stakes domains like law, and healthcare, prioritizing adherence to logical rules over satisfying all user requirements is deemed more important, incorporating logical feedback into the model would be necessary. This concept is the driving force behind reinforcement learning with logical feedback – an approach to refine the model’s performance by integrating explicit feedback on logical reasoning.

Reinforcement Learning from Logical Feedback (RLLF) [11] is an approach that focuses on addressing the limitations of Reinforcement Learning from Human Feedback (RLHF), which is susceptible to biases introduced by human feedback and the need for complex logical reasoning, especially in high-stakes domains. By using logical feedback in addition to human evaluations, the RLLF

method aims to enhance the logical reasoning capabilities of language models, striking a balance between ensuring both user satisfaction and logical accuracy.

By employing the RLLF framework, language models can enhance their logical reasoning capabilities while minimizing the influence of human biases and subjective feedback. This innovative approach is particularly suitable for logic-intensive domains, where the ability to reason logically and provide accurate information is essential for practical applications and decision-making processes. The logic verifier can be adapted to different logical frameworks and languages, making the RLLF approach versatile and applicable to a wide range of scenarios.

3 Methodology

In this section, we present our approach to enhancing LLM’s negation understanding capabilities by employing *Soft* RLLF, focusing on the balance between exploration and exploitation. The methodology consists of three steps as demonstrated in Figure 3, including *Training Reward Model*, *RLLF* and *Transfer Learning*.

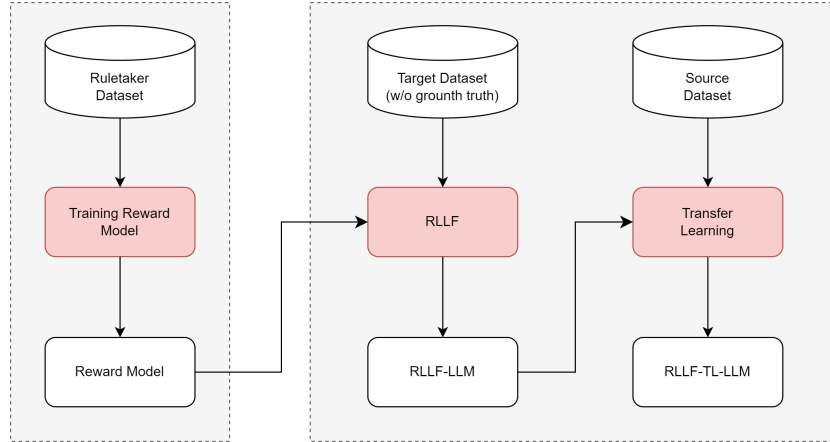


Fig. 3: Overview of the Reinforcement Learning from Logical Feedback (RLLF) methodology used to improve LLM’s negation understanding capabilities, highlighting the exploration-exploitation balance and key steps involved.

Our approach employs RLLF as a means to supplement LLM’s exploration ability. In the context of machine learning, exploration and exploitation refer to two fundamental strategies that a model can use to interact with its environment and learn from data. Exploration involves observing new, unseen samples from the environment, allowing the model to expand its knowledge about various possible scenarios and potentially improve its decision-making process. A higher

degree of exploration can lead to improved generalizability of the model as it helps the model understand more diverse situations and contexts.

With RLLF, we aim to improve the model’s negation understanding by encouraging it to explore a broader range of negation possibilities during training. The key idea is to generate negated sentences from LLM itself and verify the quality of these generated sentences through a reward model. While exploitation is necessary for ensuring that the model can make accurate predictions and decisions, relying solely on exploitation can lead to overfitting and reduced generalizability. Striking an appropriate balance between exploration and exploitation is essential for enhancing the overall performance, robustness, and generalizability of the model.

3.1 Training Reward Model

The first step in our methodology involves training a reward model using a large dataset centered around logical reasoning. To achieve this, we employ supervised learning by feeding the model a logically-focused dataset containing a broad range of sentence structures and logical contexts. Let the dataset be represented as $D = \{(x_i, y_i)\}_{i=1}^N$, where x_i is the input sentence, $y_i \in \{0, 1\}$ is the binary label indicating the logical relationship, and N is the total number of instances in the dataset.

The objective of the training process is to learn a reward model R that maps an input sentence x_i to its corresponding logical label y_i . In other words, we want to find the optimal model parameters θ^* that minimize the loss function L :

$$\theta^* = \arg \min_{\theta} L(\theta, D) = \arg \min_{\theta} \sum_{i=1}^N l(R_{\theta}(x_i), y_i), \quad (1)$$

where l is the cross-entropy loss representing the divergence between the model’s predictions and the true labels. To optimize the model parameters θ , we can employ gradient-based optimization techniques, such as stochastic gradient descent (SGD) or Adam.

3.2 Reinforcement Learning on Logical Feedback

We apply Soft RLLF to the LLM on the target dataset using the reward model as the logical verifier. Soft RLLF, in this context, is an adaptation of the RLLF approach that utilizes natural language signals instead of logic programming languages to provide logical feedback during the training process. For each sentence in the target dataset, the LLM generates a negated sentence without knowledge of the ground truth label, and the reward model evaluates the generated negated sentence based on its confidence level of negation with the original sentence:

$$R(S_i, NS_i) = L(S_i, NS_i),$$

where R is the reward signal, S_i is the original sentence from the target dataset, NS_i is the negated sentence generated by the LLM, and L is the logical evaluation score provided by the Reward Model on a scale of 0 to 1, with 1 being the highest and 0 being the lowest.

The LLM is updated using the reward signal from the Reward Model, optimizing its parameter weights to maximize the expected reward:

$$\theta_{t+1} = \theta_t + \alpha R(S_i, NS_i) \nabla_{\theta} \log P_{\theta}(NS_i | S_i),$$

where θ_t and θ_{t+1} denote the model parameters at time t and $t + 1$, respectively, α is the learning rate, and P_{θ} is the conditional probability given by the LLM for the generated negated sentence NS_i given the original sentence S_i .

By following these steps, the LLM incorporates exploration during its training process by generating and evaluating a diverse range of negated sentences, ultimately improving its negation understanding capabilities.

3.3 Transfer Learning and Model Evaluation

To demonstrate the effectiveness of our RLLF-enhanced exploration approach for improving LLM’s negation understanding capabilities, we perform transfer learning using text classification with LLM as the backbone. We compare the results of two different settings:

1. **Without RLLF:** A baseline LLM model is trained using supervised training without RLLF enhancement.
2. **With RLLF:** LLM is trained using our RLLF-enhanced exploration approach as described earlier.

For each setting, we train a text classification model using a source dataset in the near domain and evaluate its performance on the target test set. We compare the accuracy, precision, recall, and F1-score for the two settings to assess the benefits of incorporating RLLF-enhanced exploration into the LLM model training process.

The comparison between the two settings allows us to observe how incorporating RLLF-enhanced exploration during the training stage affects the model’s performance when applied to a real-world domain, such as the legal domain. Our hypothesis is that the RLLF-enhanced LLM would exhibit a more accurate understanding of negation compared to the baseline model, as the RLLF approach specifically focuses on enhancing the model’s logical reasoning capabilities using a reward signal based on logic consistency.

By using transfer learning, we effectively leverage the pre-trained LLM models adapted to different training settings and extend their capabilities in a broader scope. The evaluation of the classification performance of these models, as measured by the accuracy, precision, recall, and F1-score, helps us quantify the effectiveness of the RLLF-enhanced exploration approach for improving negation understanding capabilities in comparison to traditional supervised training without RLLF.

4 Experiment

In this section, we describe the experimental setup to evaluate the benefits of incorporating RLLF-enhanced exploration into the LLM training process.

4.1 Model Choice

Given the different versions of GPT models and other LLMs such as LLaMA [17], Bard¹, Jurassic-2², and Claude³, choosing an appropriate model for experimentation is crucial. We opted for GPT-2 as the primary model in this study for several reasons. First, GPT-2 offers cost-efficiency compared to larger models with billions of parameters, making it more accessible for research purposes. Second, GPT-2 allows for considerable customization, permitting researchers to finetune, adapt, and modify the model to fit their experimental requirements.

Basing our experiments on GPT-2 enables us to draw conclusions from a myriad of experiments across different scales and settings. While the findings based on GPT-2 may not directly mirror the results from more advanced models or the largest LLMs, the insights gleaned from our study can provide valuable guidance for further research. By examining the strengths, weaknesses, and opportunities for improvement in negation understanding abilities in GPT-2, we pave the way for future research on larger and more advanced LLMs.

Table 1 shows the performance of GPT-2 on the xNot360 dataset in a zero-shot setting. Although not as poor as GPT-3.5, which attempts to answer and achieves an accuracy below random guess, GPT-2 also demonstrates that the zero-shot setting is not sufficient for the model to provide useful information. Since xNot360 is a dataset designed for binary classification, if a model consistently selects one label, it would achieve 50% accuracy without needing any additional skills. However, these results serve as a good starting point for us to verify our hypothesis regarding the effectiveness of RLLF and transfer learning.

4.2 Datasets

Rulemaker Dataset [2]: The Rulemaker dataset is employed to train the reward model. Designed by AllenAI, this dataset aims to train transformers to reason over language or emulate reasoning, with a focus on chains of reasoning, including deductive inference in constrained settings. The dataset contains 480,000 training samples, 76,000 development samples, and 150,000 test samples. The Rulemaker dataset is designed to test reasoning over synthetically generated natural language sentences, bypassing the need for a formal representation.

The large size and diverse sentence structures and logical contexts of the Rulemaker dataset make it suitable for training a reward model focused on logical reasoning. By incorporating this dataset in the reward model training process,

¹ <https://bard.google.com/>

² <https://www.ai21.com/blog/introducing-j2>

³ <https://www.anthropic.com/index/introducing-claude>

we can enhance the logical reasoning capabilities of the GPT-2 model, with potential applications in high-stakes domains such as legal AI.

Jina Dataset [4]: The Jina dataset is used for performing transfer learning with 10,000 samples. As demonstrated by our analysis of the GPT-2 model, the zero-shot setting is insufficient for smaller LLMs like GPT-2 to be useful in negation detection problems. We use the Jina dataset as training data within the near domain of transfer learning.

This dataset consolidates various retrieval objectives, such as e-commerce search, duplicate detection, web retrieval, article retrieval for question-answering, and text classification. By reformulating this dataset into sentence pairs with 0 and 1 labels, we can indicate the presence or absence of negation in the meaning of the pair, respectively. This reformatted dataset enables us to analyze the performance of GPT-2 in the context of negation understanding through transfer learning.

xNot360 Dataset [12]: The xNot360 dataset serves two primary purposes: (1) evaluating the performance of GPT-2 in different settings, and (2) providing queries for the GPT-2 model during the RLLF process. The use of this dataset enables GPT-2 to explore negation in sentence pairs without being biased by any specific patterns. Generated negation sentences are then evaluated by the reward model, providing logical feedback to update the GPT-2 model.

The xNot360 dataset is designed to test the logical understanding of LLMs through the subtlety of natural language with 360 samples. In some cases, the word “not” does not carry a negation meaning. Instead, it requires a deeper understanding of the logical concepts underlying the language to correctly determine the presence or absence of negation in a given sentence.

Table 2 demonstrates the complexity of negation in natural language, showing an instance where the logical expressions of two related sentences differ.

A	B	$\neg A \vee B$	$A \vee \neg B$	$\neg A \vee B \neq A \vee \neg B$
T	T	T	T	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	F

Table 2: Truth table illustrating the complexity of negation in natural language. [12]

The use of the xNot360 dataset in our experiments enables a more effective evaluation of the GPT-2 model’s performance in negation detection and enhances our understanding of the capabilities of LLMs in this context. By employing this dataset, we can identify areas for improvement in the GPT-2 model’s under-

Step	Hyperparameter	Value
Train Reward Model	Learning Rate	2e-5
	Train Batch Size	48
	Eval Batch Size	48
	Num. Train Epochs	1
GPT in RLLF	Weight Decay	0.01
	Min_length	-1
	Top_k	0.0
	Top_p	1.0
PPO in RLLF	Learning Rate	1e-5
	Batch Size	16
Transfer Learning	Learning Rate	1e-5

Table 3: Hyperparameters for the experimental steps, determined through random search.

standing of negation and better assess the effectiveness of our RLLF-enhanced exploration and transfer learning approaches.

4.3 Experimental Settings

The experiment consists of the following steps:

1. **Train reward model - RulemakerBert:** Train the reward model using the Rulemaker dataset to serve as a logical verifier in the RLLF training process. We use bert-base-cased as the core model for this item. With a dataset of 480,000 training samples, we found that 1 epoch is enough for the model to achieve good accuracy, 0.9312 is the performance on the test set.
2. **RLLF training - GPT-2-RLLF:** Perform RLLF training on GPT-2 using the xNot360 dataset as the target dataset and RulemakerBert as the logical verifier. This process stops when there is no improvement in the reward score. The resulting model, RLLF-GPT-2, incorporates the benefits of RLLF-enhanced exploration.
3. **Transfer learning - GPT-2-RLLF-TT:** Apply transfer learning to RLLF-GPT-2 using the Jina Negation Dataset as the source dataset, yielding the RLLF-TT-GPT-2 model. We use an early stop setting, the training process stops when there is no improvement on the evaluation set of the Jina dataset.
4. **Model evaluation:** Evaluate the performance of the final GPT-2-RLLF-TT model on the xNot360 dataset with ground truth. Compare accuracy, precision, recall, and F1-score with those of a baseline GPT-2 model trained using supervised learning without RLLF enhancement (GPT-2-TT) and GPT-2 with zero-shot setting (GPT-2-ZS).

In the training process, we use A100 GPUs. The batch size values mentioned for both training and evaluation parts take this hardware into account for efficient utilization. The chosen hyperparameters for the training processes, determined through random search, are summarized in Table 3.

4.4 Experimental Results

The accuracy rates of the three models are shown in Table 4, their confusion matrices are display in Figure 4.

The GPT-2 model in the zero-shot setting has an accuracy rate of 50%. As discussed earlier, GPT-2-ZS demonstrates a poor understanding of negation and always predicts the same label (i.e., no negation detected). The model’s performance is equivalent to random chance, providing unreliable results for enhancing the understanding of negation.

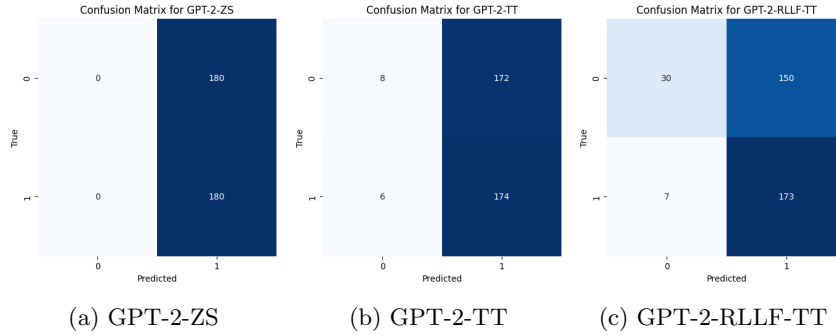


Fig. 4: Confusion matrices of GPT-2-ZS, GPT-2-TT, and GPT-2-RLLF-TT predictions on xNot360.

The GPT-2 model with transfer learning without RLLF has an accuracy rate of 50.56%. Although it exhibits a slight improvement over GPT-2-ZS, the performance of GPT-2-TT can be considered as marginally above random guessing. The inability to achieve dramatically improved accuracy highlights the limitations of solely relying on traditional supervised learning and not incorporating exploration-enhancing methods, like RLLF.

The GPT-2 model with RLLF-enhanced exploration and transfer learning attains an accuracy rate of 56.67%. As the best-performing model among the three, GPT-2-RLLF-TT shows the benefits of using RLLF and transfer learning for improving negation understanding. The improvement in accuracy, while significant, is not yet optimal, highlighting the need for further enhancements and exploration of the model’s abilities to address negation understanding in a more accurate and comprehensive manner.

The results show that the combination of RLLF-enhanced exploration and transfer learning in the GPT-2-RLLF-TT model can lead to substantial improvements in accuracy. This outcome underscores the importance of balancing exploration and exploitation in LLMs to enhance their negation understanding capabilities more effectively.

By adopting methods like RLLF-enhanced exploration and transfer learning, researchers can further improve LLMs’ performance, paving the way for the

Model	Accuracy
GPT2	0.5000
GPT2-TT	0.5056
GPT2-RLLF	0.5667

Table 4: Performance of GPT-2 with different settings on xNot360 dataset.

development of more accurate, robust, and logically consistent language models in high-stakes domains where accurate negation understanding is vital to decision-making processes.

5 Discussion

In this study, we demonstrate the potential of balancing exploration and exploitation in LLMs using RLLF to enhance their negation understanding capabilities. Our experimental findings show that incorporating RLLF-enhanced exploration and transfer learning into GPT-2 training leads to improved performance, as evidenced by the increased accuracy rate of the GPT-2-RLLF-TT model. This result highlights the benefits of using RLLF and transfer learning for promoting LLM’s negation understanding.

However, our research also exposes several limitations. First, a significant room for improvement remains in terms of accuracy. The GPT-2-RLLF-TT model exhibited a considerable improvement over GPT-2-ZS and GPT-2-TT but still remains suboptimal compared to state-of-the-art models. This gap may be addressed by further enhancing the models, adjusting the RLLF methodology, or exploring different LLMs. Additionally, the effectiveness of the Soft RLLF approach may vary depending on the domain and model size. Therefore, it is necessary to investigate its efficacy in other high-stakes domains and with larger models.

Moreover, one must consider the scalability of our approach when working with larger models like GPT-3 or GPT-4. Given the higher complexity and computational overhead associated with these models, developing cost-effective and efficient exploration-enhancement techniques becomes increasingly important. Further investigation and adaptation of our proposed approach to accommodate larger models would be necessary to maintain its effectiveness.

Despite the aforementioned limitations, our research provides valuable insights into the benefits of employing RLLF-enhanced exploration and transfer learning in LLMs. The improvements in negation understanding capabilities demonstrated by the GPT-2-RLLF-TT model have implications for the development of more accurate, reliable, and logically consistent language models in high-stakes domains that require precise negation understanding for robust decision-making processes.

6 Conclusion

In this paper, we presented a method for enhancing LLMs’ negation understanding capabilities by balancing exploration and exploitation using Soft RLLF. Our approach employed the Rulemaker dataset for training a reward model, which was integrated into the GPT-2 RLLF training process to encourage the model to explore a broader range of negation possibilities. We also utilized transfer learning and evaluated the impact of our approach on the xNot360 dataset in comparison to baseline models trained without RLLF. The experimental results demonstrate that incorporating RLLF-enhanced exploration and transfer learning leads to substantial improvements in LLMs’ negation understanding abilities. By enabling a more accurate understanding of negation, our proposed approach has the potential to significantly impact high-stakes domains such as law, healthcare. Future research may further explore the limits of the Soft RLLF approach in different domains and with various model sizes, as well as investigate its scalability when working with larger and more complex models. Additionally, leveraging cost-effective and efficient exploration-enhancement techniques in tandem with Soft RLLF to optimize the performance of large-scale language models remains an area ripe for investigation.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4, as well as the AIP challenge funding in relation with JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Advances in neural information processing systems* **33**, 1877–1901 (2020)
2. Clark, P., Tafjord, O., Richardson, K.: Transformers as soft reasoners over language. In: Bessiere, C. (ed.) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. pp. 3882–3890. International Joint Conferences on Artificial Intelligence Organization (7 2020). <https://doi.org/10.24963/ijcai.2020/537>, <https://doi.org/10.24963/ijcai.2020/537>, main track
3. Geiger, A., Richardson, K., Potts, C.: Neural natural language inference models partially embed theories of lexical entailment and negation. In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. pp. 163–173 (2020)
4. Günther, M., Milliken, L., Geuter, J., Mastrapas, G., Wang, B., Xiao, H.: Jina embeddings: A novel set of high-performance sentence embedding models. *arXiv preprint arXiv:2307.11224* (2023)

5. Ha-Thanh, N., Wachara, F., Ken, S.: LogiLaw Dataset Towards Reinforcement Learning from Logical Feedback (RLLF), vol. 379. IOS Press (2023)
6. Hossain, M.M., Chinnappa, D., Blanco, E.: An analysis of negation in natural language understanding corpora. In: 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022. pp. 716–723. Association for Computational Linguistics (ACL) (2022)
7. Hossain, M.M., Kovatchev, V., Dutta, P., Kao, T., Wei, E., Blanco, E.: An analysis of natural language inference benchmarks through the lens of negation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 9106–9118 (2020)
8. Hosseini, A., Reddy, S., Bahdanau, D., Hjelm, R.D., Sordoni, A., Courville, A.: Understanding by understanding not: Modeling negation in language models. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1301–1312 (2021)
9. Kim, M.Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 283–289 (2019)
10. Nay, J.J., Karamardian, D., Lawsky, S.B., Tao, W., Bhat, M., Jain, R., Lee, A.T., Choi, J.H., Kasai, J.: Large language models as tax attorneys: A case study in legal capabilities emergence. arXiv preprint arXiv:2306.07075 (2023)
11. Nguyen, H.T., Fungwacharakorn, W., Satoh, K.: Enhancing logical reasoning in large language models to facilitate legal applications. arXiv preprint arXiv:2311.13095 (2023)
12. Nguyen, H.T., Goebel, R., Toni, F., Stathis, K., Satoh, K.: A negation detection assessment of gpts: analysis with the xnot360 dataset. arXiv preprint arXiv:2306.16638 (2023)
13. OpenAI: Gpt-4 technical report (2023)
14. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **35**, 27730–27744 (2022)
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
16. Ravichander, A., Gardner, M., Marasović, A.: Condaqa: A contrastive reading comprehension dataset for reasoning about negation. arXiv preprint arXiv:2211.00295 (2022)
17. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
19. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision. pp. 19–27 (2015)

How to Establish Legally Binding Smart Contracts on Loan and Credit Agreements

Sieh-Chuen Huang¹[0000-0003-3571-5236], Yun-Cheng Tsai²[0000-0002-6266-7260],
and Hsuan-Lei Shao^{3*}[0000-0002-7101-5272]

¹ College of Law, National Taiwan University, Taiwan

² Dept. of Technology Application and Human Resource Development, National Taiwan Normal University, Taiwan

³ Graduate Institute of Health and Biotechnology Law, Taipei Medical University, Taiwan
Corresponding Author: hlshao@tmu.edu.tw

Abstract. Researchers proposed the concepts of smart contracts and smart laws, initiating a discourse within the legal scholarship regarding the status of smart contracts as legally binding instruments. Despite ongoing debate, may a smart contract indeed be considered a valid legal contract, contingent upon its adherence to fundamental contractual elements such as offer, commitment, and consideration? Given the diverse range of applications and functions that smart contracts serve, it becomes imperative to scrutinize the intricate details of individual smart contracts to ascertain their legal standing—especially in considering compliance with local legal regulations. Therefore, this paper endeavors to conduct a meticulous, step-by-step analysis of the legal validity of a frequently encountered smart legal contract, specifically addressing its relevance to loan and pledge transactions within the framework of the Taiwanese legal system. By elucidating the legal significance of each constituent component of the smart contract, this approach endeavors to provide clarity regarding its legal efficacy. Ultimately, this methodology has the potential to facilitate the development of legally compliant smart legal contracts for the realm of e-commerce, thereby ensuring their adherence to prevailing legal standards.

Keywords: Smart Contract · Ethereum · e-commerce · Legal Binding · Local Regulations.

1 Introduction

Clark et al. (2016) defined smart contracts and smart laws [3]. Both are based on the definition by considering the operational aspects of the contract and the legal focus. A smart contract is an encoded computer program driven by a designated event. In other words, a Smart contract is an automatable, executable protocol and computer automation. However, some parts may require manual input and control. The tamper-proofing computer code, such as a replicated or shared ledger, guarantees the enforcement of rights and obligations.

However, some scholars oppose smart contracts as legal contracts because once both parties start smart contracts, the two parties are not allowed to add any rights other than those written in the code, which is different from a traditional contract under which a modification is always possible as long as all parties involved consent to the changes. And while traditional contracts set future performance obligations for one or both parties, smart contracts do not create future obligations because neither party is legally obligated to take action after entering the contract. Another extended question is some e-commerce contracts without considering compliance with local legal regulations. Consequently, contracts formed over the internet may encounter issues of enforceability or uncertain legal standing depending on the jurisdiction of the parties involved.

Prof. Morgan N. Temte (2019) mentioned that a smart contract is still legal if it meets the elements of an Anglo-American contract, such as offer, commitment, and consideration [10]. Prof. Farshad Ghodoosi (2021) explores three possibilities for a smart contract: non-contractual, unilateral, and offer-inducing [5]. First, smart contracts are not contracts because they lack the expression of human will. Second, there is no correspondence between an offer and an acceptance in a smart contract, but both parties issue an offer. If the intentions are consistent, the smart contract is established. The kind of smart contract is closer to the nature of a unilateral contract that requires the realization of intentions instead of promises. Surely, unilateral contracts exist in civil law countries as well as in common law countries. In civil law systems, We emphasize that an unilateral contract is recognized as a contract where one party makes a promise in exchange for an act by another party. Once the act is performed, the contract becomes binding on the party who made the promise. In a unilateral contract, the offering can only accept the offer by performance but cannot make a promise. Finally, an agreement to agree is also a framework to capture the nature of a smart contract. From this point of view, pre-contractual liability can be discussed even if the contract is not established due to a procedural error.

Speaking in Detail, pre-contractual liability in the context of smart contracts can arise in several ways, even if it might seem counter-intuitive given the automated and self-executing nature of smart contracts. Smart contracts are pieces of code that automatically execute the terms of a contract when predetermined conditions are met. However, the negotiation and agreement process leading up to the deployment of a smart contract can give rise to pre-contractual liability, especially in scenarios where traditional legal principles intersect with the decentralized and automated nature of blockchain technology. To ensure that the contract code precisely reflects the agreed-upon terms to mitigate the risk of pre-contractual liability. The feature of a smart contract is that it ensures the performance of the duties and saves the cost of execution without the intervention of a trusted third party such as a bank. Even if it is designed to exclude the authorities' assistance, when it comes to disputes between the parties, they may still seek remedies from the court.

Smart contracts have applications in various sectors, especially finance [1]. Since smart contracts are used on various objects with different functions, we

consider it essential to look into the details of a specific smart contract to determine if it is legally binding. Therefore, in this paper, we present an analysis of the legal applicability of a smart contract in a specific scenario: lending money in a loan agreement with collateral. The legal framework we apply to analyze the smart contract is based mainly on Taiwanese law. Clarifying the legal meaning of every individual part of a smart contract helps determine the contract’s legal effect at the end. This step will facilitate the development of smart contracts for e-commerce. On the one hand, the complexities of legal agreement and enforcement can be automated through smart contracts. On the other hand, it is necessary to guarantee the legal validity of a smart contract used in e-commerce.

The remaining sections of this paper can be shown as follows: Section 2 presents the research question and a flowchart of a smart contract for the collateralized loan agreement. Section 3 explains the legal meaning of every step of our smart contract in the scenario set by Section 2. Section 4 is our discussion. Section ?? concludes the paper with significant findings and the possible scope of future works.

2 Research Questions

Nick Szabo proposed the concept of smart contracts in 1997 [9]. He advocates that the assistance of programming languages implements many contracts with high homogeneity. However, the technical level at that time could only overcome the related problems once Vitalike Buterin proposed Ethereum, a decentralized open-source, public blockchain platform [2]. The concept of smart contracts converted an idea into a reality in 2016. Gavin Wood, Christian Reitwiessner, Alex Beregszaszi, Liana Husikyan, Yoichi Hirai, and several former Ethereum core contributors proposed Solidity in 2017 [4]. Solidity is a ”contract-oriented” programming language for writing smart contracts on various blockchain platforms. They realize smart contracts on Ethereum platforms.

As mentioned above, to promote the applications of smart contracts in e-commerce, we need to confirm that smart contracts subvert the general concept of traditional contract law and that smart contracts are clear enough to have legal force. This paper discusses if a smart contract for borrowing money can be implemented as a ”smart legal contract”. Of course, a legal contract is not limited to one that exists on a blockchain. Legal contracts have been a fundamental part of society long before the advent of blockchain technology. A legal contract is an agreement between two or more parties that is enforceable by law. It can be written, oral, or implied by conduct, and it must meet certain criteria to be considered valid, including offer, acceptance, consideration, and the intention to create legal relations. Some or all contractual obligations are defined by a computer program and executed automatically [8]. Figure 1 shows the process from the concept of smart contracts to the smart legal contract.



Fig. 1. The process from the concept of smart contracts to a smart legal contract.

2.1 Loan Agreement

In the real world, a loan agreement is a contract between a borrower and a lender. Like any contract, a loan agreement needs an "offer" and the "acceptance of the offer." The lender shall transfer the money or other fungible things to the borrower. And the borrower shall return items of the same quality and quantity or with additional interest.

Using a smart contract to establish a loan agreement and automate its enforcement in a virtual world is also possible. Previous literature has put effort into creating a loan platform based on smart contracts [13] and applying them in the real financial loan setting [12]. Instead of platform design, we will illustrate this smart contract, elaborate on its legal meanings, and highlight differences between smart contract designers' expectations and real legal effects. The next smart contract is based on this scenario: Lender A makes an offer by deploying a smart contract, setting borrower B and the amount to be lent (10 ETH), and transferring the amount to be lent into the contract. Borrower B issues a promise by interacting with the contract and accepting the conditions of the contract. Then the smart contract would automatically transfer the amount to be lent to borrower B. The following agreement is a no-interest and no-due-date loan to avoid complexity. Borrower B repays the loan by transferring the amount to the smart contract. Figure 2 shows how the scenario of a loan for consumption is implemented as a smart legal contract.

2.2 Pledge of Personal Property

The pledge is a type of security interest in which a lender (pledgee) takes possession of the property of a borrower (pledgor) to secure repayment for a debt or obligation. In Taiwan, a pledge may only apply to personal property. Suppose the secured obligations are not satisfied by the agreed time. In that case, the lender (pledgee) can sell the property. After that, the lender (pledgee) must account to the borrower (pledgor) for any surplus after payment of the secured obligations.

In the virtual world, using a smart contract to set up a pledge to receive a loan is also possible. Take the scenario above as an example. The borrower (B) owns a non-fungible token (NFT) containing references to a digital photo, which is assumed to be more and more valuable in the future. However, at this moment, B needs money (Ethereum (ETH) coin) to make a promising investment, which is predicted to recoup quickly. Under this circumstance, B would decide to borrow

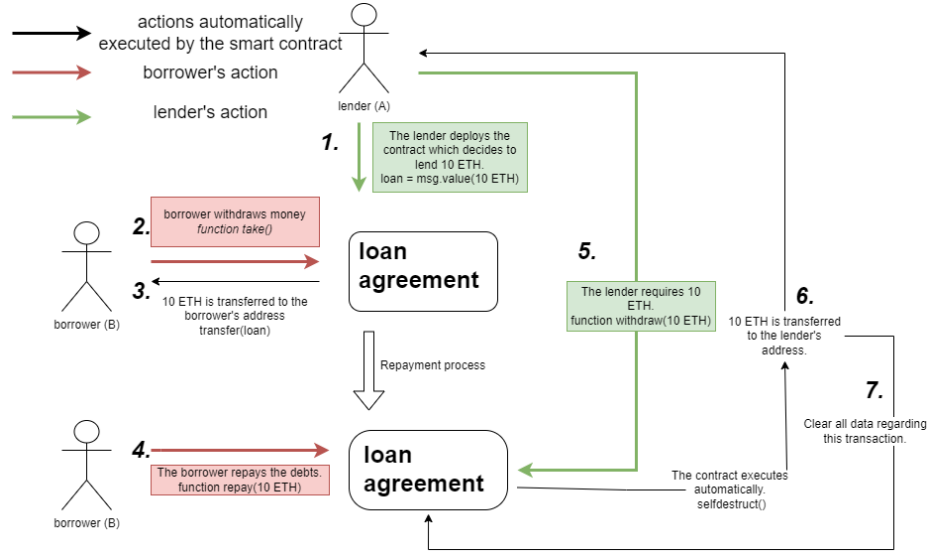


Fig. 2. A scenario of a loan agreement implemented as a smart legal contract.

ETH from others and put the NFT as a pledge instead of selling the NFT since B is likely to repay the loan due to the high chance to profit from the investment.

Figure 3 shows this contract is an interactive object for the borrower to accept ETH when repaying, which records the debtor, creditor, loan amount, loan duration, agreed repayment amount, end time, and other details. It also stipulates that when the borrower repays on time, its collateralized NFT will be returned, and the contract will be destroyed simultaneously. If the timeout expires, the collateralized NFT will be confiscated and transferred to the lender. The contract will be destroyed at the same time.

3 Implementations

This section will analyze the legal effect of every step of the smart loan contract described above to see if it is a smart "legal" contract.

3.1 Loan Agreement

Figure 2 shows the basic frame of a smart loan contract without collaterals. We divide the contract into 7 steps, further discussed in the following.

Step 1: The lender initiates a contract to decide how much to lend with amount = $10 * 10^{18}$ In the following code, the lender (A) deploys a smart contract and sets the borrower (B) and the amount ($10 \text{ ETH} = 10 \times 10^{18}$

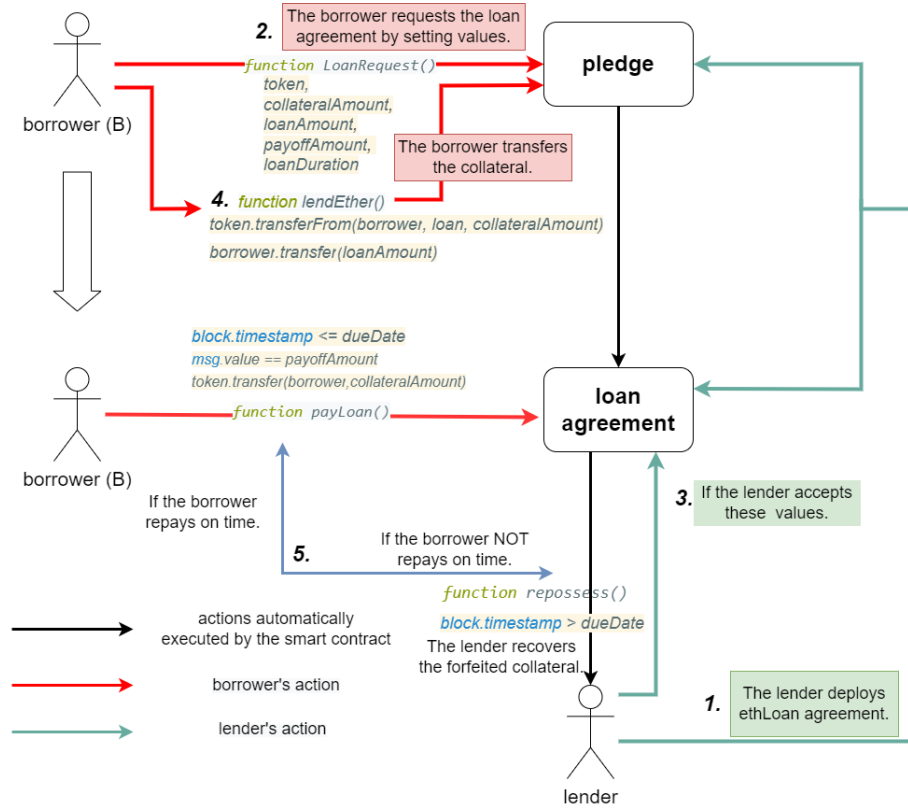


Fig. 3. A scenario of a pledge of personal property implemented as a smart legal contract.

Wei. Wei is the smallest denomination of ETH) to be lent. By doing so, the lender demonstrates his/her willingness to agree and invites the other party to agree by expressing assent. This is an offer in the law of contract as it can be expressed in many different ways, from a short and simple oral statement to a long and detailed written statement, as long as it is communicated and reasonable [11]. And, of course, an offer could also be a coding unit. An "offer" is a clear statement of the terms on which one party (the offeror) is willing to enter into a contract, with the intention that it will become binding upon acceptance by the other party (the offeree). Whether a coding unit, such as a smart contract or any other form of code, can be considered an offer depends on whether it meets the legal criteria for an offer. As the legal framework continues to evolve with technological advancements, specific legislation or case law may further clarify the status of coding units as offers. Such as our example. Therefore, the following code unit indicates the lender's offer, which expresses his/her willingness to enter into a contract of loan by setting the borrower's identity and the amount to be lent.

```

1 constructor(address A, address B) public payable{
2     lender = A;
3     borrower = B;
4     amount = 10 * 10 ** 18; //Step 1.
5 }
6 modifier inState(State _state){
7     require(state == _state,"Wrong state");
8     _;
9 }
10 modifier onlyBorrower{
11     require(msg.sender == borrower);
12     _;
13 }
14 modifier onlyOwner{
15     require(msg.sender == lender);
16     _;
17 }
18 //View contract balance
19 function watch() public view onlyOwner returns(uint256
    balances){
20     balances = address(this).balance;
21 }

```

Step 2: The borrower withdraws money by executing `tack()` In the following code, the borrower interacts with the contract and accepts the conditions of the contract. By doing so, the borrower assents to the terms of the offer from the lender. This could be considered an acceptance in the law of contract [7]. Similar to offers, "acceptance" in contract law refers to the unequivocal agreement to the terms of an offer by the offeree. When both offer and acceptance are aligned, a contract is formed, assuming all other conditions for a valid contract are met (such as consideration, capacity, and legality). In the digital realm, particularly with smart contracts, acceptance can occur through various forms of coding units or digital actions. Here are a few scenarios where a coding unit could constitute acceptance. So, it can also be expressed in many different ways, including but not limited to a coding unit.

Step 3: The contract transfers the loan amount to the borrower by executing `msg.sender.transfer(amount)` Next, the smart contract automatically transfers the amount lent to the borrower. This could be considered the transfer of rights in rem of that money. By doing so, the borrower now owns that money and has the right to use it, profit from it, dispose of it freely and exclude interference from others. At this moment, the contract of loan is legally established under the Taiwan Civil Code.

```

1 function take() public //Step 2.
2 onlyBorrower inState(State.Created) returns(bool){
3     msg.sender.transfer(amount); //Step 3.

```

```

4     emit Take();
5     state = State.Lending;
6     return true;
7 }

```

Step 4: The borrower repays by executing repay()

Step 5: The lender withdraws money by executing withdraw() **Discharge of a debt:** The next code unit shows the borrower repays the loan. Borrower B transfers the amount borrowed to the smart contract by executing repay(). It could be regarded as B's performance of his/her obligation, and it seems that B has been relieved from the agreement. But it may not be that simple from the perspective of law. The point is that the lender (A) has not withdrawn the money from the smart contract to his account. According to Article 309 of the Civil Code, the repayment obligation is extinguished when the loan has been repaid to the lender or his/her qualified representative and has been accepted. In our example, a question arises: why is B's debt being discharged? Is it the time B transferred the money to the smart contract or when A withdrew the money from the smart contract? Considering that the smart contract is under A's control, it would be reasonable to assume that when B successfully transfers the amount borrowed to the smart contract, it has been accepted by A. Thus, in our opinion, when the borrower transfers the amount to the smart contract, the debt is discharged and the obligation of borrower is extinguished.

Deadline for performance: Under Article 315, Article 316, and Article 478 of the Civil Code of R.O.C., if there is no deadline for repayment unless otherwise decided by the nature of the obligation or other situations, the borrower may repay the loan at any time. If there is a deadline, the borrower may repay the loan before it when the lender does not object. Since this smart contract has no clear due date, the borrower can repay the loan anytime. In addition, this no-due-date contract is written and announced by the lender, which could be assumed that the lender should understand the above-mentioned legal rules so that he cannot reject the loan repayment from the borrower at any time. Therefore, even if this smart contract has no clear deadline, it complies with the Civil Code of R.O.C.

```

1 function repay() public payable //Step 4.
2 inState(State.Lending) returns(bool){
3     require(msg.value == amount);
4     emit Payback();
5     state = State.End;
6     return true;
7 }

```

Step 6: The contract transfers the loan amount to the lender by executing selfdestruct(A)

Step 7: After the payment is completed, the contract is destroyed with `inState(State.End)` The contract is destroyed, meaning no one can borrow money from this smart contract. It shows that the lender is no longer willing to invite the other party to agree to a loan contract for consumption. This is a withdrawal of an offer in the law of contract. By executing self-destruct (`msg.sender`), the lender withdraws his/her offer from the public.

```

1 function withdraw() public onlyOwner //Step 5.
2 inState(State.End) returns(bool){ //Step 7.
3     selfdestruct(A); //Step 6.
4     return true;
5 }
6 } //End Contract

```

3.2 Pledge of Personal Property

According to Figure 3, the following smart contracts on the blockchain are implementations of smart legal contracts.

Step 1: The lender deploys the `ethLoan` contract In the following code, the lender (A) deploys a smart contract (contract `ethLoan`) so that any other party can interact with the contract and borrow money from the lender. By doing so, the lender demonstrates his/her willingness to invite the other party to make an offer. Note that this smart contract is slightly different from that in the above 3.1 as the lender's deployment should be viewed as an invitation to negotiate but is not an offer as there is no immediate intention to be legally bound. This is an invitation to treat the law of contract. Therefore, deploying an `ethLoan` contract indicates the lender's invitation to treat, which expresses his/her invitation to negotiate an offer of a loan with a pledge of personal property as a security.

```

1 //loan.sol
2 //SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.4;
4 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5
6 contract ethLoan { //Step 1. Start loan.sol ethLoan Contract.
7     address public lender;
8     address public borrower;
9     ERC721 public token;
10    uint256 public collateralAmount;
11    uint256 public payoffAmount;
12    uint256 public dueDate;
13
14    constructor(
15        address _lender,
16        address _borrower,
17        TestToken _token,
18        uint256 _collateralAmount,

```

```

19     uint256 _payoffAmount,
20     uint256 loanDuration
21 ) payable
22 {
23     lender = _lender;
24     borrower = _borrower;
25     token = _token;
26     collateralAmount = _collateralAmount;
27     payoffAmount = _payoffAmount;
28     dueDate = block.timestamp + loanDuration;
29 }

```

Step 2: The borrower requests the loan contract by setting values to public variables of LoanRequest In the following code, the borrower interacts with the contract and sets values, such as the amount to be borrowed (uint256 public loanAmount ETH), the loan duration (uint256 public loan duration), and the pledge (uint256 public collateralAmount). These values could be viewed as contract terms. By doing so, the borrower demonstrates their willingness to enter into a loan agreement with a pledge as a security and invites the lender to agree by expressing assent. This is an offer in the law of contract. Therefore, the following code unit indicates the borrower's offer, which expresses their willingness to enter into a contract by setting the contract terms.

```

1 //collateral.sol
2 //SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.4;
4
5 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
6 import "../loan.sol";
7
8 contract LoanRequest { //Step 2. start LoanRequest contract.
9     address public borrower = msg.sender;
10    ERC721 public token; //Step 2. the NFT.
11    uint256 public collateralAmount; //Step 2. the pledge.
12    uint256 public loanAmount;
13    uint256 public payoffAmount; //Step 2. the amount to be
        borrowed.
14    uint256 public loanDuration; //Step 2. the loan duration.
15
16    function loanRequest(
17        TestToken _token,
18        uint256 _collateralAmount,
19        uint256 _loanAmount,
20        uint256 _payoffAmount,
21        uint256 _loanDuration
22    ) public
23    {
24        token = _token;

```

```

25     collateralAmount = _collateralAmount;
26     loanAmount = _loanAmount;
27     payoffAmount = _payoffAmount;
28     loanDuration = _loanDuration;
29 }

```

From a legal viewpoint, establishing a pledge of personal property has three requirements. First is the existence of an obligation which could be a contract of loan for consumption. Second, both parties have to agree that to guarantee the performance of a specific claim, the debtor or a third party would transfer their personal property to the lender's possession. Last, the lender has to take control of the private property. In our smart contract, since the lender (A) and the borrower (B) have reached an agreement on the loan and the pledge at steps 2 and 3, at the moment the borrower approves the transfer of the NFT (ERC721 public token), the pledge is legally established under the law.

Step 3: The lender accepts these values from the borrower's setting and deploys a new Loan contract to enforce the loan terms. In the following code, the lender would deploy a new Loan contract with the condition set by the borrower. By doing so, the lender shows their consent to the terms of the offer from the borrower. This is an acceptance of the law of contract. [6].

3.3 Step 4. The borrower transfers the NFT collateral to the smart contract, and the loaned ether is sent to the borrower.

In the following code, the smart contract would automatically transfer the amount to be lent to the borrower. The lender has to approve the transfer of the loaned ether, which could be considered the transfer of rights in rem of that money. By doing so, the borrower now owns that ETH and can use it to make the intended investment. The loan contract for consumption is legally established under the Taiwan Civil Code.

```

1 //collateral.sol
2 ethLoan public loan;
3
4 event LoanRequestAccepted(address loan); //Step 3.
5
6 function lendEther() public payable {
7     require(msg.value == loanAmount);
8     loan = new ethLoan(
9         msg.sender,
10        borrower,
11        token,
12        collateralAmount,
13        payoffAmount,
14        loanDuration
15    );
16     address loan_addr = address(loan);

```

```

17     address payable addr = payable(loan_addr);
18
19     require(token.transferFrom(borrower, addr,
20                               collateralAmount)); //Step 4.
21     payable(borrower).transfer(loanAmount);
22     emit LoanRequestAccepted(addr);
23 }
24 } // End collateral.sol Contract

```

After the function "lendEther" , the borrower holds the loan ETH, and the loan contract has the NFT as collateral tokens. Due to public loan state variables, both parties can easily find loan contracts. What if the borrower fails to approve the NFT transfer? If the transfer is not endorsed, the function "transferFrom" will fail, and the entire transaction will be aborted, meaning the borrower will not lose their ether. What happens to its NFT if the transaction fails and the borrower doesn't get the ETH? Nothing happens to them. What happens if the lender never arrives and the borrower no longer wants to get the loan? The borrower can cancel the token transfer approval before the lender comes.

From a legal viewpoint, establishing a pledge of personal property has three requirements. First is the existence of an obligation which could be a contract of loan for consumption. Second, both parties must agree that to guarantee the performance of a specific claim, the debtor or a third party would transfer their personal property to the lender's possession. Last, the lender has to take control of the personal property. In our smart contract, since the lender (A) and the borrower (B) have reached an agreement on the loan and the pledge at steps 2 and 3, at the moment the borrower approves the transfer of the NFT, the pledge is legally established under the law.

After this function, the borrower holds the loan ETH, and the loan contract has the NFT as collateral tokens. Due to public loan state variables, both parties can easily find loan contracts. What if the borrower fails to approve the NFT transfer? If the transfer is not endorsed, function "transferFrom" will fail, and the entire transaction will be aborted, meaning the borrower will not lose their ether. What happens to its NFT if the transaction fails and the borrower doesn't get the ETH? Nothing happens to them. What happens if the lender never arrives and the borrower no longer wants to get the loan? The borrower can cancel the token transfer approval before the lender comes.

3.4 Step 5: The borrower repays on time or not

In the following code, the smart contract would allow the borrower to get their tokens back when the loan has been repaid within the loan period. The event "LoanPaid" signals the lender (A) that his ETH is available, which complies with the law. When the obligation secured by the pledge is extinguished, the pledge is also extinguished, and it must be returned to the pledgor (B). Usually, the pledgor is the borrower.

```

1 //loan.sol

```

```

2 event LoanPaid();
3
4 function payLoan() public payable {
5     require(block.timestamp <= dueDate);
6     require(msg.value == payoffAmount);
7
8     require(token.transfer(borrower, collateralAmount));
9     emit LoanPaid();
10    address payable target = payable(lender);
11    selfdestruct(target);
12 }
13
14 function repossess() public {
15     require(block.timestamp > dueDate);
16     require(token.transfer(lender, collateralAmount));
17     address payable target = payable(lender);
18     selfdestruct(target);
19 }
20 } //End loan.sol Contract

```

On the other hand, if the loan has not been repaid within the loan period, according to the following code of our smart contract, the lender is allowed to recover the forfeited collateral. However, this does not cope with the law of pledge, which requires the surplus money from the sale, after deducting the loan amount, to be paid over by the pledgee to the pledgor. Take the rules in Taiwan Civil Code as examples. If the obligation of a pledge is not satisfied, a pledgee has three ways to do it. The default way is to sell the thing pledged by auction and receive payment from the sale proceeds. Or a pledgee may enter into a contract to acquire ownership of the pledge. Otherwise, they may dispose of the pledge by any means other than an auction. Since the lender (A) and the borrower (B) do not enter into another contract about acquiring the ownership of the pledge, nor do they apply to the court for auction, it seems to be unlawful for the lender (A) to recover the forfeited NFT.

4 Discussion and Conclusion

The application of smart contracts in consumer lending allows borrowers to transfer funds to a designated smart contract address for repayment instead of directly transferring funds to the lender's address. According to Article 309, Paragraph 1 of the Civil Code, the debt must be paid to the creditor or another authorized person following the original purpose of the debt. The smart contract design ensures that the lender has control over the funds in the contract, which can be withdrawn at their discretion.

In cases of unclaimed surplus resulting from the sale of a pledge, the surplus is treated as a trading profit once the borrower's right to recover the surplus has expired, as established in the case of *Jay's the Jewellers Ltd v CIR* [1947] 29TC274. The Consumer Credit Act 1974 governs the sale of unredeemed articles, stating that if the article is not redeemed before the end of the redemption

period, the pawnbroker is entitled to sell the paper and use the net sale proceeds to pay any amount owed. If the redemption period is six months or less and the credit secured does not exceed £75, the property in the paper passes to the pawnbroker. However, the borrower has the right to recover the surplus within six years after the sale. Alternatively, suppose the practice treats the surplus on a sale as arising on the date of purchase and any recovery by the borrower as an expense arising on the date of payment. In that case, it may be continued, provided that this basis is consistently applied.

Using smart contracts in e-commerce has become an increasingly popular topic in the legal field. Even though smart contracts are not recognized as having the right to receive, it can be argued that the creditor has control over them, as they can withdraw money from the smart contract anytime. When money is transferred into the smart contract, the debtor's intention to repay is considered to have reached the creditor and is under their control. While there is a Supreme Court Civil Judgment in Taishang Zi No. 976 in 105 that establishes the debtor's account in the creditor's bank as a consumer trust relationship, meaning that the money in the account should not be considered as the content of the creditor bank's repayment, this does not directly apply to smart contracts. In the case of smart contracts, it is clear that the smart contract address is used for loan repayment, and remitting money to the smart contract is considered a refund.

As our research motivation, some e-commercial contracts do not consider compliance with local legal regulations, which can encounter issues of enforceability or uncertain legal standing depending on the jurisdiction of the parties involved

In contrast, this study adopts a divergent approach by emphasizing the programming of contracts based on the existing legal framework in Taiwan, effectively transforming them into electronic contracts. In other words, this research operates from a "legal-domain" perspective and practical implementation.

In perspective of legal validity, this paper underscores the necessity for developers to possess a sound understanding of legal principles and collaborate closely with legal professionals when crafting e-commerce contracts (or any other types of contracts). Only through this interdisciplinary collaboration can the efficacy of such contracts be ensured.

Furthermore, when dealing with cross-border or international transactions, e-commerce contracts will necessitate a deeper understanding of international private law and the application of legal norms in different jurisdictions. This highlights the importance of considering the global legal landscape in the development and deployment of smart legal contracts, particularly in the context of e-commerce.

This research will make a valuable contribution to developing digital legal acts for smart legal contracts, which can enhance the functioning of e-commerce. Moreover, it will serve as an encouragement for other researchers to create templates that are appropriate for legal contexts. In future studies, we aim to broaden the scope of smart legal contracts to include other digital business practices and improve the tools used for creating contracts.

References

1. Brown, R.G.: A simple model for smart contracts, 2015. URL: <https://gandal.me/2015/02/10/a-simple-model-for-smart-contracts/> (visited on 2022/11/28)
2. Buterin, V.: Ethereum: platform review. Opportunities and Challenges for Private and Consortium Blockchains (2016)
3. Clack, C.D., Bakshi, V.A., Braine, L.: Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771 (2016)
4. Dannen, C.: Introducing Ethereum and solidity, vol. 1. Springer (2017)
5. Ghodoosi, F.: Contracting in the age of smart contracts. Wash. L. Rev. **96**, 51 (2021)
6. Kadian-Baumeyer, K.: What Is Acceptance in Contract Law? - Definition, Rules & Examples. <https://study.com/academy/lesson/what-is-acceptance-in-contract-law-definition-rules-examples.html> (visited on 2022/11/28) (2021)
7. Pastrana, G.: What is Acceptance in Contract Law? <https://study.com/learn/lesson/what-is-acceptance-in-contract-law-rules-examples.html> (visited on 2022/11/28) (2022)
8. Rühl, G.: Smart (legal) contracts, or: Which (contract) law for smart contracts? In: Blockchain, law and governance, pp. 159–180. Springer (2021)
9. Szabo, N.: Formalizing and securing relationships on public networks. First monday (1997)
10. Temte, M.N.: Blockchain challenges traditional contract law: Just how smart are smart contracts. Wyo. L. Rev. **19**, 87 (2019)
11. UpCounsel Technologies, I.: What is an Offer in Law Of Contract: Everything You Need to Know. <https://www.upcounsel.com/what-is-an-offer-in-law-of-contract> (2022)
12. Wang, H., Guo, C., Cheng, S.: Loc—a new financial loan management system based on smart contracts. Future Generation Computer Systems **100**, 648–655 (2019)
13. Yang, Q., Xu, F., Zhang, Y., Liu, F., Hu, W., Liao, Q.: Design and implementation of a loan system based on smart contract. In: International Conference on Smart Blockchain. pp. 22–31. Springer (2018)

On Debugging Structural Legal Rules in Natural Languages using Large Language Models

Wachara Fungwacharakorn¹[0000–0001–9294–3118], May Myo Zin¹[0000–0003–1315–7704], Ha Thanh Nguyen¹[0000–0003–2794–7010], Hideaki Takeda¹[0000–0002–2909–7163], and Ken Satoh¹[0000–0002–9309–4602]

National Institute of Informatics, Tokyo, Japan
{wacharaf,maymyozin,nguyenhathanh,takeda,ksatoh}@nii.ac.jp

Abstract. Several approaches have been proposed to resolve counter-intuitive consequences in law. One approach is legal debugging, which is based on debugging logic programs that represent legal rules. This makes legal debugging difficult to understand for legal scholars, as they are more familiar with natural languages. Following recent debugging approaches that use large language models to ease automatic program repair, in this paper, we explore the debugging of structural legal rules in natural language. We propose assisting the traditional legal debugging approach with large language models. We demonstrate our approach using an example case related to the Japanese Civil Code Article 612, simplified from the real case decided by the Supreme Court of Japan. From the results, it shows that the proposed approach can prune unintended interpretations using traditional legal debugging, and explore new and potentially useful interpretations using large language models.

Keywords: Legal representation · Legal debugging · Large language models

1 Introduction

Since counterintuitive consequences may occur when literal interpretations of legal rules meet exceptional cases, several approaches have been used to formalize interpretative arguments from courts to resolve such counterintuitive consequences. Legal debugging [1] is one of such approaches based on debugging formalized legal rules. In legal debugging, a debugger follows the structure of legal rules to detect the problematic legal concept that intuitively causes counterintuitive consequences. Then, the debugger suggests legal rule revisions, such as adding an exceptional condition to the problematic legal concept, to resolve counterintuitive consequences. Legal debugging extends from algorithmic debugging [10] to guarantee correctness and completeness, that is, a problematic legal concept is eventually detected from a counterintuitive consequence if legal rules could be formalized as a stratified logic program.

However, formalizing legal rules as a logic program could be considered as one limitation of legal debugging, even though there are some projects that

aim to formalize legal rules, such as CATALA [5] or PROLEG [7]. One reason comes from the fact that most legal scholars are not familiar with logic programs. Although legal rules and logic programs have some similarities, such as that they commonly divide rules into consequence-antecedent-exception, legal scholars still prefer natural-language descriptions over logical formulations. Furthermore, they find that natural-language sentences prevail over responses from debuggers which are derived from logical formulation. For example, the sentence “the sublease contract was effective in the case” is better than the sentence “**effective-sublease-contract** is true in the case”. We can see that the first sentence is more natural than the second sentence, which is derived from the logic formulation.

The hardness of understanding responses from debuggers is a limitation not only of legal debugging but also of general program debugging in software engineering (e.g., type-error debugging [6]). To address this, explainable debugging approaches using large language models (LLMs) have emerged in software engineering. For example, automated scientific debugging [4] aims to prompt LLMs to detect and resolve bugs in programs, with Python as an example. The approach extends from scientific debugging [14], which divides the debugging approach into observation-hypothesis-experiment-conclusion steps as in the scientific method. Another example is RTLFixer [12], which aims to prompt LLMs to detect and resolve errors in RTL Verilog codes. The approach is based on ReAct prompting [13], which divides the explanations into thoughts-action-observation steps. These approaches leverage LLMs to propose and explain bug resolutions in natural language, reducing the burden of considering numerous possibilities. However, LLMs can sometimes generate inaccurate hypotheses or explanations.

In this paper, we propose a novel approach for debugging structural legal rules in natural language. We retain the structure-based nature of traditional legal debugging for correctness and completeness guarantees, but maintain the rules in natural language for scholar accessibility and LLM interaction. When a problematic legal concept is identified, the LLM is prompted to suggest interpretations for resolution. This approach allows us to capitalize on the strengths of both previous approaches: traditional debugging effectively prunes unintended interpretations, while LLMs excel at exploring new and potentially useful interpretations to resolve counterintuitive consequences. To evaluate the effectiveness of our approach, we conduct an experiment analyzing an example case based on the Japanese Civil Code Article 612, simplified from the real case decided by the Supreme Court of Japan. We report the comparison of the interpretations generated by CHATGPT (GPT-3.5) and GPT-4 under two conditions: with and without user interactions through traditional legal debugging approach.

The paper is structured as follows. Section 2 describes the example case and demonstrate legal debugging using the example case. Section 3 evaluates debugging structural legal rules in natural languages with only large language models. Section 4 describes our approach and evaluates the approach using the same example case. Section 5 discusses the potentials and limitations of our approach. Finally, Section 6 concludes the paper.

2 Example Case and Legal Debugging

In this paper, we examine an example case simplified from the Tokyo High Court Case 1994 (O) 693 [11] as follows.

Example 1. Suppose that an example case involves the following facts.

1. The plaintiff agreed to lease his room to the defendant.
2. The room was handed over to the defendant.
3. When the defendant returned home for a while, he agreed to sublease the room to his son, without the approval of the plaintiff.
4. The room was handed over to the defendant's son.
5. The son used the room.
6. The plaintiff found out, and he claimed that the contract was ended by his cancellation due to sublease.

The related legal rule is Japanese Civil Code Article 612, stating that

Phrase 1: A lessee may not assign the lessee's rights or sublease a leased thing without obtaining the approval of the lessor.

Phrase 2: If the lessee allows any third party to make use of or take profits from a leased thing in violation of the provisions of the preceding paragraph, the lessor may cancel the contract.

According to the literal interpretation of the article, the cancellation is valid because the defendant subleases the room without the approval of the lessor. However, the judges indicate that literal interpretation should not be intended as following statements.

Phrase 2 is not applicable in exceptional situations where the sublease does not harm the confidence between a lessee and a lessor, and therefore the lessor cannot cancel the contract unless they prove the lessee's destructing of confidence [11].

Now, we use the example case to demonstrate legal debugging [1], which is based on the principle that legal rules can be considered as combinations of legal concepts in the form of consequences, antecedents, and exceptions [8]. When a case presents to the court, judges need to find correspondences between legal concepts and facts in the case. The consequence is derived if all antecedents are provable and all exceptions are not provable. Following the principle, the legal rules could be formalized as follows.

Definition 1 (Rule). *Let R be a rule of the form:*

$$h : - b_1, \dots, b_m, \text{not } b_{m+1}, \dots, \text{not } b_n.$$

where $h, b_1, \dots, b_n$ are propositions and $: -$ could be read as if. We say h as a head of a rule or a consequence of a rule denoted by $\text{head}(R)$, $\{b_1, \dots, b_m\}$

as a positive body of a rule denoted by $\text{pos}(R)$ (each element of a positive body is called an antecedent), $\{b_{m+1}, \dots, b_n\}$ as a negative body of a rule denoted by $\text{neg}(R)$ (each element of a negative body is called an exception). We write $h.$ if the body of the rule is empty and we call such a rule a fact. We call a set of rules a normal logic program, or a program, in short.

not in a rule is a negation as failure, namely *not* p is derived if p cannot be derived. Negation as failure can be used for representing default assumptions, which the opposite can attack by proving their negations [8, 9]. Typically, legal rules would express this behavior with the terms “except” or “unless”, etc.

Example 2. Article 612 corresponds to the following rule [3].

- The contract was ended due to sublease if
1. The lease contract was effective.
 2. The sublease contract was effective.
 3. The third party was using the leased thing.
 4. The plaintiff manifested the cancellation of the lease contract.
 5. The sublease was not approved by the lessor.

Consequently, Article 612 and related rules about effective contracts can be represented as follows.

```
cancellation_due_to_sublease :-
    effective_lease_contract, effective_sublease_contract,
    using_leased_thing, manifesting_cancellation,
    not approval_of_sublease.
effective_lease_contract :-
    agreement_of_lease_contract, handover_to_lessee.
effective_sublease_contract :-
    agreement_of_sublease_contract, handover_to_sublessee.
approval_of_sublease :- approval_before_the_day.
```

In civil litigation, a judge would make correspondence between factual situations in a case and factual concepts in legal rules. Then, the judge would conclude a legal decision based on related legal rules. To reflect this civil litigation, we determine a proposition occurring in a head of a rule as a *rule proposition* and a proposition not occurring in a head of a rule as a *fact proposition*. By this determination, we denote a set of all fact propositions by $\mathcal{F}$ called a *fact-domain* and we denote all fact propositions occurring in a program T by $f(T)$ hence, $f(T) \subseteq \mathcal{F}$. We call a program RB a rule-base if RB has no propositions in $\mathcal{F}$ occurring in a head of a rule, and all propositions in RB that do not occur in a head of a rule, are in $\mathcal{F}$. When judges make correspondence between factual situations in a case and factual concepts, those factual concepts are represented by fact propositions. Hence, we call a set of facts (rules with empty bodies) constructed from a subset of $\mathcal{F}$ a *fact-base*, which represents a case.

Example 3. The fact-base representing the example case is as follows:

```

agreement_of_lease_contract. handover_to_lessee.
agreement_of_sublease_contract. handover_to_sublessee.
using_leased_thing. manifesting_cancellation.

```

An *answer set* is a set of propositions (including rule propositions and fact propositions) which can be concluded when compiling a fact-base FB with a rule-base RB (denoted by $FB \cup RB$). In this paper, we apply the stable model semantics [2] for finding an answer set. The answer set of $FB \cup RB$ represents the literal interpretation of legal rules (represented by RB) when applying in a particular case (represented by FB). Since we presume a non-recursive and stratified program, $FB \cup RB$ has a unique answer set [2]. This also reflects the principle that judges need one unique judgement from legal rules.

When a judge applies the literal interpretation of legal rules in a particular case and it leads to counterintuitive consequences, the judges may revise interpretation of legal rules. We call such a case an exceptional case. To distinguish the present case as an exceptional case, judges would introduce a new factual concept in the present case. Then, judges use the introduced factual concept to revise the legal rule so that the counterintuitive consequence is resolved. To formalize this process, legal debugging is a debugging approach that extends algorithmic debugging [10] to debug formalized legal rules. The approach considers a user as an oracle query of an unknown set of *intended interpretation* and the counterintuitive consequence is the symmetric difference between literal interpretation and intended interpretation. Legal debugging iterates to ask a user whether related consequences are counterintuitive until it can no longer find any counterintuitive consequences related. The last counterintuitive consequence found is called a *culprit*, which is defined as follows. The term *culprit* is used to distinguish from *bug* because this is an error in the interpretation rather than an error in the formalization.

Definition 2 (Culprit). *A proposition p is a culprit with respect to an intended interpretation IM and a program T if*

- $p \notin IM$ but there is a rule $R \in T$ that supports p with respect to IM (i.e. $pos(R) \subseteq IM$, $neg(R) \cap IM = \emptyset$, and $head(R) = p$), or
- $p \in IM$ but there is no such rule in T that supports p with respect to IM .

It was shown that if there is a counterintuitive consequence, we can eventually find a culprit using the culprit detection algorithm [1]. The algorithm works by interacting with the user to check whether the related proposition is counterintuitive, and the last found counterintuitive consequence is hence a culprit. Unlike general debugging, legal debugging concerns that the fact of the case cannot be counterintuitive, and hence it would not check the fact of the case with the user. Below shows an example dialogue between the debugger and the user (the texts after |: are the user responses).

```

Which consequence did you find counterintuitive in this case?
|: cancellation_due_to_sublease.

```

```

Considering the following rule:
cancellation_due_to_sublease:-
    effective_lease_contract, effective_sublease_contract,
    using_leased_thing, manifestation_cancellation,
    not approval_of_sublease.
effective_lease_contract is true in this case
Counterintuitive (yes/no) ? |: no.
effective_sublease_contract is true in this case
Counterintuitive (yes/no) ? |: no.
Detect a culprit: cancellation_due_to_sublease

```

3 Debugging using Large Language Models

In this section, we demonstrate one recent debugging approach using large language models (LLMs) called automated scientific debugging [4]. This approach aims to assist developers to understand why the program produces unexpected outcomes and how to resolve them using LLMs. The approach shows its strength to provide an understandable explanation to the user, based on scientific debugging approach [14], in which developers need to maintain debugging logs consisting of the following items.

- *Hypothesis*: a possible explanation of unexpected outcomes
- *Prediction*: an expected outcome if the hypothesis is correct
- *Experiment*: a means to verify the prediction
- *Observation*: the result of the experiment
- *Conclusion*: a judgement of the hypothesis, based on the observation.

Automated scientific debugging prompts LLMs to generate hypotheses and predictions, and then executes the experiment to command the compiler to verify the predictions. The paper [4] argues that automated scientific debugging is promising, as it offers competitive performance compared to existing methods, while also providing helpful explanations for its suggestions. These explanations are valuable to developers, improving their assessment of the patches and making them feel more comfortable with the results. The performance of automated scientific debugging also improves as the LLMs become more capable. However, the approach has several limitations, including that it requires multi-step interaction with the code, and the generated explanations may lead to inaccurate interpretations.

Now, we test using automated scientific debugging to formalize handling of exceptional situations from the same example case. We utilize CHATGPT (GPT-3.5) to generate hypotheses by inputting the following prompt. Subsequently, the ensuing response is received from the model, providing insights and potential outcomes. (<< >> marks a part of the question or the response that is omitted due to space limitation)

```

(Prompt)
Considering the case with the following facts and rules:

```

Facts:

<<The list of facts of the case as in Example 1 is put here.>>

Rules:

The contract was ended by the cancellation due to sublease if all of the following conditions are satisfied:

- 1) The lease contract was effective.
- 2) The sublease contract was effective.
- 3) The third party was using the leased thing.
- 4) The lessor manifested the cancellation of the contract.
- 5) The sublease was not approved by the lessor before the cancellation.

A lease contract was effective if all of the following conditions are satisfied:

- 1) A lessor agreed to lease a property to the lessee.
- 2) The property was handed over to the lessee.

A sublease contract was effective if all of the following conditions are satisfied:

- 1) A lessee agreed to sublease a property to the sublessee.
- 2) The property was handed over to the sublessee.

According to the given rules, your tasks are the following:

- 1) Evaluate that the contract was ended by the cancellation of the lessor due to sublease,
- 2) Generate hypotheses as to why the contract might not be ended by the cancellation of the lessor due to sublease in the case, and
- 3) Evaluate the plausibility of each hypothesis.

(ChatGPT Response)

Evaluation of Contract Cancellation due to Sublease:

According to the provided facts and rules, we can evaluate whether the contract was ended by the cancellation due to sublease:

Lease Contract Effectiveness:

Fact 1 indicates that the lessor (plaintiff) agreed to lease the room to the defendant. Fact 2 mentions that the room was handed over to the defendant. Therefore, the lease contract is effective.

Sublease Contract Effectiveness:

Fact 3 states that the defendant agreed to sublease the room to his son. Fact 4 indicates that the room was handed over to the defendant's son. Therefore, the sublease contract is effective.

Third Party Usage:

Fact 5 states that the son used the room. Therefore, the third party was using the leased property.

Plaintiff's Claim and Lack of Approval:

Fact 6 mentions that the plaintiff found out and claimed the cancellation

of the contract. Fact 3 indicates that the sublease was not approved by the plaintiff.

All conditions are satisfied, so based on the given rules, the contract appears to have been ended by the cancellation of the lessor due to sublease.

Hypotheses for Contract Not Being Ended:

Approval of Sublease:

Hypothesis: The lessor (plaintiff) may have implicitly or explicitly approved the sublease.

Plausibility: It's plausible if there is evidence or communication indicating the lessor's approval. Lack of explicit disapproval might also be considered implicit approval.

Awareness of Sublease:

Hypothesis: The lessor might not have been aware of the sublease.

Plausibility: If the lessor was not informed about the sublease, it could explain the lack of cancellation.

However, the lessor's awareness needs to be established.

<<The rest part of the response is omitted.>>

In the provided response, CHATGPT demonstrates the ability to align fact situations in the case with the antecedents of the rules. It adeptly follows rule structures and comprehensively considers all conditions stipulated in the rules. From several attempts that we tested, CHATGPT still deduces the consequence logically, e.g. if it says the antecedent is false, the consequence is not derived. We also conduct this test with GPT-4. Similarly, GPT-4 exhibits proficiency in aligning fact patterns with rule antecedents, adhering to rule structures, and diligently considering all specified conditions within the rules. In terms of hypothesis generation, it appears that both CHATGPT and GPT-4 formulate hypotheses based on the domain of knowledge rather than the structure of the rule. It is noteworthy that GPT-4 demonstrates a higher level of consistency in generating hypotheses. One observation is that both models generate approximately 3 to 6 hypotheses and occasionally assess all generated hypotheses as implausible. However, we can instruct the model in the prompt to generate more or specific numbers, such as 10 or 15, and request the inclusion of plausible hypotheses.

4 Assisting Legal Debugging with LLMs

In this section, we propose the use of large language models to assist with legal debugging. The proposed approach still uses traditional legal debugging as the main approach, but it prompts LLMs to generate plausible hypotheses after the culprit is detected. To operate traditional legal debugging on structural legal rules in natural languages, one option is to keep a mapping between formal representation and structural legal rules in natural languages. Another option is

to use only structural legal rules in natural languages and use LLMs to derive consequences, but derivations may need to be later checked by humans.

In the same manner as traditional legal debugging, if a user finds a counter-intuitive consequence in the literal interpretation, the debugger still traces each antecedent and exception of the rule that derives the consequence. If antecedents or exceptions of the rule are consequences of other sub-rules (i.e., rule propositions), the approach consults the literal interpretation, and checks with the user whether the antecedents and the exceptions of the sub-rule are counterintuitive. If the antecedents and the exceptions of the sub-rule are not counterintuitive, the consequences of the sub-rules are hence a culprit, and then we prompt the LLM to explore hypotheses to resolve the culprit.

The following dialogue demonstrates a prototype dialogue between the newly proposed debugger, the large language model, and the user. The first part of the dialogue interacts as in the traditional legal debugging approach. Once the culprit is detected, the debugger utilizes the fact-base, the detected culprit, and any user-provided interpretations to prompt the LLM for hypotheses for resolution. Symbolic propositions shown in square brackets could be omitted, but we show them to illustrate that the debugger can cooperate between propositional rules and structural legal rules in natural languages to determine the literal interpretation. (The texts after |: are the user responses, and << >> marks a part of the question or the response that are omitted due to space limitation)

```
Which consequence did you find counterintuitive in this case?
|: [cancellation_due_to_sublease]
The contract was ended by the cancellation due to sublease.
Considering the following rule:
[cancellation_due_to_sublease]
The contract was ended by the cancellation due to sublease if:
    1) [effective_lease_contract] The lease contract was effective.
    2) [effective_sublease_contract] The sublease contract was effective.
    3) [using_leased_thing] The third party was using the leased thing.
    4) [manifestation_cancellation]
    The plaintiff manifested the cancellation of the contract.
Unless:
    1) [approval_of_sublease] The sublease was approved by the lessor.

(Ask user) Do you think the lease contract was effective in the case
(yes/no) ? |: yes.
(Ask user) Do you think the sublease contract was effective in the case
(yes/no) ? |: yes.
Detect a culprit: [cancellation_due_to_sublease]
The contract was ended by the cancellation due to sublease.

(Prompting ChatGPT)
Considering the case with the following facts:
    <<The list of facts of the case as in Example 1 is put here.>>
And it is determined that
    1) "The lease contract was effective" is true in this case
```


- 2) "The sublease contract was effective" is true in this case
- 3) "The third party was using the leased thing" is true in this case
- 4) "The plaintiff manifested the cancellation of the contract" is true in this case.
- 5) "The sublease was approved by the lessor" is false in this case

Please generate hypotheses why "the contract was ended by the cancellation due to sublease" might be false in this case, and evaluate the plausibility of each hypothesis.

<<ChatGPT response here>>

In the proposed approach, a debugger can prompt the LLM with intended consequences obtained from the user interaction through traditional legal debugging. This aims to prune some unintended interpretations suggested from the LLM. We evaluated the hypothesis generation process by feeding the prompt generated through the user interaction into GPT-4 and analyzing the resulting hypotheses across five trials, as presented in Table 1. The results still show interpretations that are contradicted with the intended interpretation, e.g. hypothesis 3 is contradicted with the fact that the plaintiff manifested the cancellation but both statements could be considered as somehow different. This leads to clarification of the fact-base with the user. However, GPT-4 also generate a hypothesis *the son's use of the room did not materially affect the lease agreement.*, which is closely related to the exceptional situation of *non destruction of confidence* described in the real supreme court case.

Table 1. Examples of hypotheses generated from GPT-4 across five trials

No.	Hypotheses	Number of trial occurrences
1	The law does not support the cancellation of the contract in such situations.	5
2	The lease contract doesn't contain explicit prohibitions of subleasing.	4
3	The plaintiff didn't take legal action to enforce the cancellation.	4
4	The lease contract may contain a clause that permits subleasing.	2
5	The plaintiff discovered the sublease but took no action immediately upon discovery.	2
6	The son's use of the room did not materially affect the lease agreement.	2

5 Discussion

The preceding sections have delved into the application of large language models (LLMs) in the context of debugging structural legal rules in natural languages.

This discussion section aims to elucidate the key findings, implications, and challenges encountered during the exploration of this novel approach.

When using LLMs in legal debugging, striking a balance between exploration and exploitation is crucial. Exploration refers to investigating new and potentially useful interpretations of legal rules and cases, while exploitation involves leveraging knowledge and strategies that have proven effective. Traditional rule-based approaches often struggle with exploration, as they tend to lack the flexibility and creativity inherent in LLMs such as GPTs. In the context of legal debugging, the uncertainty and randomness of LLMs can facilitate exploration, helping legal experts identify and resolve problematic legal concepts and situations beyond the domain of rule-based systems.

Traditional legal debugging research mainly focuses on dissecting rule structures and their possible interpretations. However, fact-based debugging, which seeks to resolve ambiguities in the context of specific cases, offers a new direction for exploration. By incorporating natural language understanding and contextual reasoning, LLMs can handle the inherent uncertainty and multiple meanings often found in natural language expressions. For instance, let’s consider two separate examples. In the first example, the plaintiff invites the defendant to have ramen, to which the defendant replies, “I had ramen yesterday.” In this case, the meaning of the response is ambiguous, as it is unclear whether the defendant is accepting or declining the invitation. In the second example, within the context of a lease agreement, the defendant states, “I let my son live there temporarily while I was away.” This statement creates ambiguity regarding whether the defendant subleased the room to his son without authorization or not. In both examples, it is necessary to evaluate the surrounding context to determine the intentions behind these statements. By employing an LLM, we can consider multiple interpretations based on the context and explore the ambiguity more effectively. This highlights the importance of fact-based debugging in integrating context-driven information to facilitate a more effective legal debugging process.

6 Conclusion and Future Work

In this paper, we explore the debugging of structural legal rules in natural language. We propose our debugging approach based on the traditional legal debugging approach and the recent debugging approaches using large language models. With the proposed approach, the user can prune unintended interpretations using traditional legal debugging and explore new and potentially useful interpretations using large language models. However, large language models do not always guarantee intended interpretations. Future research could investigate in designing prompts and knowledge base integrations to achieve more just and intended legal interpretations, and could explore fact-based debugging to resolve ambiguities in fact-bases using large language models and user interactions.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Numbers, JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JP-MJCR20G4.

References

1. Fungwacharakorn, W., Satoh, K.: Legal debugging in propositional legal representation. In: JSAI International Symposium on Artificial Intelligence. pp. 146–159. Springer International Publishing, Cham (2018)
2. 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)
3. Ito, S.: Basis of Ultimate Facts. Yuhikaku (2001)
4. Kang, S., Chen, B., Yoo, S., Lou, J.G.: Explainable automated debugging via large language model-driven scientific debugging. arXiv preprint arXiv:2304.02195 (2023)
5. 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 (2021)
6. Ribeiro, F., de Macedo, J.N.C., Tsushima, K., Abreu, R., Saraiva, J.: Gpt-3-powered type error debugging: Investigating the use of large language models for code repair. In: Proceedings of the 16th ACM SIGPLAN International Conference on Software Language Engineering. pp. 111–124 (2023)
7. 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)
8. 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)
9. 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)
10. Shapiro, E.Y.: Algorithmic Program DeBugging. MIT Press, Cambridge, MA, USA (1983)
11. Tokyo High Court 1994 (O) 693: Case to seek removal of a building and surrender of lands (Oct 1996)
12. Tsai, Y., Liu, M., Ren, H.: Rtlfixer: Automatically fixing rtl syntax errors with large language models. arXiv preprint arXiv:2311.16543 (2023)
13. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022)
14. Zeller, A.: Why programs fail: a guide to systematic debugging. Elsevier (2009)

Mind the Gap - The Rules of the Road for Humans and Machines^{*}

Galileo Sartor¹, Adam Wyner², Giuseppe Contissa³, and Jacinto Dávila⁴

¹ Department of Computer Science, Swansea University, Swansea, United Kingdom
`galileo.sartor@swansea.ac.uk`

² Department of Computer Science, Swansea University, Swansea, United Kingdom
`a.z.wyner@swansea.ac.uk`

³ Department of Law, University of Bologna, Bologna, Italy
`giuseppe.contissa@unibo.it`

⁴ Centre for Modelling and Simulation, Universidad de Los Andes, Venezuela
`jacinto@ula.ve`

Abstract. In this paper, we model reasoning with respect to a fragment of the United Kingdom’s Highway Code (HC). The HC outlines legal requirements and good practices that learners and licensed drivers must see to in order to satisfy reasonable duty of care. As humans and automated vehicles (AVs) will interact on the roads for some time, we claim that an accessible, unitary, high-level computational model should be used by both types of users. A key justification for such a model is that liability should be consistently and uniformly borne by all road users. Relatedly, road use should be consonant with human expectations. As the model should be accessible to humans and executable by machine, *there should be no gap in a level of representation which serves as a model for humans and for machines*, even if there are differences in implementation or execution, e.g., how data is acquired and processed by humans and AVs. To fill the gap, we propose the use of Logical English, a natural language interface to Prolog. The computational model of road use, driver practice, and law complements sensors and data-driven reactive systems. The paper contributes a methodology and a computational model in Logical English to reason with respect to the HC, considering cognitive and contextual aspects of HC. It lays a basis for further unitary modelling.

Keywords: legal reasoning, natural language processing, autonomous vehicles

1 Introduction

Autonomous vehicles (AV) and humans are likely to interact on the roads for some time. Consequently, AVs and humans must both respect the rules of the

^{*} The authors wish to thank Prof. Bob Kowalski for leadership of the Logical English project. We also thank Prof. Giovanni Sartor for supporting this work in the context of the H2020 ERC Project “CompuLaw” (G.A. 833647).

road such as the United Kingdom’s Highway Code (HC), which outlines what is prescribed, good practices to fulfil duty of care, and assignment of liability. The assumption explored in this paper (cf. Section 3.2) is that the same high-level model of rules should be available to both humans and AVs such that they can (by and large) draw the same inferences, have related expectations, and behave similarly on a shared road. The paper does not consider isolated roads with only AVs, since that environment would enable different rules and considerations. Thus, we aim to provide one uniform representation of the rules of the HC such that machines can automatically execute them and humans can understand them and behave accordingly.

To reason with and execute the HC rules, we must provide the vehicle with a formal, computational model of (a fragment of) the HC along with an adequate evaluation strategy. The rationale for this focus on the HC, rather than case law and legislation, is that human drivers must explicitly learn and demonstrate competence with respect to the HC. Reasoning with case law and legislation is the purview of legal professionals. An AV’s greater data and processing capabilities does not obviate the requirement to abide by the HC. On the other hand, it is essential to characterise some aspects of a human driver’s cognitive capacity.

The goal is to model the rules in a unitary version, for AVs as well as human drivers. This is done by modelling the rules in a way that is isomorphic with the rules available in the HC for human drivers. While there may be the need to introduce additional rules, these should be kept to a minimum. In particular this might be needed to clarify the meaning of the rules in the HC, to make explicit something that is clear to humans, or to interact with the world (sensors and actuators). Having only one representation of the rules means that we can guarantee consistency. If a change were introduced in the natural language version, that would directly be available to the autonomous agents as well. Having multiple distinct versions would introduce additional burden in keeping the two in sync as well as potentially introducing inconsistencies.

This natural language representation could be queried by the AV while driving, or by users presenting a specific scenario, and returning a human understandable explanation of the reasoning and behaviour of the autonomous agent. By telling the system what the vehicle can perceive (the scenario, composed of other vehicles, traffic lights, intentions, and other relevant facts) we can replicate the reasoning of the agent. This may be useful to determine behaviour in risky scenarios (e.g., in case of accidents or rule violations).

How we deal with violations is not addressed fully in this paper and will be further explored in future development of the system.

We also focus on the issue of *liability and negligence* (See 3.2), where we claim that liability and negligence should be consistently and uniformly borne by all road users. Human drivers ought not to be disadvantaged due to an AVs greater perceptual and processing capacity. These points lead us to advocate a unitary model that is accessible to humans and executable by machine; that is, *there should no gap in a high-level representation which serves as a model for humans and for machines*, even if there are differences in implementation

or execution, e.g., how data is acquired and processed by humans and AVs. To represent this unitary model, we develop it in Logical English [1] (LE), which is a controlled natural language system which compiles to Prolog (see Section 3.3). It is a representation that is accessible to the human yet also executable by the AV; it should allow for accessibility, explainability, transparency, inference, and simulation. The executable LE is discussed in Section 5 and is available online for experimentation and inference.⁵

There are known challenging issues in representing and reasoning with legal rules: the very long, thin (or “gappy”) trail of data, highly parameterised alternative circumstances; context dependent, open textured concepts (e.g., *safe gap*); interpretation and ambiguity; models of objects (e.g., *long trucks*); mental models of other road users; prescriptive rules; and compliance checking. As a preliminary work, this paper scopes and addresses only some of these issues, leaving others for future work. Nonetheless, the paper presents initial directions for future work.

Part of the future work will also be dedicated to evaluation and validation of the system. At the moment this is not yet developed, but there are multiple possible paths we intend to explore, such as building a simulation environment, in which to apply the rulebase, and visualise the behaviour of the vehicles. With the agents running the modelled rules, it could be possible to introduce concepts from formal methods [2]. Violations of rules or unwanted outcomes could be identified and used to further enhance the rules. Extracting the logic based structure of the norms may already help in identifying potential inconsistencies in the existing text, and in combination with the simulation, help in clarifying useful concepts.

User studies could be carried out, to ensure that the rules and the explanations expressed in the controlled natural language are understandable and useful to the users, and that they ensure that the autonomous vehicles act in a predictable and understandable way with regards to the human drivers that share the road with them. An important factor when implementing autonomous agents in shared spaces is that humans have specific expectations on the behaviour of others and that the autonomous agent should respect those expectations as much as possible. In this paper we are not interested in determining whether AVs are safer than human drivers in general, but rather to determine the rules necessary for them to move in shared spaces with human agents.

In the following, we discuss related work in Section 2. The HC, liability, and Logical English are presented in Section 3. The requirements are presented in Section 4. The methodologies and results are presented in Section 5. Section 6 closes the paper with some reflections on limitations and future work.

2 Related Work

There has been recent work on AVs from both a legal/regulatory and technical point of view[3]. The UK Department for Transport reported in 2015 on how

⁵ <https://github.com/LyzardKing/mind-the-gap/tree/mindTheGap>

to integrate AVs in UK roads [4] and has since published a number of reports on testing and implementing AVs; it is currently working towards an *Automated Vehicles Bill*⁶.

Some research efforts focus on the use of higher order logic and temporal logic to express actions (e.g., overtaking) and concepts (e.g., safe distance), in order to monitor and respect traffic rules and highlight the issue of liability as described in Section 3.2 [5]. For example, the (R)ules (o)f (T)he (R)oad (A)dvisor (RoTRA) describes a formal specification in logic (and encoding in Prolog) of rules from the HC, defining what actions can or must be taken depending on the context, beliefs about the road situation, and the intentions of the vehicle [6].

There are also more data driven approaches. One problem is the lack of diverse and complete datasets [7]. Often the information contained in these datasets is limited to events that caused accidents. It is complicated to abstract them to a general set of rules, particularly to serve normative good behaviour. A second problem is that many data-driven approaches rely heavily on the automated interaction between autonomous vehicles [8]. This does not address the expectations of human agents on shared roads. Further research is ongoing on how to apply data-driven predictive systems to mixed traffic (human and AVs).

Not only would all users of the road work from shared data, but they would also need to work from shared inferences and understandings of road use. Otherwise, different agents could derive different, potentially conflicting inferences that possibly lead to an accident or different understandings of liability (cf. Section 3.2), which is relevant for insurance. In our view, these are good reasons for all participants on the road to have the same rules and expectations of other road users so as to successfully plan a course of action [9].

An alternative approach is to provide AVs with the capacity to identify in real time the road factors, carry out case-based reasoning, and argue over the case base [10]. While it might be feasible for an AV to reference and reason over a case-base in making a decision, it may be unrealistic to expect a human driver to do so with the same resources. For future discussion, we could explore just how humans reason about unusual situations, which may be a form of case-based reasoning, but over more abstract representations. With respect to liability in Section 3.2, it would be unreasonable to expect that a human driver would be negligent in not carrying out the reasoning that a machine can *ceteris paribus*. This is a point on which some functional parity between human and autonomous system might need to be found.

In AI and Law, it has long been key to represent legal provisions as code [11], [12]. There are a range of well-known issues. One is the presence of vagueness and open texture in the law [13]. [14] discusses the legal context related to open textured concepts. The same issues apply to the natural language version of the HC and have been discussed in the context of AVs with reference to natural language [15] and commonsense reasoning [16].

[17] presents an automatic compliance checking framework to assess AVs behaviour with respect to the traffic rules of Queensland, Australia. It consid-

⁶ Accessed 29 January 2024 <https://bills.parliament.uk/bills/3506>

ers issues related to open texture, exceptions, and conflicts amongst rules. The natural language rules are translated by a manual procedure into Defeasible Deontic Logic (DDL) along with some interpretation of the source rules. In this respect, [17] does not fulfil our aim of a unitary model. An automated compliance checker assesses the rules expressed in DDL to, amongst other matters, determine whether an AV’s behaviour in an environment is compliant with the rules. There is extensive evaluation. At this point, our work does not carry out compliance checking nor evaluation. In contrast to [17], we assume the driving environment has not been sterilised of human drivers, but rather includes them. Thus, human drivers and AVs must interact. Human understanding of and behaviour in the driving environment must be taken into consideration, which may go beyond the specification of the traffic rules (i.e., the interpretations) and require a unitary, transparent representation for both sorts of drivers.

In this research, the rules are expressed in Logical English (Section 3.3), which is a controlled natural language[18]; frames structure the language and enable unambiguous translation to Prolog. Other CNLs bridge the gap between logic formalisms, natural language, and execution, e.g., Attempto Controlled English[19], C&C/Boxer[20], and PENG-ASP[21]. However, these are either not under further development, currently unavailable, or are proprietary; none have they been applied to legal examples. LE is under active development, open source, and applied to legal examples.

The natural language version of the rules can be directly queried and can give answers in natural language. This could be used to test or explain the potential behaviour of the autonomous vehicle. The same Prolog code could be integrated in a vehicle for direct on-board use. Thus, the rules would be human readable and understandable, while still being executable by the autonomous agent.

3 Background

We focus on rules 170-183 of the UK Highway Code and, in Section 3.1, summarise the relevant heterogeneous aspects of the rules. There are rules that are neither prescriptive or legal, but relate to *duty of care*, which bears on the evaluation and allocation of *negligence* and *liability*, which is outlined in Section 3.2. The section ends with Section 3.3, which reviews LE.

3.1 Highway Code Rules on Junctions Rules 170-183

The HC has a section (Rules 170-183) on junctions; to save space, we briefly summarise these here⁷. There are a range of descriptions and instructions, for example:

- junction types - dual carriage ways, box junctions,
- road indicators - road markings, signs, and signals in junctions,
- driver behaviour - give way, remain behind, stop, wait,

⁷ See <https://www.highwaycodeuk.co.uk/road-junctions.html>

- rules - obligations and prohibitions
- open texture terms - safe gap, road users,
- anticipation of road users - predicting behaviour of other road users.

There are many other topics to consider, e.g., rule references, context dependency, temporal reasoning, and others.

This selection from HC presents a heterogeneous range of problems in terms of data, cognition, behaviour, and interpretation. The rules highlight that the behaviour of the vehicle is not solely derived from data on the circumstances (which may not exist or be too thin), as significant information is hypothetical and dependent on cognitive models of potential behaviours of others. All road users would need to share these models to cooperate on the road. Notably, several of the rules are not legal prescriptions in that they could be “violated” without incurring penalty. They are nonetheless relevant for tort law in the determination of negligence *duty of care*, which is further discussed in Section 3.2. Some rules, e.g., Rule 145 which prohibits driving on a pavement, are *defeasible*; there are circumstances where not only could be violated, but must be, e.g., to make way for police cars or ambulances on emergency calls. Moreover, these observations pose challenges to data-driven or ‘hard-coded’ (strict rules) approaches. Given the complexity of this small sample, we model a selection by way of example.

3.2 Liability and Negligence

Issues about *liability* appear throughout the legal system - legislation, regulation, Highway Code, tort law, legal services, and insurance. We focus on *negligence*.

Negligence is ‘the omission to do something which a reasonable man, guided upon those considerations which ordinarily regulate human affairs, would do, or doing something which a prudent and reasonable man would not do’ (Blythe v Birmingham Waterworks (1856) 11 Exch 781, at p 784).

The tort of negligence requires: an injury, harm or damage; the injurer owes a *duty of care* to the victim; the injurer has broken this duty of care (fault); the damage (or injury) is a reasonably relevant consequence of the injurer’s carelessness (causal connection)[22]. In the legal discourse, “negligence” denotes carelessness, neglect, or inattention, which are mental stances that can be ascribed only to human minds. A driver of the vehicle has an asymmetrical duty of care toward pedestrians or other individuals in the vicinity. The rules of 170-183 in part characterise how to meet this duty of care; broadly speaking, the driver should proceed defensively and cautiously, anticipating behaviours of others which might create circumstances in which the likelihood of an accident increases.

Clearly, the concept of negligence is linked to the idea of a human fault. In contrast, liability for technological failures is usually evaluated and allocated on the grounds of product liability, which requires evidence of the following:

an injury, harm or damage; a defective technology; and a causal connection between the damage (or injury) and the defect, namely that the former must be a reasonably relevant consequence of the latter. A technology may be considered defective if there is evidence of a design defect, a manufacturing defect, or a warning defect. Design defect, where the design is unreasonably unsafe, is the most relevant, and is usually determined by courts taking into account one of the following tests: the state of the art, the evidence of alternative design, or the reasonable expectations of users/consumers with regard to the function of the technology[23]. Key for our purposes is that negligence and product liability contrast with respect to duty-of-care and defective technology.

Yet, human and AI liability converge around performance: when a technology is used in substitution of a human, there is a reasonable expectation that the AI will perform an assigned task in a way that ensures the same level of safety that would be expected from a human performing the same task.

We reasonably assume that technology is presumed to have at least the same level of performance and safety as the human user. Thus, an AV ought to bear the same liability and duty of care as a human user. We propose that one code of conduct should rule both human and autonomous driver as a matter of fairness and equality on the road. This implies that we ought to be able to interrogate the automated vehicle on the same grounds as the human driver. To realise this, our modeling language should provide a unitary model, which yet allows for alternative means of realisation and data input.

3.3 Logical English

LE is a general purpose controlled natural language interface which translates to Prolog, which is used to efficiently and accessibly describe and model rules. An LE document consists of a knowledge base (facts and rules), scenarios, queries, and templates. The templates are declarations of the predicates contained in the knowledge base and scenarios, with the position of argument terms (e.g. variables) indicated using asterisks. The knowledge base is written in logical form with rules of the form *conclusion if conditions*. The knowledge base is converted into Prolog rules by a metainterpreter, and the program is evaluated by a Prolog reasoner to draw inferences. The solutions are then translated to English again to be shown as the answer or explanation.⁸ In LE, higher-order predicates can represent deontic modalities (obligation, prohibition, and permission) as well as propositional attitudes (beliefs or perceptual reports). Exceptions are expressed in LE with negation as failure, i.e. by stating the it is *not the case that* an exception applies to the rule.

LE satisfies three requirements. First, it enables ethical and legal requirements to be given a computable form which is understandable to domain experts without specific programming experience. This supports a high degree of isomorphism between the linguistically expressed norms and the program; domain experts can more readily translate and validate the requirements; drivers

⁸ For more details on Logical English and its legal applications see [1]

can understand the rules they are expected to follow, much as they can the source rules in HC. Second, we can precisely understand the computational meaning of the model and how it will be processed. Finally, a meta-interpreter provides understandable explanations by the proof trees. Together, LE provides a unitary model for humans and AVs about the rules of the road and common knowledge.

4 Requirements

In this section, we outline some of the requirements to represent a selection of HC Rules 170-183.

AI and Law Knowledge Modeling The features necessary to model the HC are a subset of those identified in general for legal knowledge modeling [24], [25], namely, machine and human readability, isomorphism, defeasibility, deontic operators, exceptions, reification, open textured concepts, and justification [26]. The knowledge base should be executable by machines, while also providing a clear human-readable justification/explanation. Modeling the HC requires representation of legal prescriptions and directives on behaviour so as to fulfill a duty of care.

Background and Tacit Knowledge Broadly speaking, human knowledge is often needed to interpret the rules and infer the missing parts based on common knowledge, presuppositions, and the specific context. It is not sufficient to simply transpose from the language of the HC into LE, as the formal model may require elaboration from the text, making tacit linguistic or practical knowledge explicit. That is, modeling may require explicit representations which are otherwise implicit and automatic for humans [27]. While modeled rules should be as isomorphic as possible to the source text [24], this is more a guideline, as the model may vary to address the complexities of the text, such as structure, cross-references, exceptions, temporal values, terms that vary in contexts, domain knowledge, or interpretation. In practice, different interpretations and augmentation needs to be included in the model and justified.

Open Textured Terms - safe gap Open textured terms are those where the provided “definitions” do not cover all possible instances of the term, as they vary with respect to circumstances or interpretation. To understand what the term means requires the introduction of background knowledge in the representation [9]. For example, the intuitive meaning of *safe gap* refers to the size of the gap which counts as a *safe gap* in a given circumstance. For example, in certain poor weather conditions (e.g., night-time, on icy roads, in heavy fog), the *safe gap* may be larger than in good weather conditions (e.g., daylight, on dry roads, with clear air). Moreover, the parameters themselves may be open textured; that is, the size of the safe space may be determined by the extent of road ice and visibility. In the HC, the concept of *safe gap* has some structure, although not every mention

is relevant to our study here on junctions.⁹ In Rule 170, there must be a *safe gap* for entering the junction. Yet, in terms of requirements on representation, the more general analysis is relative to circumstance. Open textured terms are treated as a constant in a main rule and are defined by subsidiary rules.

Temporal Reasoning In the HC, there are temporal restrictions on sequences of action (e.g., stopping at a stop sign before proceeding) and rules parameterised according to applicability (e.g., rules that apply on certain calendar dates and not on others). While a portion of this is introduced here, a full treatment of temporal reasoning remains for future work such as *relative time* of two moving vehicles. Temporal terms do not appear to be necessary in every rule.

Mental Models - long trucks and pedestrians Rule 170 references the driver’s understanding and expectation of the behaviour of other road users such as long trucks and pedestrians. In relation to trucks, the driver needs a mental model of the geometry of the truck and how it might move in the roadway. In relation to pedestrians, the mental model is epistemological in that the driver needs to interpret the behaviour of the pedestrian in the context and anticipate what it may know and intend to do in the near future; for example, whether the pedestrian is simply waiting beside the curb or about to cross the street; whether the pedestrian is aware of oncoming vehicles or not. Physical data is insufficient.

The Rules also reference the driver’s awareness or perception. While this makes sense in relation to the human driver, we make no claims about the meaning of such terms in an AV beyond assuming that such terms would be associated with sensor readings and analysis. We treat mental awareness terms as a high level, which are realised in different ways by different types of drivers. Keeping to our main claim, it is convenient in the representation to use one term to cover different realisations in order to provide a unitary model.

As with *safe gap*, we treat long trucks and pedestrians as constants in some main rule, then reason about it in subsidiary rules.

Deontic expressions - obligation, prohibition, permission Rule 171 about stop signs contains an obligation, which is linked to legally binding norms and penalties such as the Road Traffic Act 1988. In Rule 170, we find a prohibition on crossing the road without reference to legislation, while in Rule 176, we find a prohibition in relation to red lights with reference to the legislation. Some rules may be violated in specific, exceptional circumstances without incurring (alternatively lessening) penalty, e.g., a vehicle may move in the junction to make way for an emergency vehicle, or a vehicle may mount the pavement despite the prohibition against it. Such exceptions may involve also a temporal dimension. The treatment of permissions on an action in the HC is more complex and may not be explicitly, consistently expressed. Generally, the permission holds either

⁹ See HC on Speed limits, Rule 125, Rule 162, Rule 168, Rule 170, Rule 171, Rule 173, Rule 180, Rule 183, Rule 260.

where explicitly permitted so long as exceptions do not hold (strong permission), or it is an exception to an explicit prohibition (weak permission). A strong permission requires checking all possible exceptions before the action is taken, while a weak permission needs only to check the specific instance is an exception. For example, *When the road is clear and there are no pedestrians, then cross the intersection* is a strong permission; *Do not traverse a stoplight unless in the way of an emergency vehicle* is a weak permission.

It is worth noting that legal norms aim to regulate behaviour by introducing obligations and prohibitions along with the relevant sanctions where the norms are violated; we model a sample of these in Section 5. How agents behave with respect to the sanctions is guided by the aims and capacities of the agents, which is out of scope in this paper and remains for future work.

5 Methodology and Model

In this section, we present several sample models of HC expressions in LE along with methodological notes. Due to space restrictions, a sample inference is illustrated by way of an output explanation Figure 1; further inferences can be exercised using the implementation online.¹⁰

For a working example, we translate Rule 170 into LE: *Take extra care at junctions. You should [...] watch out for [other road users]* Note that an unstated primary aim is to avoid collisions with objects or persons; while on the road and moving, a safe gap (discussed below) is highly important (though not essential). Three general strategies are employed in the methodology:

1. The decision tree of rules chains from high level concepts which are defined in terms of lower level concepts or input data.
2. Implementation details at lower levels are specified by the agent.
3. The form of data or how it is collected is not specified, so the concepts and rules are applied given the relevant input.

From Requirements to Rules The first rule in our representation is:

Listing 1.1. Rule 170

```
A vehicle is permitted to enter a junction at a time if
the vehicle has watched the junction before the time
and it is not the case that
    there are users crossing the junction at the time
and there is a safe gap for the vehicle in the junction at the time.
```

This initial rule is high level, in keeping with our approach. A refined representation would refer to relative positions in time and space of the other agents as well as calculations based on the information coming from perceptions, sensors, cameras, oncoming vehicles, etc. Similarly, rather than a moment in time, we can add a temporal period, which we leave for future development.

¹⁰ See: <https://github.com/LyzardKing/mind-the-gap/tree/mindTheGap>

Safe Gap The determination of safe gap is anticipatory; that is the driver must determine the safety of the gap in the near future. While some of the parameters that are needed for the calculation are explicit in the HC, others must be filled in using commonsense human knowledge about e.g., variations depending on context. If there were only AVs on the road which communicated position and speed, this might not be at issue; but sharing the road with human drivers requires that AVs calculate movement to some extent based on the human driver’s point of view while the human driver needs to use commonsense with respect to the other vehicles (both human and AV). The AV may calculate a value using techniques familiar from robotics or multi-agent systems; for a human driver, “rules of thumb” may be explicitly taught or acquired over time, becoming tacit with experience. *A key point is not that AVs and human drivers process data and control behaviour in the same way, but rather that these are done in an explicit and comparable way with respect to a unitary model.* According to the HC, the vehicle could start with a *presumptive* safe gap, determined, among other factors, by the length/weight of the vehicle, its stopping distance, and some presumed context. It would then alter this depending on the specific contextual *dynamic* features. To an extent, such parameters and presumptions may be reasoned with by all vehicles. As this requires more detail than there is space here, a preliminary expression of *safe gap* in LE is available on the link.¹¹:

Mental Models Safe gaps are also determined with respect to models of objects, e.g., trucks, pedestrians, bicycles, and horses. We assume the driver recognises the entity and reasons, given the properties of the entity, and whether the junction is safe to enter. In this case the rule could be seen as a strict permission, which is more specific than appears in the source text.

In particular let us consider the first item in the list of Rule 170, which states that the vehicle should “watch out for cyclists, motorcyclists, and pedestrians [...]”. In this case we are dealing with an incomplete list of possible road users, some open textured terms (e.g., vulnerable road users), and the commonsense meaning of “watch out”. The rule could be made more general by not stating the permission/prohibition to enter the junction, but by instead having the vehicle slow down in certain risky scenarios, only for the vehicle to stop depending on the action of the other actors. In this case the previous rule could be rewritten as:

```
a vehicle has the obligation that
  the vehicle slows at the intersection
  if there are vulnerable road users close to the vehicle
    or there are vulnerable road users waiting to cross the
      intersection
    or there is an other vehicle in the intersection
    or there is the other vehicle on the right waiting to enter the
      intersection
  [...]
```

¹¹ See: https://github.com/LyzardKing/mind-the-gap/blob/mindTheGap/safe_gap.le

This is a general rule that would need to be refined with rules defining, in the circumstances, *vulnerable road user* and a *other vehicle* such as a truck. Moreover, the extent of slowing might be relative to the type of road user, for example, one might slow more around a young child close to the road than for another car that is signalling and moving. In addition, the extent of slowing would be relative to the conditions associated with safe gap.

Attention The original text does not state that the driver should or should not enter the junction, but rather that they should *pay attention* to certain events or agents. In our case, it becomes necessary to add an additional rule stating the desired goal - what the vehicle does. Modeling the norm as it is written would require introducing notions such as *take extra care* and *look all around*. For example, we could have the clause as earlier:

the vehicle has watched the junction before the time

What it means for an AV or a human driver to “watch the junction” may vary, but in our view, there ought to be a convergence to an extent in order to address concerns about duty of care and due diligence. In our analysis, “watching the junction” relates to the series of observations made in the evaluation of *safe gap* and articulated further with:

the vehicle has watched the junction before the time if
 the vehicle has watched the left of the junction and
 the vehicle has watched the right of the junction and....

Of course, as with other instances, an engineered analysis of what “counts as” having watched the junction for an AV may differ from the human psychological analysis. For example, in comparison to an AV, a human has more limited attention capacities, different sensors, and slower reaction times; we use common-sense to determine whether a human has “watched the junction”. Nonetheless, the evaluation and implementation is tied to some unitary model.

Deontic Rules Rules such as 171 and 172 express deontic rules. For example, we can model obligations in LE as follows:

a vehicle has the obligation that
 the vehicle stops at the intersection
 if the vehicle has a stop sign at the intersection.

a vehicle can enter the intersection
 if the vehicle has the obligation that
 the vehicle stops at the intersection
 and the vehicle stops at the intersection
 and there is a safe gap for the vehicle
 at the intersection.

Since these rules contain obligations (similarly for prohibitions), they are linked to other rules and regulations that determine the sanction for breaking such a rule. In this particular case, the driver will have a certain number of

```

E =
It is the case that: red car can enter intersection 1 at 2 as proved by KB Text
because
  • It is the case that: red car can enter intersection 1 at 2 for rule 2 as proved by KB Text
    because
      ◦ It is the case that: red car has the obligation that red car stops at intersection 1 at 1 as proved by KB Text
        because
          ▪ It is the case that: red car has a stop sign at intersection 1 as proved by hypothesis in scenario
          ▪ It is the case that: red car approaches intersection 1 at 1 as proved by hypothesis in scenario
      ◦ It is the case that: red car stops at intersection 1 at 1 as proved by hypothesis in scenario
      ◦ It is the case that: red car can enter intersection 1 at 2 for rule 1 as proved by KB Text
        because
          ▪ It is the case that: red car has watched intersection 1 before 2 as proved by hypothesis in scenario
          ▪ It is NOT the case that: there are users crossing intersection 1 at 2 ~ KB Text
          ▪ It is the case that: there is a safe gap for red_car in intersection_1 at 2 as proved by hypothesis in scenario

```

Fig. 1. Human-readable explanation given by the system

points deducted from his license. Some penalties are included in the Highway Code, in “Annex 5. Penalties: Information and rules about penalties, including points and disqualification, a penalty table, new drivers and other consequences of offending”. We model the penalties in LE and refer to the related obligation. In our case the penalty to assess for not stopping at a stop sign is that for “Traffic light offences”, which is a £1,000 fine/Discretionary disqualification maximum penalty and a 3 point deduction. In keeping with our view of unitary model, it is worth raising the question whether AVs (or the manufacturer) and humans would face similar penalties in similar circumstances. This is, however, out of the scope of this paper, so we will instead simply consider how the penalty rule could be modelled in Logical English.

Let us here consider the 3 point deduction:

```

an agent loses 3 points
if the agent has the obligation that
  the agent stops at the intersection
and it is not the case that
  the agent stops at the intersection.

```

A sample of reasoning is given in Figure 1 by way of explanation, showing in human readable form that in LE it is possible to reason with obligations, and their fulfillment/violation, penalties and reparations. It is also possible to directly execute rules in Prolog, and use the generated answer as a log.¹²

6 Conclusions

In this paper we explored a methodology for formally modelling road rules for AVs which can be used for inference. In particular we looked at a small section of the UK Highway Code, and its representation in Logical English.

The assumptions at the basis of this research are that: 1) the AV should not make use of a different high-level representation from that used by humans; and

¹² See: <https://github.com/LyzardKing/mind-the-gap/tree/mindTheGap>

2) AV actions should be transparent and explainable in terms that are consonant with the rules of the road and with the understanding of legal professionals. This would make for a “level-playing field” between human and AV drivers in the common driving environment.

There are requirements in general from AI and Law of legal rule modelling and in specific from AV and human-machine interaction. The rules particularly need to address open texture, commonsense, and tacit knowledge, among others as discussed in Section 5.

In order to realise these assumptions, we have a close link between the source code and the original text in a unitary model. Logical English enables this by having a human readable syntax that is directly executable by machines, as described in section 3. The representation in LE could also be used to test the system in different scenarios and could present a detailed readable log, usable as a justification for its behaviour.

The Highway Code contains not only explicit rules, but also rules for *best practice*, which relate to duty of care, liability, and negligence. A unitary model with a natural language interface aims to provide rules of the road shared by humans and AVs.

References

- [1] R. Kowalski, J. Davila, G. Sartor, and M. Calejo, “Logical english for law,” in *Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022)*, 2022.
- [2] N. Mehdipour, M. Althoff, R. D. Tebbens, and C. Belta, “Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges,” *Automatica*, vol. 152, p. 110 692, 2023.
- [3] H. Prakken, “On the problem of making autonomous vehicles conform to traffic law,” *Artif. Intell. Law*, vol. 25, no. 3, pp. 341–363, 2017.
- [4] UK Department for Transport, *The pathway to driverless cars: a detailed review of regulations for automated vehicle technologies*. Feb. 2015.
- [5] A. Rizaldi, J. Keinholz, M. Huber, *et al.*, “Formalising and monitoring traffic rules for autonomous vehicles in isabelle/HOL,” in *Integrated Formal Methods*, N. Polikarpova and S. Schneider, Eds., Cham, 2017, pp. 50–66.
- [6] J. Collenette, L. A. Dennis, and M. Fisher, “Advising autonomous cars about the rules of the road,” in *Proceedings Fourth International Workshop on Formal Methods for Autonomous Systems (FMAS)*, M. Luckcuck and M. Farrell, Eds., vol. 371, 2022, pp. 62–76.
- [7] A. Hammoud, A. Mourad, H. Otrouk, and Z. Dziong, “Data-driven federated autonomous driving,” in *Mobile Web and Intelligent Information Systems*, I. Awan, M. Younas, and A. Poniszewska-Marañda, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2022, pp. 79–90.

- [8] J. Wang, Y. Zheng, Q. Xu, and K. Li, “Data-driven predictive control for connected and autonomous vehicles in mixed traffic,” in *2022 American Control Conference (ACC)*, 2022, pp. 4739–4745.
- [9] M. Houtenbos, H. M. Jagtman, M. P. Hagenzieker, P. A. Wieringa, and A. R. Hale, “Understanding road users’ expectations: An essential step for ADAS development,” *European Journal of Transport and Infrastructure Research*, vol. 5, no. 4, Dec. 2005.
- [10] S. Vacek, T. Gindele, J. M. Zollner, and R. Dillmann, “Using case-based reasoning for autonomous vehicle guidance,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4271–4276.
- [11] G. Governatori, T. Bench-Capon, B. Verheij, M. Araszkiewicz, E. Francesconi, and M. Grabmair, “Thirty years of artificial intelligence and law: The first decade,” *Artificial Intelligence and Law*, vol. 30, no. 4, pp. 481–519, 2022.
- [12] T. Bench-Capon, M. et al. Araszkiewicz, K. Ashley, *et al.*, “A history of AI and law in 50 papers: 25 years of the international conference on AI and law,” *Artificial Intelligence and Law*, vol. 20, no. 3, pp. 215–319, 2012.
- [13] T. Bench-Capon and P. R. S. Visser, “Open texture and ontologies in legal information systems,” in *Eighth International Workshop on Database and Expert Systems Applications, DEXA*, R. R. Wagner, Ed., IEEE Computer Society, 1997, pp. 192–197.
- [14] B. H. Bix, “Defeasibility and open texture,” in *The Logic of Legal Requirements: Essays on Defeasibility*, J. F. Beltrán and G. B. Ratti, Eds., Oxford University Press, 2012.
- [15] P. Irvine, A. A. B. Da Costa, X. Zhang, S. Khastgir, and P. Jennings, “Structured natural language for expressing rules of the road for automated driving systems,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2023, pp. 1–8.
- [16] S. Kothawade, V. Khandelwal, K. Basu, H. Wang, and G. Gupta, “AUTO-DISCERN: Autonomous driving using common sense reasoning,” presented at the International Conference on Logic Programming 2021 Workshops, J. Arias, F. A. D’Asaro, A. Dyoub, *et al.*, Eds., ser. CEUR Workshop Proceedings, vol. 2970, Porto, Portugal (virtual): CEUR, Sep. 2021.
- [17] H. Bhuiyan, G. Governatori, A. Bond, and A. Rakotonirainy, “Traffic rules compliance checking of automated vehicle maneuvers,” *Artif. Intell. Law*, vol. 32, no. 1, pp. 1–56, 2024.
- [18] T. Kuhn, “A survey and classification of controlled natural languages,” *Comput. Linguistics*, vol. 40, no. 1, pp. 121–170, 2014.
- [19] N. E. Fuchs and R. Schwitter, “Attempto controlled english (ACE),” 1996.
- [20] J. R. Curran, S. Clark, and J. Bos, “Linguistically motivated large-scale NLP with c&c and boxer,” in *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, J. A. Carroll, A. van den Bosch, and A. Zaenen, Eds., The Association for Computational Linguistics, 2007.

- [21] S. Guy and R. Schwitter, “The PENGASPSystem: Architecture, language and authoring tool,” *Language Resources and Evaluation*, vol. 51, no. 1, pp. 67–92, Mar. 2017.
- [22] G. Brüggemeier, *Common principles of tort law: a pre-statement of law*. British Institute of International and Comparative Law, 2006.
- [23] H. Schebesta, “Risk regulation through liability allocation: Transnational product liability and the role of certification,” *Air and Space Law*, vol. 42, no. 2, pp. 107–136, 2017.
- [24] T. Bench-Capon and F. Coenen, “Isomorphism and legal knowledge based systems,” *Artificial Intelligence and Law*, vol. 1, no. 1, pp. 65–86, 1992.
- [25] K. V. D. Berg, T. V. Engers, and P. Kordelaar, “Requirements for legal knowledge representation,” p. 12, 2007.
- [26] S. Costantini and G. Lanzarone, “Explanation-based interpretation of open-textured concepts in logical models of legislation,” *Artificial Intelligence and Law*, vol. 3, Jul. 2000.
- [27] E. Mueller, *Commonsense Reasoning*. Elsevier Science & Technology Books, 2010.

Forensic Analysis of Social Media iOS Apps via Reconstructing Timelines

Oluwafisayo Theophilus¹ and Hongmei Chi^{1[0000-0003-4610-6479]}

¹Florida A&M University, Tallahassee FL 32307-5100, USA
oluwafisayo1.theo@famu.edu, hongmei.chi@famu.edu

Abstract. Acquiring evidence that occurs over time is a significant consideration in digital forensics, mainly social media forensics. Cybercriminals are using the rise in web usage and the amount of time people spend on social media to their advantage. Digital forensics requires the analysis of immense numbers of events due to the growth of the internet, social media platforms, interconnected devices, and innovative technology. Performing digital forensic timeline analysis is becoming essential because of the number of storage devices and the increase using smartphones. This paper will give some case studies and extract digital evidence based on timeline analysis. The paper aims to use machine learning and timeline analysis to extract evidence and event timestamps. This methodology will help forensic investigators to recover artifacts from IOS devices.

Keywords: iOS device, Timeline Analysis, Social Media Forensics, Machine Learning, Natural Language Processing

1 Introduction

Social media is a convenient way of connecting with strangers online, but it could make the user vulnerable to predatory behaviors such as fraud, stalking, etc. Recently, the activities and time users spend online have significantly increased due to staying and working from home, increased social media users, increased awareness, etc. This increase in usage could have security and safety implications for users. Smartphones now have much more storage space and typically have many installed applications. The investigators still have trouble deciding which application would be best for their inquiry and locating where the artifacts are stored. In digital forensics, "artifacts" are the data remnants or traces left on electronic devices or systems, usually stored temporarily or permanently in the device's memory. In the context of the WhatsApp message, this artifact provides valuable chronological context, date, time, or period, potentially aiding investigators in establishing timelines, understanding the sequence of events, and corroborating or refuting alibis or claims made by individuals involved in the investigation. Using secure private messaging apps like WhatsApp, Facebook Messenger, and other social media apps does not make it easier for users to maintain their anonymity. These applications require users to register with their phone numbers and direct communications through centralized servers, making it difficult to conceal IP addresses when utilizing anonymity tools like VPNs [3].

In digital forensics, capturing conversations among suspects or between a suspect and a victim can be essential for obtaining evidence for criminal cases. Due to the rising popularity of social networking apps, forensic investigators can find a wealth of information on these devices. These gadgets may contain potential evidence, which can be retrieved with the correct instruments and inspection techniques. The objective of the forensic examinations will be to ascertain if the actions carried out via this application (WhatsApp) were saved on the internal memory of the device in question. If so, it established what information could be found on the device, the location of the data obtained from each device's logical image, and its extent and importance. As a result, this communication must be accessible to a digital forensic investigator by being simple to find, save, examine, and report on [6]. This digital landscape has witnessed a surge in online scams, encompassing various deceptive practices. Our method will speed up digital forensics investigation.

This paper delves into a unique and complex scam, the "Pig Butchering Scam," which has evolved at the intersection of dating and other social media iOS apps. We present a comprehensive analysis of this scam, its origins, modus operandi, and the role of digital forensics in its detection and prevention. This study also highlights the resilience of scammers and the need for ongoing vigilance and technological advancements to protect individuals from falling victim to such elaborate frauds. Creating a forensics timeline is one of the critical processes in digital evidence. The purpose of forensic timelines is to give a summary of the events and their dates. After recursively reading every file from a forensic picture, time information is extracted from the metadata to create a forensic timeline. The process additionally retrieves events from log files or other artifacts. One of the tools for creating forensic timelines is Autopsy. A timeline can be made using various files and events containing timestamp information [12]. Forensic chronology is created from different iOS artifacts, such as call database logs, Bluetoothplist files, and application conduit logs.

1.1 Identify the Need

Smartphones and other linked devices, like PCs, may contain traces of incident-related evidence as digital technology and communication progress. Finding relevant artifacts poses new and complex challenges for digital investigators because there are many different sources of digital evidence due to the heterogeneity of devices and storage units. For law enforcement, cybersecurity specialists, and legal professionals, the ability to extract, analyze, and interpret data from these apps has become essential due to the proliferation of mobile devices and the extensive adoption of social media platforms. The complexity of the problem is linked to the complex encryption methods employed by these platforms, the intricate data structures used by iOS apps, and the dynamic nature of social networking sites. Timeline analysis can be a great solution to acquiring artifacts and timestamps of the incident. One of the most effective methods for identifying, authenticating, and visualizing evidence is machine learning (ML). Our proposed framework combines timeline analysis and machine learning to provide a method for acquiring and visualizing artifacts.

The research questions to be addressed are as follows:

- Are WhatsApp messages' forensically recoverable on iOS devices?
- Are the timelines of events also recoverable?
- What impact could the storage ability of iOS devices have on forensic criminal investigation?

1.2 Contribution

The primary focus of this paper is to conduct forensic examinations of social media applications such as WhatsApp on iOS (iPhone) devices. Our contributions lie in retrieving active and deleted artifacts from WhatsApp's primary storage areas and related data from local files, exchanged messages, deleted messages, file evidence, user accounts, contact lists, and incident timelines. We ensure the credibility and reproducibility of our findings by conducting analyses on non-jailbroken iOS devices. Additionally, we explore avenues beyond mere chat preservation, investigating various data locations within WhatsApp on iOS. Leveraging Natural Language Processing (NLP) and machine learning (ML) techniques, we delve into sentiment analysis of conversations. By integrating forensic tools alongside NLP and ML, our research aims to facilitate tracking incident timestamps, artifacts, and crucial metadata.

The Challenge of extracting temporal events from social media streams is significant, and the structure of the problem is similar to the general problem of reconstructing temporal sequences of accumulated streams of evidence in regular police forensic analysis. The significance of machine learning for timeline analysis from a vast amount of social media evidence in digital forensics investigations cannot be overstated. This approach enables investigators to efficiently reconstruct chronological sequences of events from enormous amounts of digital data, providing invaluable insights into the activities and interactions of individuals across various platforms.

The following summarizes the remaining portion of this paper: Section 2 includes relevant works and discusses how these studies pinpoint knowledge gaps. Section 3 presents the approach and uniqueness of the research. Section 4 presents our tools and techniques for analysis, while Section 5 details our experiments and analysis. Results and Implementation are discussed in Section 6, while Section 7 contains the conclusion.

2 Related Work

Related work in social media forensic analysis on smartphone devices focused on Android devices using tools such as Android Debug Bridge (ADB) to extract artifacts [4]. This research, titled Forensic Analysis of Social Media Android Apps via Timelines, addressed the Challenge of acquiring events and artifacts that occur over time on social media in digital forensics using iOS devices. Acquiring these artifacts on social media is quite challenging because of the growth of the internet, interconnected devices, and their storage ability. Performing digital forensics is becoming difficult. This study reconstructed a case study with an Android device on WhatsApp's social media using timeline analysis with

an Android Debug Bridge (ADB) application. This method extracted device info and properties, explored available file system artifacts, and gathered specific data.

Shawn Knox et al [6] explained the acquisition tools and forensic analysis of Android and iOS on the Happn dating app. Mobile device forensics and network analysis on the app were conducted. The paper highlighted the potential for malicious actors to uncover other users' personal information. A forensic analysis of the app's activity on Android and iOS devices was conducted using commercial and freely available tools. The research included creating profiles, capturing network traffic, acquiring device images, and backing up iOS devices with iTunes. These images and backups are analyzed to reveal further artifacts. The findings revealed the information of the user and the profile they interacted with, such as exact location with longitude and latitude, URL links containing all photos uploaded by the user and others they interacted with, access tokens, user IDs, and profile information posted by users. Tools, such as Fiddler proxy, Packet Capture, MOBIL edit Forensic Express, FTK, and Autopsy, were utilized in the study. The acquisition tools and forensic analysis of Android and iOS on the Happn dating app entailed mobile device forensics and network analysis on the app [6].

Research on the forensic analysis of ephemeral messaging applications [5] conducted a thorough forensic examination of 'disappearing communications' across many digital platforms (mobile, desktop, and Cloud) and instant messaging apps (WhatsApp, Snapchat, and Telegram) to assess whether they can be recovered in the shortest possible time. Heath et al. [5] research the forensic analysis of ephemeral messaging applications and conduct a thorough forensic examination of 'disappearing communications' across many digital platforms (mobile, desktop, and Cloud) and instant messaging apps (WhatsApp, Snapchat, and Telegram) to assess whether they can be recovered in the shortest possible time. From the findings, WhatsApp is the least destructive for prospective evidence, but retrieving data via the Cloud is time-consuming. Snapchat is the second most destructive for prospective evidence, with iOS devices requiring manual inspection, whereas physical Android analysis could collect all the essential artifacts for admissible evidence. Finally, Telegram is the most destructive way to gather prospective evidence during the investigation.

The findings from [3] demonstrated a thorough examination and comparison of the information stored in each app's database to pinpoint the local user's multiple instant messaging accounts, contact list, contents of messages sent to contacts, and conversational history. Akinbi et al.[3] executed a forensic investigation of Monal and Siskin IM, two decentralized open-source XMPP multi-client social networking apps on iOS devices that use OMEMO end-to-end encryption to ensure anonymity and privacy. Each app's databases and storage places within the iOS file system that store local copies of user information and metadata were identified. These databases and storage places were searched for forensic evidence evidentiary data. The findings of this article demonstrated a complete analysis and correlation of data contained in each app's database to identify the local user's various IM accounts and contact list, contents of messages exchanged with contacts, and conversation chronology. According to the findings, no evidence of social networking activity could be recovered from BlackBerry smartphones. However, iPhones and Android phones saved substantial useful data that forensic investigators could recover and use. However, none of these works utilized our forensically sound framework that leverages ML, NLP, and timeline analysis of iOS devices.

3. Approach and Uniqueness

Our approach involves deploying WhatsApp applications to recreate scenarios on an iOS (iPhone) smartphone. Engaging with a separate device was realistic because the app is compatible with iOS smartphones. In the studies, the contact information of the test local user was stored on the iOS device. To gather as many forensic artifacts as possible, we set up an experiment that involves one-on-one chats between a local user and other users, which depicts the scammer and the victim in the contacts list, the sharing of multimedia files, blocking and deleting contacts, creating groups, and chats. Additionally, we will show that a realistic user interaction has occurred. A timeline analysis of the logical image using iTunes and Autopsy forensic tool on the device representing the test local user's profile will be carried out to search for forensic artifacts and erased data. The instruments used for data collection and processing are depicted in Fig. 2.

A logical copy of all the data from the iOS device, including WhatsApp messages, SMS, text messages with attachments, multimedia files, call logs, contact lists, application data, etc., was created using iTunes backup, which synchronizes iOS devices with desktops and extracts logical images from the Cloud. Fig. 1 provides a summary of the suggested system data flow.

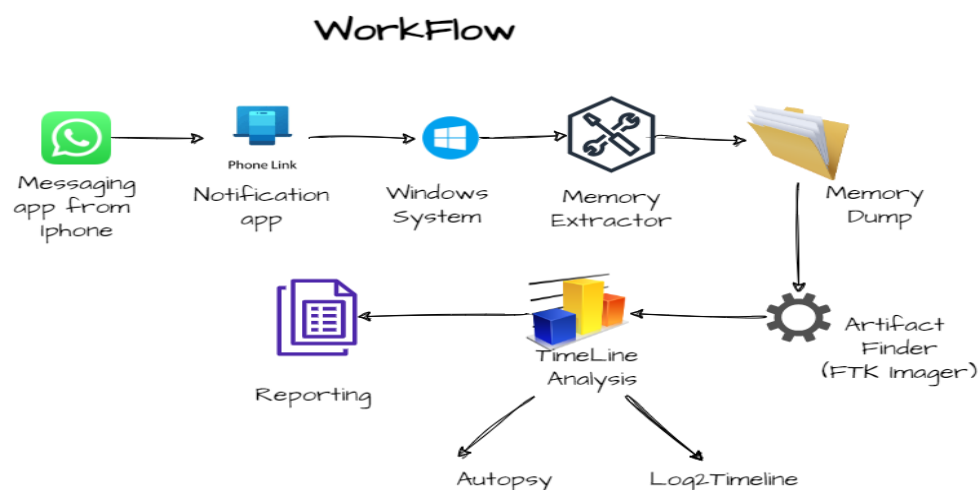


Fig. 1: Proposed System Data Flow

This approach also takes input from the raw data from a communication network involving the suspects/victim and their conversation and is labeled as vertices and edges, respectively. In the next phase, unsupervised NLP was utilized to extract meaningful topics from the conversation. Then, a feature selection algorithm was adopted in this phase to avoid infringing on the user's privacy, and with the selected feature set obtained from the prior phase from NLTK, a sentiment intensity analyzer was imported to analyze the conversation. Applying NLP and digital forensics in social media analysis unveils hidden relationships. The prevalence of hidden relationships in social media, often concealed

within the vast realm of social media, has raised concerns regarding trust, privacy, and ethics. This paper outlines a pioneering approach using NLP and digital forensics techniques to unveil such relationships [8,9]. Through a timeline-based analysis, we demonstrate how NLP-driven digital forensics can aid in uncovering hidden relationships on social media platforms.

4. Methodology and Tools

Our method allowed for generality, reproducibility, and completeness, which is crucial because the goal of any forensic investigation is to give the analyst access to the digital evidence generated by the programs being looked at. In that order, we created an investigative scenario for the application, followed by the "Installation of application" and "Design of experiments" phases. During the "Installation of application" phase, we downloaded and installed WhatsApp version 2.21.221. We used DB Browser SQLite 3.12.2 to open and view SQLite databases on the logical extraction. The program might be useful for forensic investigators who need to access the database files commonly found on tablets and smartphones.

4.1 Heterogenous Data Format

Due to the expanding range of technologies available to individuals, forensics investigators frequently confront several difficulties when reviewing and interpreting data. When gathering evidence, they frequently must process and assess numerous systems manually. The abundance of data has directly impacted investigators as they are now required to review and evaluate more files in every case. Due to the heterogeneity of the evidence, investigations may include multiple resources and applications, making it challenging for investigators to consistently identify evidence across all digital devices without putting an undue strain on their cognitive abilities [4]. This focuses on the significance of varied evidence from social media sites and different types of technology in the context of cybercrime.

Digital evidence handling errors, damage, and loss can occur at any point in the chain of custody, including the crime scene, packing, transmission, and examination and storage of the evidence. Even though forensic scientists are concerned about contamination and have created processes to identify and lower these risks, law enforcement needs to be more cautious and try to prevent contamination.

4.2 Timeline analysis

Using timeline analysis, law enforcement organizations may discover much about the actions, communications, and conduct of those involved in criminal activity. This facilitates a thorough inquiry process and helps build a solid evidence foundation. The social media data gathered by the investigators is arranged in a timeline format. To create a logical sequence of events, they align the activities based on timestamps, emphasizing significant user interactions, posts, and changes in user behavior.

5. Experiment

In the experiments phase's design, we define a set of experiments that involve using the WhatsApp application and sending text messages between the devices. These will aid in the generation of artifacts. Fig. 3 shows the exchange of text between the scammer and the victim through social media application. Below is the List of tools and applications used in the experiment.



Fig. 2: Devices used for data acquisition

- (1) One iPhone 11, 64GB
- (2) One Windows 11 pro laptop
- (3) One Android phone (Samsung)
- (4) Apple iTunes Application
- (5) Plist Editor for Windows
- (6) SQLite Database Browser
- (7) WhatsApp application
- (8) Autopsy Application

Three steps were made up for the test procedure: scenario, logical Acquisition, and analysis. Details about each step are provided in the following Sections.

5.1 Scenarios

In this stage, common user activities were conducted on smartphone social networking applications. The WhatsApp application was installed on smartphones; the application was utilized because of the volume of users and the end-to-end encryption of messages on the application. A fictional user was chosen to simulate the case statement below. For the device, a predefined set of activities was initiated, such as exchanging text messages between the scammer and the victim.

5.1.1 Case Statement

Online Dating and Romance Scams

"Jason" received a friend's request from "Emily," a total stranger. After Jason agreed to the request, Emily revealed she was an American in Germany. They appeared to click, so they started organizing a trip for that summer when Emily would be back in the country. After sending Jason a \$5,000 check to cover the trip's expenses, Emily abruptly requested that he return her \$4,500 since she had lost her job and required the cash for rent. Jason deposited the check and wired the funds, but his bank quickly called him to inform him that the \$5,000 check was a forgery and that he was responsible for \$4,500 in bank fees. The false "Emily" vanished, and Jason never heard from her again, despite losing the money [1].

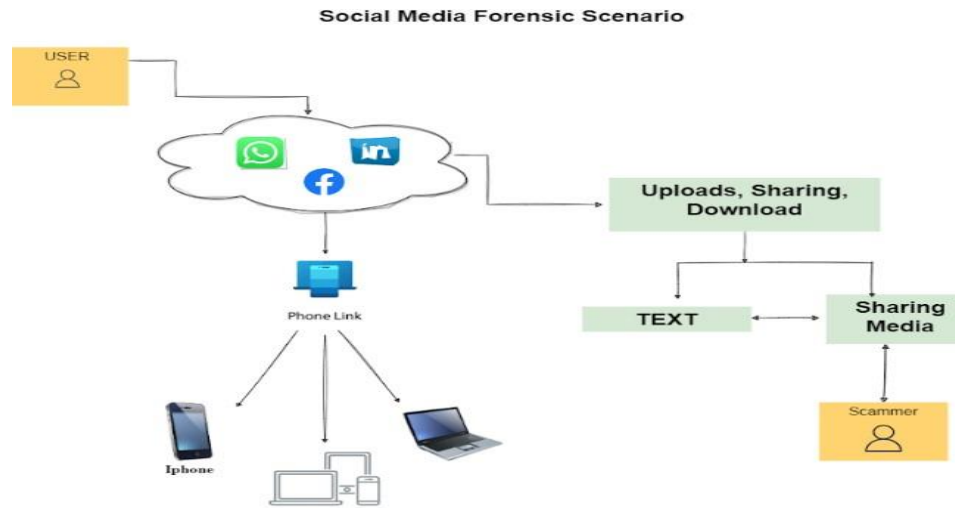


Fig. 3: Social media forensic scenario

5.2 Logical Acquisition from iOS

The second step was to get a logical image of each device's internal memory. The acquisitions were carried out in a controlled setting using forensically sound methodologies to guarantee the integrity of the collected data and its possible acceptability in court. It is possible that during such an acquisition, data traces of inactive files would be overlooked, as is always the case with all logical acquisitions. Fig. 5 shows some of the extracted logical evidence and their locations.

5.2.1 iPhone Forensic Logical Acquisition

The iPhone 11 devices were logically acquired using the Apple iTunes software (version 10.4.0.80). iTunes is a free synchronization and management utility that is available for download from the Apple website. Information, media files, and apps are synchronized between the host computer and Apple devices (i.e., iPhone, iPad, and iPod). It also stores data from the Apple device's backup on the host computer. It is not meant for forensic investigators to use iTunes. However, it can be utilized by forensic specialists to obtain a backup file or logical image of an Apple device. If the suspect has upgraded their software, synced their device, or reset it to factory settings, backup data may also be on their computer.

A forensic examiner must confirm the validity of the collected evidence. Therefore, the logical acquisitions of the iPhone were conducted in an environment appropriate for forensics. Before connecting any iPhone to the forensic workstation, turning off the iTunes application's automatic synchronization capability was imperative. iTunes immediately syncs the device with the host computer as soon as it is connected. Disabling automatic synchronization preserves the integrity of the data on the iPhone by preventing user data from being transmitted between the device and the host computer. After iTunes was configured, the iPhone was attached to the forensic workstation via a USB data cord.



Fig. 4: chat conversation between the victim and scammer

Upon identification of the device by the iTunes software, one might initiate a manual logical acquisition by right-clicking on the device name and selecting "backup." Next, iTunes created a copy of the iPhone and by default stored the backup files in this directory: This file can be found at C:\Users\[user]\Apple \MobileSync\Backup\[unique identifier]. The forensic workstation and iPhone were disconnected when the backup process was complete. Fig. 5 shows the List of artifacts available on the device. The backup files were visualized using plists: A plist file is a binary file from the iOS operating system that contains the characteristics of a particular activity. To extract timestamps and events from the file, we must parse it. A macOS native tool like plist Editor for Windows was used to parse and read the contents of a plist file.

5.3 Timeline Analysis

The third phase entailed conducting forensic examinations on the obtained logical image of the device. The goal was to ascertain whether the activities performed through various applications were stored in the device's internal memory. If such data existed, the examination focused on determining the quantity, location, and significance of the retrievable information from the logical image of each device. These examinations were carried out manually, utilizing Autopsy to inspect the acquired images. The process involved identifying unique headers or signatures in each structure, searching for data associated with social networking applications, and understanding the storage methods of this data on each device. The Autopsy method is used to construct a forensic timeline from an iOS forensic image obtained through logical Acquisition.

XML View	List View	
Key	Type	
Root	dict	
Applications	dict	
Build Version	string	
Device Name	string	
Display Name	string	
GUID	string	
ICCID	string	
IMEI	string	
IMEI 2	string	
Installed Applications	array	
Last Backup Date	date	
MEID	string	
Phone Number	string	
Product Name	string	
Product Type	string	
Product Version	string	
Serial Number	string	
Target Identifier	string	
Target Type	string	
Unique Identifier	string	
iTunes Files	dict	
iTunes Settings	dict	
iTunes Version	string	

string	com.etsy.etsyforios
string	com.lifewave.babyheartso
string	com.oculus.twilight
string	com.lightricks.Enlight-Edi
string	com.uba.vericash
string	org.igggymedia.periodtrack
string	com.wellocution.iosapp
string	com.m-w.dictionary
string	com.freelancer.messenger
string	com.kinderlime.dev
string	com.facebook.Messenger
string	net.whatsapp.WhatsApp
string	com.hearify
string	com.storytoys.duploworld
string	ph.telegra.Telegraph
string	am.impossible.hear-my-b
string	com.truistmobile.truist
string	com.bossrevolution.umus
string	com.simpleradio.SimpleR
string	com.stj.idphoto
string	com.groupme.iphone-app
string	com.joinhandshake.Hand
string	com.engage.Chick-fil-A
string	com.netflix.Netflix
string	com.necoretext.duplicat
string	com.google.Drive
string	com.sourdis.cash

Fig. 5: List of various artifacts (left) and List of applications (Right) on the device

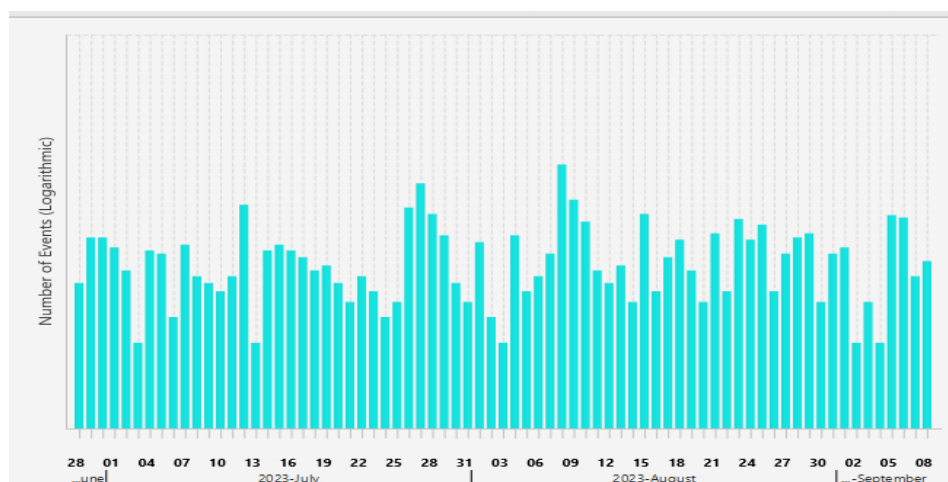


Fig. 6: Graphical representation of the timeline analysis of iOS

5.4 Sentiment Analysis of the WhatsApp Conversation

Autopsy and similar forensic tools lack advanced Natural Language Processing (NLP) capabilities to extract event mentions from textual data accurately. While these tools can parse and analyze text data, they may not provide sophisticated NLP algorithms. As a result, investigators may manually download the victim's conversation and extract the events that occurred during the investigation to analyze the communication with the identified scammer. This session elaborates on the detailed steps involved in conducting sentiment analysis of the WhatsApp conversation.

Extract WhatsApp Conversation: Extracting a WhatsApp conversation involves retrieving the messages exchanged between the scammer and victim from the WhatsApp platform. This can be done by accessing the chat history on the device where the conversation occurred or by obtaining the conversation data from a backup file. WhatsApp typically stores conversation data in a structured format that includes information about the sender, timestamp, and message content.

Data Preprocessing: Data preparation, which includes procedures to guarantee the data's homogeneity and cleanliness, is an essential stage in getting extracted conversation data ready for analysis. Tokenization, which divides the text into individual words or tokens, lowercasing which ensures that all text is in lowercase for consistency removing stopwords which eliminates common, non-informative words like "and" or "the" removing punctuation from the text, handling numerical values determining whether to retain or remove them based on analysis goals and addressing missing or null values in the dataset are some of the key steps involved in the process. The main goal of data preparation is to provide a clean, consistent dataset that can be accurately analyzed. This will make it easier to do tasks like sentiment analysis and other natural language processing applications later.

Analyze Sentiment Using NLTK: Once the data is preprocessed, the next step is to analyze sentiment using the Natural Language Toolkit (NLTK). NLTK is a powerful library in Python for natural language processing tasks. Here's a basic outline of the sentiment analysis process:

- Tokenization: Break the preprocessed text into sentences or words.
- Sentiment Analysis: Use NLTK's sentiment analysis tools to determine the sentiment of each sentence or document. This could involve using pre-trained sentiment analysis models or lexicons.
- Sentiment analysis typically classifies text sentiment into positive, negative, or neutral categories. The output can provide insights into the emotional tone of the conversation, helping to understand the sentiment expressed by the scammer and victim.

6 Examination and Analysis of Logical Acquisition

To build a logical image of the devices, the logical file that was backed up was kept in a folder specifically designated by an alphanumeric name (hash value) for each iPhone acquired through iTunes. In both directories, there were many backup files without apparent extensions and three plist files—one for each of the databases, mbdb and mbdx. For every backup file, a unique 40-character alphanumeric identity is created.

When the backup files were opened and examined in a text editor, it became clear that they were either plain text or binary files that might have contained further plist files, SQLite database files, or encapsulated images. The files needed to be decoded and read using the right tools to review the contents. Binary plist data was contained in files beginning with the header "bplist00," and SQLite databases were contained in files beginning with the header "SQLite format 3." Various technologies were employed to manually review the backup files' contents based on their classification. SQLite Database Browser (version 1.3) was used to assist in reading and examining backup files containing SQLite databases. In contrast, Plist Editor for Windows (version 3.0.0) was utilized to assist in reading and examining backup files containing Plist data.

Upon manual inspection, the backup files revealed user information, such as SMS sent and received, calendar entries, phone logs, and address book entries. However, this study's main goal was to find out if these backup files contained any footprints of social networking programs. The files in the backup directory were manually searched using the Command-line utility for terms related to the social networking activities throughout the experiment. After the files containing the keywords were decoded using the proper software (such as Plist Editor and SQLite Database Browser), their contents were carefully checked for evidence of the experiment's prior activity, some of the retrieved artifacts can be seen in Fig. 7.

Table 1: Social Network data recovered from the logical image of each smartphone.

Device	Artifacts
iPhone 11	Device name, Phone number, WhatsApp images, installed applications, phone name, messages, location of WhatsApp chats, etc

An iOS forensic image can be used to evaluate a variety of data. From the timeline analysis of the iOS device, the location of the net.WhatsApp.Whatsapp is /private/var/mobile/Containers/Data/Application/1FA7E123-0DD7-4C47-B4B9-D4C0BE6754f7 according to Fig. 5. Several pieces of information that can be found on the device are WhatsApp stored location, call logs, contact names, groups, number of messages per contact, phone calls, sent messages, and media files.

In this Section, we give the details of our investigation and reconstruct our timeline for this specific case.

To build a forensic timeline, we used the Autopsy tool. Autopsy, a powerful open-source digital forensics platform, offers various functions for investigating and analyzing digital evidence. Some key functions include file system analysis, keyword search, timeline analysis, metadata extraction, hashing for data integrity verification, and the ability to parse and interpret various artifacts, such as email, chat messages, and multimedia files. The timeline generated can be written to CSV files, e.g., forensic-timeline.csv. An example of forensic generated by the Autopsy from an iOS image can be shown below in Fig 5. The autopsy tool was able to generate the various events that took place on the phone over a period, as shown in Fig. 7. The timeline analysis generated some messages, call log and web history for July and August 2023

Program Name	Path	Path Source
com.apple.FunCamera.ShapesPicker	/Applications/FunCameraShapes.app	/private/var/mobile/Containers/Data/Application/9C9...
com.apple.mobilephone	/Applications/MobilePhone.app	/private/var/mobile/Containers/Data/Application/D9E...
com.apple.CloudKit.ShareBear	/Applications/iCloud.app	/private/var/mobile/Containers/Data/Application/6D...
net.whatsapp.WhatsApp	/private/var/containers/Bundle/Application/F9C951F3...	/private/var/mobile/Containers/Data/Application/1F4...
com.burbn.instagram	/private/var/containers/Bundle/Application/C013A60E...	/private/var/mobile/Containers/Data/Application/528...
com.zhiliaapp.musically	/private/var/containers/Bundle/Application/61D6CEE0...	/private/var/mobile/Containers/Data/Application/07A...
com.instructure.icanvas	/private/var/containers/Bundle/Application/565D4DFE...	/private/var/mobile/Containers/Data/Application/166...
com.microsoft.Office.Outlook	/private/var/containers/Bundle/Application/AB1AC07...	/private/var/mobile/Containers/Data/Application/158...
com.google.Gmail	/private/var/containers/Bundle/Application/F66A653E...	/private/var/mobile/Containers/Data/Application/CE...
us.zoom.videomeetings	/private/var/containers/Bundle/Application/CCD39C1...	/private/var/mobile/Containers/Data/Application/99C...
com.hp.printer.control	/private/var/containers/Bundle/Application/2ABF7E72...	/private/var/mobile/Containers/Data/Application/4D...

Fig. 7: iOS artifacts and their location

6.1 Result

The iPhone 11 logical backup copies were acquired, analyzed, and examined forensically for this investigation. Installing a social networking app on the device, carrying out typical user actions via the software, obtaining a forensically sound logical image of the device, and doing manual forensic analysis on the logical image were the tests' main components. The forensic study identified the location, quantity, and importance of social networking data that could be located and extracted from the device's logical image. In this experiment, we used WhatsApp as a case study. The analysis generated a DateTime, timestamp, source, message, phone number, web history, and messages.

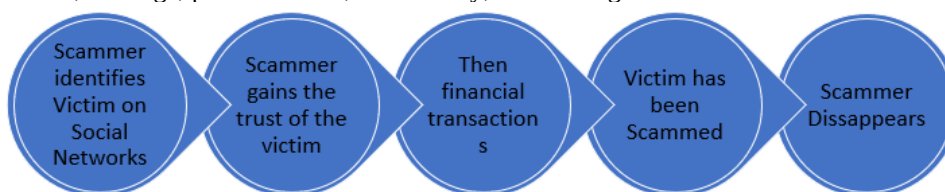


Fig. 8: Timeline of a Pig butchering scam victim

Based on the sentiment analysis of the WhatsApp conversation, the conversation was neutral, and the names of the victim and scammer, the location, and some content of the conversation were shown in the word Cloud. Fig. 8 shows the trend of dialogue between the scammer and the victim. The findings demonstrated that forensic investigators could retrieve and utilize a substantial quantity of valuable data that was stored on iPhone phones. The document detailed each device's recoverable social networking data characteristics and its specific locations within backup files. We aspire for this paper to serve as inspiration for developing digital forensics tools capable of extracting and reconstructing social networking data from a diverse range of contemporary smartphones.

The artifacts that Autopsy was able to recover from WhatsApp on an iPhone device to conduct timeline analysis include a variety of components that are essential for reconstructing patterns and activities of communication. Time stamps, contact details, media files, phone logs, group chats, message metadata, deleted messages, and user account information are some examples of these artifacts. When these artifacts are examined in the context of a timeline, important information about WhatsApp usage events, interactions, and conversation histories can be obtained. This thorough analysis makes it easier for forensic investigators to recreate digital evidence and create inquiry timelines by helping them comprehend the chronological order of WhatsApp actions.

6.2 Discussion

The paper focuses on timelines as a forensic tool, but it may not sufficiently address the dynamic nature of social media content. This study uses Autopsy to construct a forensic timeline from an iOS forensic image. Autopsy provided an overview of all files and artifacts in a tree structure. Social media platforms frequently update their features, algorithms, and data storage mechanisms. By leveraging advanced algorithms, investigators can identify patterns, detect anomalies, and uncover critical information that may otherwise go unnoticed. This method not only streamlines the investigative process but also enhances accuracy and effectiveness in piecing together digital evidence to support legal

proceedings. Ultimately, timeline analysis through machine learning empowers forensic experts to extract actionable intelligence from social media data, aiding in the resolution of complex cases and contributing to the pursuit of justice.

"Pig-butcher" scams have stolen at least \$75.3 billion in recent years (January 2020 to February 2024) [11]. How to help average people protect their money is a complicated question. In this paper, we propose this timeline analysis can help the investigation to help victims and search for criminals quickly.

7. Conclusion and Future Work

Apple's closed ecosystem imposes restrictions on accessing the internal workings of iOS devices, thereby limiting the capabilities of forensic tools. This constraint may impede comprehensive analysis by restricting access to certain device areas and data. However, utilizing Autopsy enabled us to identify where WhatsApp information and images were stored, along with other available device information, during the investigation. Additionally, interpreting metadata linked to iOS artifacts demands specialized knowledge and tools. To accurately comprehend the context and significance of digital evidence, specialized tools were employed to interpret some of the information generated from iTunes and the Autopsy forensic tool.

Data recovery from smart devices has proven difficult, but our method makes retrieving artifacts from heterogeneous devices and complex storage systems possible. The artifacts timestamps of incidences and deleted evidence to help the investigation. So that an investigator can have a complete and comprehensive forensic timeline from an iOS image, our combined approach also allows law enforcement to obtain data from iOS devices. This study underscores the importance of interdisciplinary cooperation between dating platforms, law enforcement, and digital forensics experts to combat the ever-evolving landscape of online scams. Future work will add iOS artifacts such as Apple Maps, VoIP search logs, reboot logs, and iOS notification logs that are not included in this work [13]. We will apply our methods to Discord and Tiktok [14, 15].

Acknowledgments. This research is based upon work supported partly by IBM Master Fellowship and by the National Science Foundation under Grant CNS- 2104519 and CNS 2101161. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. <https://www.ag.state.mn.us/consumer/Publications/SocialMediaScams.asp>
2. Nadia, M., Yasmin, M., Abbas, A., Guillaume, H., Isabelle L., & Reda Y. (2023). Improvements in Cybersecurity: Coupling Python for Digital Forensics
3. Akinbi, A., & Ojie, E. (2021). Forensic analysis of open-source XMPP multi-client social networking apps on iOS devices. *Forensic Science International: Digital Investigation*, 36, 301122.
4. Ayodeji Ogundiran, Hongmei Chi, Jie Yan. 2024. Forensic Analysis of Social Media Android Apps via Timelines. FICC 2024

5. Heath, H., MacDermott, Á., & Akinbi, A. (2023). Forensic analysis of ephemeral messaging applications: Disappearing messages or evidential data?. *Forensic Science International: Digital Investigation*, 46, 301585.
6. Knox, S., Moghadam, S., Patrick, K., Phan, A., & Choo, K. K. R. (2020). What's really 'Happning'? A forensic analysis of Android and iOS Happn dating apps. *Computers & security*, 94, 101833.
7. Moreb, M. (2022/ xwd). Evidence Identification Methods for Android and iOS Mobile Devices with Facebook Messenger. In *Practical Forensic Analysis of Artifacts on iOS and Android Devices: Investigating Complex Mobile Devices* (pp. 427-457). Berkeley, CA: Apress.
8. Bowling, H., Seigfried-Spellar, K., Karabiyik, U., & Rogers, M. (2023). We are meeting on Microsoft Teams: Forensic analysis in Windows, Android, and iOS operating systems. *Journal of Forensic Sciences*, 68(2), 434-460.
9. Ukwen, D. O., & Karabatak, M. (2021, June). Review of NLP-based systems in digital forensics and cybersecurity. In 2021 9th International Symposium on digital forensics and Security (ISDFS) (pp. 1-9). IEEE.
10. Joseph, D. P., & Viswanathan, P. (2023). SDOT: Secure Hash, Semantic Keyword Extraction, and Dynamic Operator Pattern-Based Three-Tier Forensic Classification Framework. *IEEE Access*, 11, 3291-3306.
11. Griffin, J. M., & Mei, K. (2024). How Do Crypto Flows Finance Slavery? The Economics of Pig Butchering. *The Economics of Pig Butchering* (February 29, 2024).
12. Studiawan, H., Ahmad, T., Santoso, B. J., & Pratomo, B. A. (2022, October). Forensic Timeline Analysis of iOS Devices. In *2022 International Conference on Engineering and Emerging Technologies (ICEET)* (pp. 1-5). IEEE.
13. Sarhan, S. A. E., Youness, H. A., Bahaa-Eldin, A. M., & Taha, A. E. (2024). VoIP Network Forensics of Instant Messaging Calls. *IEEE Access*.
14. Jones, B. (2023). Forensic fandom: True crime, citizen investigation and social media. In *True crime in American media* (pp. 163-179). Routledge.
15. Gupta, K., Varol, C., & Zhou, B. (2023). Digital forensic analysis of discord on google chrome. *Forensic Science International: Digital Investigation*, 44, 301479.

Enhancing Legal Document Retrieval: A Multi-Phase Approach with Large Language Models

Hai-Long Nguyen ^{*1}, Duc-Minh Nguyen ^{*2}, Tan-Minh Nguyen¹,
Ha-Thanh Nguyen³, Thi-Hai-Yen Vuong¹, and Ken Satoh³

¹ VNU University of Engineering and Technology, Hanoi, Vietnam
`long.nh@vnu.edu.vn`

² RMIT University, Vietnam

³ National Institute of Informatics, Tokyo, Japan

Abstract. Large language models with billions of parameters, such as GPT-3.5, GPT-4, and LLaMA, are increasingly prevalent. Numerous studies have explored effective prompting techniques to harness the power of these LLMs for various research problems. Retrieval, specifically in the legal data domain, poses a challenging task for the direct application of Prompting techniques due to the large number and substantial length of legal articles. This research focuses on maximizing the potential of prompting by placing it as the final phase of the retrieval system, preceded by the support of two phases: BM25 Pre-ranking and BERT-based Re-ranking. Experiments on the COLIEE 2023 dataset demonstrate that integrating prompting techniques on LLMs into the retrieval system significantly improves retrieval accuracy. However, error analysis reveals several existing issues in the retrieval system that still need resolution.

1 Introduction

In a rapidly developing digital world, there are enormous amounts of information and queries that need to be processed every minute. Therefore, there is a demand for an effective and efficient search engine that would free law practitioners from heavy manual work. Legal information retrieval is a challenge that has gained interest from both researchers and industry recently. This task study is about the automation of retrieving accurate pieces of information for a given query from users.

In this study, we focus on Legal Article Retrieval, an essential task for many countries that follow a statute law system. The primary objective is to develop a retrieval system that can effectively pinpoint a subset of legal articles that pertain to a given user query. Let Q denote the initial query and C represents a corpus of legal articles (i.e., $C = \{a_1, a_2, \dots, a_n\}$). The main goal is to extract a subset $A \subseteq C$, where each $a_i \in A$ is relevant to the query Q . To achieve this, we propose an instructional **relevance computation** approach, instructing Large

* These authors contributed equally to this work

Language Models (LLMs) to generate a set of relevance scores S of a group of legal articles A to a given query Q :

$$S = LLMs(Q, A) \quad (1)$$

The proposed prompting technique enables flexibility during ensembling with other retrieval models by offering relevance scores.

This paper thoroughly discusses the implemented experiments and provides an in-depth analysis of the effectiveness, limitations, and real-world applications of our proposed approach. By examining both successful and unsuccessful cases, we aim to identify the strengths and weaknesses of the current methodology. Furthermore, we outline potential directions for future improvements, modifications, and enhancements to the process of legal information retrieval, ultimately contributing to the development of a valuable tool for legal practitioners.

2 Related Work

Legal information retrieval is an important problem in juris-informatics. The methodologies have evolved considerably over time, with methods ranging from simple to complex, and from older to newer approaches [7, 11, 5, 3]. Early studies focused on extracting relevant legal information using information retrieval (IR)-based approaches. For instance, Gao et al. (2019) employed topic keywords to retrieve legal documents relevant to a given situation by using an information retrieval model [1]. However, this method relies heavily on the extracted keywords, which may not be sufficient for representing complex legal texts.

As the research shifted toward using deep learning architectures, attention-based models were proposed to achieve better representations of legal texts. Nguyen et al. (2022) employed attentive deep neural networks to construct two hierarchical network architectures, Attentive CNN and Paraformer, to represent long sentences and articles in legal texts [4]. These architectures utilized attention mechanisms to capture important information in lengthy legal documents.

Building upon these previous works, Hai-Long et al. (2023) introduced a joint learning model to improve the performance of information retrieval and entailment tasks on legal texts [2]. This approach exploited the relationship between relevance and conclusion to obtain better results with multi-task learning.

Moreover, chain-of-thought prompting in large language models has been investigated, as demonstrated by Wei et al. (2022) [10]. This research leveraged the power of large attentive language models to display connected reasoning results, providing an improved way of eliciting reasoning in LLMs for legal text processing.

As the focus shifted to large language models, Sun et al. (2023) explored the potential of using generative LLMs like ChatGPT and GPT-4 for relevance ranking in IR [8]. They showed that these large language models could achieve better performance than conventional supervised methods, even outperforming state-of-the-art models like monoT5-3B on various benchmarks.

3 Prompting-supported Retrieval Pipeline

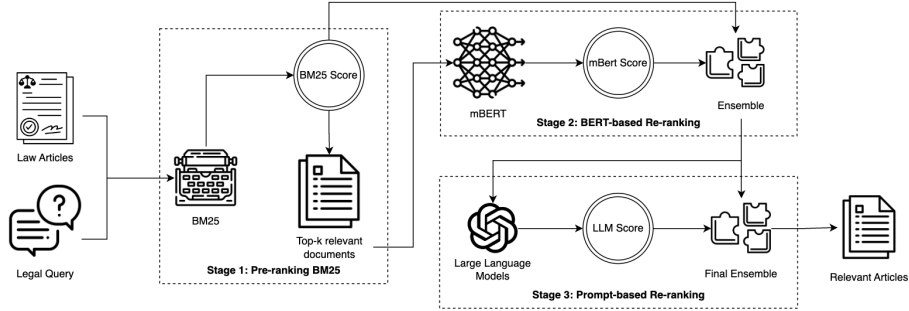


Fig. 1: Prompting-supported Retrieval Pipeline.

Due to the considerable number of articles present in the Legal Statute Corpus, ensuring efficient retrieval time and optimizing computational complexity necessitates the implementation of a multi-phase retrieval system to gradually filter candidates from the corpus. Drawing inspiration from the Retrieval pipeline proposed in [2], the search pipeline in this paper comprises three main phases: Pre-ranking utilizing BM25, Re-ranking phase employing a model based on the BERT architecture, and finally, another Re-ranking phase employing Prompting techniques on large language models (LLMs). Figure 1 visualizes specific steps of the retrieval pipeline.

3.1 Pre-ranking BM25

The selected dataset for experimentation is derived from the COLIEE competition, encompassing a corpus of 768 articles*. Given the substantial number of articles, the primary objective in the initial phase of the model is to ensure both speed and an adequate recall score to cover a significant portion of ground-truth articles. Further refinement to identify truly relevant documents will be conducted in subsequent phases. To meet these requirements, the Okapi BM25 ranking model [6] based on the lexical features is employed to evaluate the relevance score. Leveraging statistical analysis and token frequency computations, the BM25 model exhibits rapid execution and maintains the recall score irrespective of document length or query structure, making it suitable for the specified objectives.

3.2 BERT-based Re-ranking

The utilization of the BM25 [6] model in the pre-ranking phase primarily relies on the lexical feature, emphasizing statistical token frequency without consid-

* <https://www.japaneselawtranslation.go.jp/en/laws/view/3494/en>

ering the semantic similarity between the query and legal articles. Semantic similarity, representing the meaning when combining tokens and putting them in a context, is a crucial feature in the retrieval system. It aids in eliminating documents that, despite lexical relevance to the query, lack semantic relevance. Incorporating semantic similarity can significantly enhance precision scores for the entire retrieval system.

Training the BERT-based Re-ranking model To address semantic similarity, the model based on the BERT architecture with a multi-task learning design, as proposed in [2], has been employed in the re-ranking phase. The arguments and experiments conducted in the aforementioned research have demonstrated that information regarding the correctness of queries and the relevance of a (*query*, *legal document*) pair mutually support each other. Employing a multi-task model to predict both pieces of information enhances the efficiency of the training process and improves the model’s ability to compute semantic correlations [2]. Specifically, the input to the multi-task model consists of a sequence generated by concatenating the query and the article, separated by a special token [SEP]. The output will consist of two main components: a probability (ranging from 0 to 1) indicating the level of semantic relevance between the query and the article, and another probability (ranging from 0 to 1) indicating whether the query is true or false based on the content of the article. The model will be trained based on two labels provided in the COLIEE dataset: relevant articles of the query and labels for the correctness of the query.

Given that the Legal Corpus data from the COLIEE workshop is derived from the Japanese Civil Code, the original language of the dataset is Japanese. Previous research [2, 9, 3] has consistently demonstrated higher performance when experimenting with Japanese datasets compared to English. Therefore, the pre-trained parameters of Multilingual-BERT*, trained on a multilingual dataset including Japanese, are utilized. The utilization of a multilingual language model also facilitates future research and comparisons when employing pipelines with other legal datasets containing various languages.

Ensemble relevance score and inference After the training process, the model will then be utilized for inference in the re-ranking phase. To leverage the strengths of both the BM25 and Multi-Task BERT models, the relevance scores from the two models will be ensemble to calculate the correlation score for the current re-ranking phase. The equation 2 describes how to combine the relevance scores of the BM25 model and the BERT model.

$$R_{reranking-phase-1} = \alpha * BM25_score + \beta * BERT_score \quad (2)$$

whereas α and β are weighting factors for the two models, *BM25_score* represents the correlation score calculated by the BM25 model, and *BERT_score* represents the correlation score calculated by the BERT model. The final correlation score $R_{reranking-phase-1}$ will be normalized to a range from 0 to 1.

* <https://huggingface.co/bert-base-multilingual-cased>

Afterwards, all query-article pairs with a correlation score $R_{reranking-phase-1}$ greater than a predefined *threshold1* will be considered relevant and added to the candidate set for the next phase. An optimal hyperparameter set of $(\alpha, \beta, threshold1)$ will be determined through a grid-search process on the validation set.

The objective of the re-ranking phase using the BERT model is to filter articles that have semantic relevance to the query while still maintaining a relatively high number of relevant candidates according to the ground truth. Specifically, the goal was to retain an F2 score greater than or equal to 0.5 while achieving the highest possible recall score.

3.3 Prompting-based Re-ranking

In the previous retrieval phases, models mainly relied on lexical and semantic similarity. However, in reality, queries may not always directly address the core issue and may require logical reasoning to find relevant legal articles related to the query. Large language models (LLMs) are deep learning models with billions of parameters, trained on vast amounts of data. One notable point is that LLMs can learn to handle downstream tasks through instruction inputs. Therefore, applying Prompting techniques to LLMs can leverage their basic-level reasoning capabilities. Additionally, their broad domain knowledge can help LLMs perform well with queries that do not use specialized legal terminology.

Due to the relatively long length of legal texts, and considering the token limit for input in large language models, typically around 25000 words, we have opted to explore the Zero-shot prompting method to preserve the complete content of both the query and the candidates. The Few-shot method, wherein the LLMs are provided with a set of samples for learning, will be discussed and investigated in detail in subsequent studies.

The input content for the prompting method proposed will include the query content, as well as the content and identifiers (e.g. Article 705, Article 01, ...) of all candidate articles obtained from the previous re-ranking phase using the Multi-Task BERT model. The output of the prompting sentences will include relevance scores for candidates ranging from 0 to 100. The output format specified in the prompting sentence will adhere to JSON format.

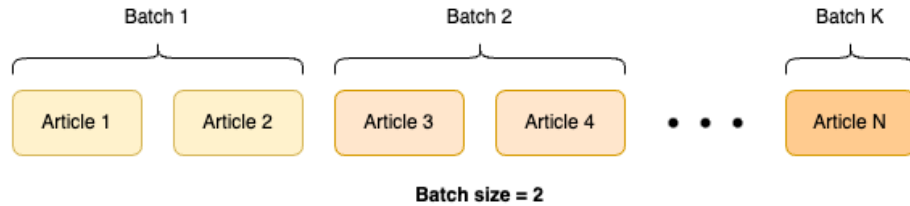


Fig. 2: Visualization of sliding window technique.

Adaptive Sliding window To encompass the complete content of as many candidates as possible within a single prompt, the adaptive sliding window prompting technique has been employed. Specifically, the content of candidates is included in the prompt in its entirety until approaching the token limit, at which point the process is halted. Employing this adaptive sliding window prompting technique minimizes the need for subdividing the prompt for a single query, allowing the LLMs to consider as many candidates simultaneously as possible. An example of this technique is illustrated in Figure 2.

Prompting Template As identified in Section 3.2, given that the original data is in Japanese, to ensure the utmost integrity of information, the Japanese dataset will be utilized for prompting. The prompting template is described below:

```
1 以下は法的なクエリです: {query}
2 以下の記事それぞれにスコアを付けてください。
3 スコアは0から100の間である必要があります。
4 スコアは記事がクエリに関連している確率です。
5
6 これは非常に重要なタスクですので、
7 ゆっくりと時間をかけて最善を尽くしてください。
8
9 以下が記事です: {articles}
10 {format_instructions}
```

The {articles} variable is a list of legal articles, below is an example:

```
1 [{
2   'article_id': '6',
3   'content': '第六条一種又は数種の営業を許された未成年者は、その
               営業に関しては、成年者と同一の行為能力を有する。2\前
               項の場合において、未成年者がその営業に堪えることができない
               事由があるときは、その法定代理人は、第四編（親族）の規定に
               従い、その許可を取り消し、又はこれを制限することができる。'
4 }]
```

The values of the {format_instructions} is described as follows:

```
1 出力は、以下のJSONスキーマに準拠するJSONインスタンスとしてフォー
   マットする必要があります。
2 例として、スキーマ {
3   "properties": {
4     "foo": {
5       "title": "Foo",
6       "description": "a list of strings",
7       "type": "array",
8       "items": {
9         "type": "string"
10      }
11    }
12  }
```



```

12     },
13     "required": [ "foo" ]
14 }
15 の場合、オブジェクト {"foo": ["bar", "baz"]} はスキーマの適切な
    形式です。
16 オブジェクト {"properties": {"foo": ["bar", "baz"]}} は適切な形
    式ではありません。
17 以下が出力のスキーマです:
18 {
19     "properties": {
20         "articles": {
21             "title": "Articles",
22             "description": "記事の選定リスト",
23             "type": "array",
24             "items": { "$ref": "#/definitions/Article" }
25         }
26     },
27     "required": [ "articles" ],
28     "definitions": {
29         "Article": {
30             "title": "Article",
31             "type": "object",
32             "properties": {
33                 "article_id": { "title": "Article Id", "description": "
                    記事の識別子", "type": "string" },
34                 "score": {
35                     "title": "Score",
36                     "description": "記事がクエリに関連している確率、スコア
                        は0から100の間であるべきです。",
37                     "type": "integer"
38                 }
39             },
40             "required": [ "article_id", "score" ]
41         }
42     }
43 }

```

Ensemble with BERT re-ranking relevance score. Since each LLMs model as well as BERT-based re-ranking has its own strengths and weaknesses, the technique of ensembling the relevance scores of each model is carried out and evaluated for effectiveness based on the F2 score of the entire pipeline. To leverage the strengths of both models, their relevance scores are combined with weights using the formula 3.

$$R_{reranking-phase-2} = \beta * BERT_score + \gamma * LLM_score \quad (3)$$

whereas *BERT_score* represents the correlation score of the BERT model, and *LLM_score* represents the correlation score obtained from the prompting re-

sults. β and γ are the respective weights assigned to the BERT and LLMs. Similar to the previous phase, *threshold2* is used to determine which documents are ultimately considered relevant. Likewise, a grid-search process will be conducted on the validation set to identify the optimal set of hyper-parameters.

4 Experiment and Results

4.1 Datasets

COLIEE is an annual competition with the main purpose is nurturing a world-wide legal research community. The competition involves different challenges, covering both case law and statute law systems. The statute law database uses the 768 articles in the Japanese Civil Code. Meanwhile, original queries are questions selected from the Japanese Bar Exam and come along with an English translation version. Every year, COLIEE organizers select new questions from the Exam Bar to construct a new testing set, while testing and training sets of the previous year are merged to form a new training set.

In this work, we validate the proposed pipeline on the COLIEE 2023 dataset. Table 1 presents the statistics of articles’ length in the dataset. On average, each article contains 109 tokens in Japanese and 100 tokens in English. Articles’ length focuses on the range of 75 to 125 tokens, and can vary significantly from 10 tokens to more than 800 tokens as shown in Figure 3a. For statistics of the query corpus, details are reported in Table 2. There are 996 samples in the training set and 101 queries in the testing set, with the range of length starting from 13 tokens to 248 tokens. Figure 3b illustrates the length distribution of queries in the dataset. Most queries contain around 30 to 70 tokens, though some queries have extreme lengths with more than 200 tokens.

	Min	Max	Avg
Japanese	10.0	886.0	109.6
English	5.0	867.0	100.2

Table 1: Statistics of articles’ length in COLIEE 2023.

	Training set	Testing set
Number of queries	996	101
Minimum length of a query	13.00	25.00
Maximum length of a query	248.00	130.00
Average length of a query	62.21	65.28

Table 2: Statistics of queries’ length in COLIEE 2023.

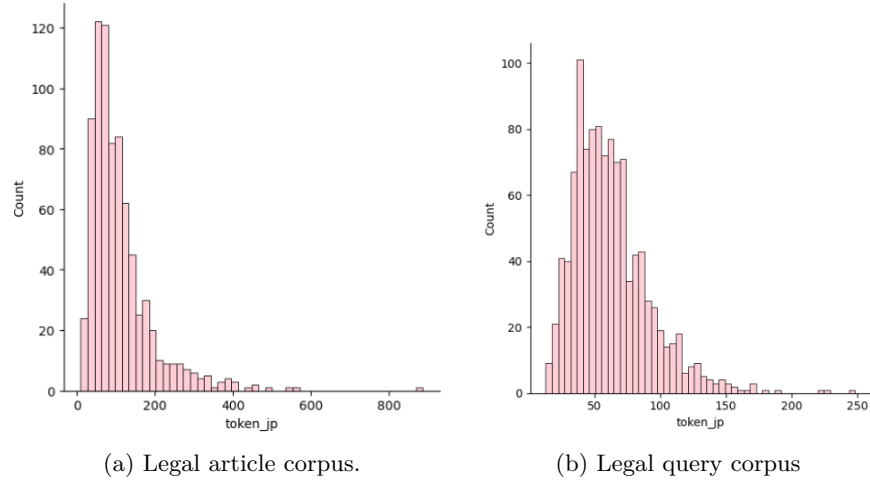


Fig. 3: Length distribution statistics of the COLIEE 2023.

4.2 Experiment Setup

To achieve optimal efficiency, models in each retrieval phase need to be fine-tuned with appropriate parameters to accomplish the goals of each retrieval phase best. During the fine-tuning and pipeline construction process, several hyperparameters need to be adjusted to achieve the most effective search efficiency. Therefore, 20% of the training set samples have been separated to form the validation set. After the training process, the hyperparameters will be adjusted based on the F2 score of the entire pipeline on the validation set. The specific deployment of models in each phase will be described in the following sections.

Pre-ranking with BM25 As mentioned in Section 3.2, the goal of the pre-ranking phase is to select documents with lexical relevance to the query. The expectation for the re-ranking phase is to maximize the removal of irrelevant legal texts while maintaining a relatively high recall score. After obtaining relevance scores from the BM25 model, the top-k documents with the highest relevance scores are chosen as candidates for the subsequent phases.

To select the optimal top-k, a recall score analysis of the BM25 model is conducted on the entire training dataset. Table 3 presents the recall scores of the BM25 model for each top-k value. Based on the table 3, it can be observed that

Top-k	500	400	300	200	100	50	30	10	5
Recall	0.9467	0.9317	0.9186	0.8931	0.8515	0.8050	0.7784	0.7076	0.6444

Table 3: Recall scores of the BM25 model.

choosing a top-k of 500 can eliminate approximately 34.90% of legal documents from the total corpus while still maintaining a suitable recall score of 0.9467.

Re-ranking with Multi-Task BERT-based model Based on the analysis from paper [2], the training process of the Multi-Task BERT model involves using the top 30 candidates from BM25 to form the training set and utilizing the pre-trained parameters of Multilingual-BERT*.

The model, after training, is then utilized for inference in the re-ranking phase. As mentioned in Section 3.2, the optimal parameter set on the validation set ensures the specified requirements: ($\alpha = 0.17$, $\beta = 0.83$, $threshold1 = 0.921$). The results obtained after the BERT-based re-ranking phase on the validation set are $F2 = 0.527$, $Precision = 0.274$, and $Recall = 0.921$. The first row of Table 4 describes the result of the retrieval pipeline that contains two phases BM25 Preranking and BERT-based Re-ranking.

Re-ranking with Prompting LLM The prompting process to obtain relevance scores will be sequentially conducted on three versions of large language models, namely: GPT-3.5-turbo-instruct, GPT-3.5-turbo-1106, and finally, GPT-4-1106-preview, utilizing the API provided by OpenAI*. As described in section 3.3, a grid-search phase is executed on the validation set, resulting in the optimal set of hyperparameters being ($\beta = 0.5$, $\gamma = 0.5$, $threshold2 = 0.52$) for ensembling the relevance score of BERT-based reranking model and the score of prompting process.

4.3 Experiment Result

After the experimentation process, Table 4 describes the obtained results of the retrieval system with different settings and various large language models. The first row (1) of the table depicts the results of the retrieval system consisting of only two phases: pre-ranking with BM25 and re-ranking with the BERT-based model. Rows (2), (3), (4) and (5) show the results of the retrieval system with all three phases, where the final phase uses *gpt-3.5-turbo-instruct* (for row 2), *gpt-3.5-turbo-1106* (for row 3), and *gpt-4.0-1105* (for both row 4 and 5). The pipeline’s last phase used at row (2), (3) and (4) has the relevance scores only derived from LLMs. Meanwhile, the last phase of the pipeline used in row (5) has relevance scores that are ensembled from the LLMs’ relevance scores and re-ranking BERT’s relevance scores (which were derived at previous phase).

Comparing the retrieval system (1) with the remaining systems, it is evident that the prompting-based re-ranking phase significantly increases precision scores on both the test and validation sets. The retrieval systems (1), (2), and (3) make significant improvements of 0.1346, 0.1808, and 0.4826 in terms of precision on the testing set compared to the baseline (1). It can be observed that the final retrieval phase does not considerably decrease the recall score, leading to a

* <https://huggingface.co/bert-base-multilingual-cased>

* <https://openai.com>

Retrieval System	Validation set			Testing set		
	F2	P	R	F2	P	R
BM25 + BERT-reranking (1)	0.5000	0.2620	0.9116	0.5268	0.2738	0.9207
gpt-3.5-turbo-instruct (2)	0.5969	0.3600	0.8674	0.6213	0.4084	0.8514
gpt-3.5-turbo-1106 (3)	0.6649	0.5467	0.7666	0.6620	0.4546	0.8613
gpt-4-1106-preview (4)	0.7621	0.7841	0.7799	0.7647	0.7564	0.7871
gpt-4-1106-preview ensemble (5)	0.7678	0.7029	0.8308	0.8085	0.7277	0.8712

Table 4: Experiment results on COLIEE 2023.

substantial improvement in the F2 scores of the query systems. Particularly, the F2 score of the retrieval system (4) is enhanced by 0.2379 from 0.5268 to 0.7647 compared to the baseline (1). Indeed, this result indicates the effectiveness of our proposed retrieval pipeline in improving the accuracy of retrieved articles.

When comparing retrieval systems (4) and (5), it can be observed that combining the scores of the BERT-based re-ranking model with the scores from the LLMs prompting output, although slightly decreasing precision from 0.7564 to 0.7277, maintains recall scores from the two preceding phases. Specifically, the recall score of the retrieval system (5) is 0.8712, which is higher by roughly 8% compared to the retrieval system (4). Therefore, the F2 score of the retrieval system (5) improves by 4% overall. Through this observation, it is evident that ensemble relevance scores of the two models have been effective in combining the strengths of both language models: the BERT-based model with the ability to understand semantic similarity between legal documents, and the LLMs with basic logical inference capabilities across a broad knowledge domain.

Team	F2	P	R
CAPTAIN	0.7645	0.7333	0.8000
JNLP	0.7526	0.6517	0.8300
NOWJ	0.7345	0.6892	0.7750
HUKB	0.6793	0.6342	0.7150
LLNTU	0.6600	0.7400	0.6500
UA	0.5698	0.6267	0.5700
Our system			
gpt-4-1106-preview (4)	0.7647	0.7564	0.7871
gpt-4-1106-preview-ensemble (5)	0.8085	0.7277	0.8712

Table 5: Results of participating teams in the COLIEE 2023.

Table 5 presents the best runs of participating teams in the COLIEE 2023. The retrieval system (5) achieves an F2 score of 0.8085, which outperforms the best team in the competition by 4%. Both our proposed retrieval systems using the GPT-4 model surpass participants in precision and recall scores. Indeed,

these results emphasize the advantages of LLMs and prompting techniques in the retrieval task.

4.4 Error Analysis

Overview

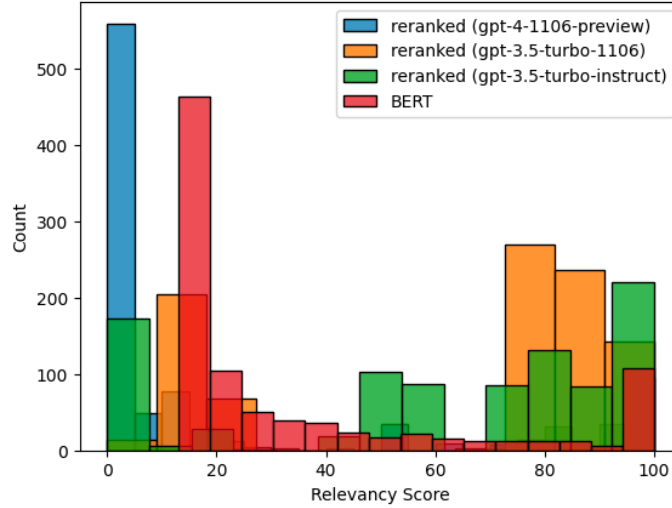


Fig. 4: Relevancy score distribution on test set before and after re-ranking with different LLMs.

	Precision	Recall	F2
Increased	77 (avg. +0.6508)	0	69 (avg. +0.4298)
Unchanged	13	81	13
Decreased	11 (avg. -0.1253)	20 (avg. -0.6749)	19 (avg. -0.2960)

Table 6: Count and changes in precision, recall, and F2 by query.

Relevancy scores from BERT-based reranking and Prompting-based reranking’s distribution of test set relevancy score are illustrated in 4. From the figure, it can be observed that gpt-3.5-turbo-1106 and gpt-3.5-turbo-instruct often give higher scores, eliminating fewer candidates. Meanwhile, gpt-4-1106-preview shares a similar distribution to BERT re-ranking, with most candidates eliminated with a score of 0.

Detailed changes in F2, precision, and recall per legal query are provided in Table 6. It is noticeable that Prompting Re-ranking often leads to drastic changes

in both precision and recall, while it can help significantly increase precision in most cases, it comes with sacrificing an average of 0.67 recall in one-fifth of the test set. This highlights the limitations and trade-offs of this approach.

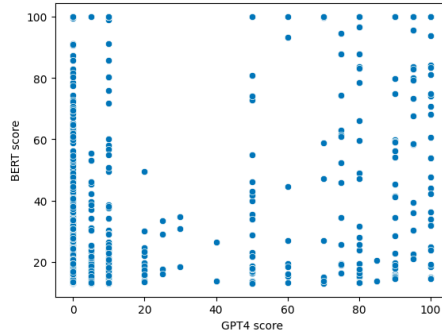
Complicated samples (low precision and/or low recall)

From outputs in the test set, this analysis aims to look for areas where LLMs struggle the most while re-ranking and discussing potential future research directions. The statistics in this section are taken from re-ranking outputs of only gpt-4-1106-preview without ensembling.

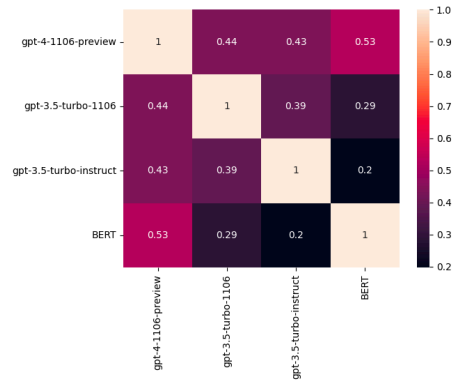
Many noisy candidates: In cases where precision and recall are both 0, it is often due to the only matching articles getting eliminated, resulting in 0 precision and 0 recall at the same time. It seems that the model struggles to improve in cases where many noisy candidates are present. 14/101 samples were not able to recall any matching article, all of them have low precision and a high number of candidates from the previous stage, with up to 41 candidates for re-ranking averaging only 9% precision. Around 3% of samples already have 0 recall after the BERT re-ranking phase which LLMs re-ranking can not improve. To reduce noisy candidates from the BERT re-ranking phase, more improvements to the first two retrieval phases (BM25 and BERT re-ranking) are needed. This can be through techniques such as query extension, improved training regimes for the cross-encoder model, or other pre- and post-processing techniques.

Complex legal situations with actors: Half of the low recall queries (recall ≤ 0.5) or queries that saw recall decline after LLMs re-ranking are complex legal situations involving 2 or more actors. This highlights the challenges these models still face with complex reasoning and more techniques should be investigated to tackle this limitation.

LLMs vs BERT



(a) Scatterplot of GPT-4 scores against relevance scores derived from BERT-based re-ranking model.



(b) Correlation between models' scores.

Fig. 5: Correlation Analysis

To further understand the differences between LLMs’ outputs and BERT’s, GPT4’s scores are plotted against BERT’s scores in 5a and the correlation coefficient is calculated between models in 5b. Despite sharing a similar distribution as shown in 4, outputs from gpt-4-1106-preview and BERT show no significant correlation with Pearson correlation of only 0.532 between the two score sets.

This shows that GPT and BERT re-ranking models have notable differences in outputs and are good at different scenarios. This partly explains why when gpt-4-1106-preview and BERT are ensembled together, despite large F2 discrepancies shown in 4, still lead to a sizable increase in F2—these 2 models when ensembled correct each other’s biases hence leading to a more robust output.

5 Conclusion

The experiment in constructing a three-phase query system has demonstrated that, if implemented and utilized wisely, a large language model combined with prompting techniques can be fully leveraged to enhance the precision and recall in the search process. Simultaneously, analyzing query system errors has continued to uncover several unresolved weaknesses, including the complexity of queries and semantic interconnections among legal texts. These challenges will be the focal points of future research to develop a comprehensive solution for retrieval tasks in the legal data domain.

Acknowledgement

Hai-Long Nguyen was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.ThS.075.

This work was supported by JSPS KAKENHI Grant Numbers, JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Gao, J., Ning, H., Sun, H., Liu, R., Han, Z., Kong, L., Qi, H.: Fire2019@ aila: Legal retrieval based on information retrieval model. In: FIRE (Working Notes). pp. 64–69 (2019)
2. Hai Long, N., Vuong, T.H.Y., Nguyen, H.T., Phan, X.H.: Joint learning for legal text retrieval and textual entailment: Leveraging the relationship between relevancy and affirmation. In: Preoŧiuc-Pietro, D., Goanta, C., Chalkidis, I., Barrett, L., Spanakis, G.J., Aletras, N. (eds.) Proceedings of the Natural Legal Language Processing Workshop 2023. pp. 192–201. Association for Computational Linguistics, Singapore (Dec 2023). <https://doi.org/10.18653/v1/2023.nllp-1.19>, <https://aclanthology.org/2023.nllp-1.19>
3. Nguyen, C., Nguyen, P., Tran, T., Nguyen, D., Trieu, A., Pham, T., Dang, A., Nguyen, L.M.: Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks. arXiv preprint arXiv:2401.03551 (2024)

4. Nguyen, H.T., Phi, M.K., Ngo, X.B., Tran, V., Nguyen, L.M., Tu, M.P.: Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law* pp. 1–30 (2022)
5. Nguyen, H.L., Nguyen, D.Q., Nguyen, H.T., Pham, T.T., Nguyen, H.D., Nguyen, T.A., Nguyen, H.T., et al.: Neco@ alqac 2023: Legal domain knowledge acquisition for low-resource languages through data enrichment. In: 2023 15th International Conference on Knowledge and Systems Engineering (KSE). pp. 1–6. IEEE (2023)
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. Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., Ren, Z.: Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542* (2023)
9. Vuong, T.H.Y., Nguyen, H.L., Nguyen, T.M., Nguyen, H.T., Nguyen, T.B., Nguyen, H.T.: Nowj at coliee 2023: Multi-task and ensemble approaches in legal information processing. *The Review of Socionetwork Strategies* pp. 1–21 (2024)
10. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **35**, 24824–24837 (2022)
11. Yoshioka, M., Suzuki, Y., Aoki, Y.: Hukb at the coliee 2022 statute law task. In: *JSAI International Symposium on Artificial Intelligence*. pp. 109–124. Springer (2022)

An Empirical Evaluation of Using ChatGPT to Summarize Disputes for Recommending Similar Labor and Employment Cases in Chinese

Po-Hsien Wu[‡]

Chao-Lin Liu[†]

Wei-Jie Li[‡]

National Chengchi University, Taiwan

{chaolin[†], 111753120[‡], 110753128[‡]}@nccu.edu.tw

Abstract. We present a hybrid mechanism for recommending similar cases of labor and employment litigations. The classifier determines the similarity based on the itemized disputes of the two cases, that the courts prepared. We cluster the disputes, compute the cosine similarity between the disputes, and use the results as the features for the classification tasks. Experimental results indicate that this hybrid approach outperformed our previous system, which considered only the information about the clusters of the disputes. We replaced the disputes that were prepared by the courts with the itemized disputes that were generated by GPT-3.5 and GPT-4, and repeated the same experiments. Using the disputes generated by GPT-4 led to better results. Although our classifier did not perform as well when using the disputes that the ChatGPT generated, the results were satisfactory. Hence, we hope that the future large-language models will become practically useful.

Keywords: civil cases, similar cases recommendation, machine learning, large language models, semantic clustering, semantic classification, convolutional neural networks.

1 Introduction

Searching and finding similar previous cases are the basis for various possible applications in the legal domain. Similar previous cases may provide clues about how judges may judge a new case and what lawyers can do in oral arguments and cross examinations [4][7]. The research for searching similar previous cases started many years ago, and has attracted the attention of many researchers in recent years.

We will mention some closely related issues now, but we cannot offer a comprehensive survey for the research of recommending similar previous cases. Researchers have explored two possible research directions. Some suggested that the citation-network-based method may be less applicable to the legal domain [5], and we and many others took the text-based approach [11]. In the text-based camp, researchers must determine how they define the similarity between two cases. This includes at least two issues. First, how do we define similarity between two cases numerically? Second, do we compare every statement in the two texts as a whole, or do we compare selected parts partially [11]? With the support from advanced artificial intelligence techniques, it is

possible to consider more specific background information about the lawsuits in comparing the cases [2]. Low-resource remains a challenge for similar case recommendation research. CAIL-2019 [18], which represents the national corpus of China, contains slightly more than 8000 labeled pairs of cases [2][3].

In previous work, we explored using a clustering-based method for recommending similar cases [10]. Two cases were judged to be similar if they had similar distributions in the clusters of disputes. Hence, we have a narrow but practical standard of similarity. The disputes between the employees and the employers are the main source of the litigations. The scale of our experiments was thus limited by the number of cases that explicitly listed disputes between the litigants. In this paper, we report the results of our classification of whether or not two cases are similar. We built the classifiers with deep-learning approaches, and judged whether the two given cases were similar based on the disputes between the litigants of civil cases, i.e., the labor and employment cases. This can be achieved with our previous dataset.

With the availability of ChatGPT, we evaluated the disputes that we asked the ChatGPT to generate from the litigants’ claims. If ChatGPT would serve as a reliable source of itemized disputes, we could expand the scale of our experiments [19]. We replaced the GPT-generated disputes with the original disputes in our classification experiments. If the new outcomes are satisfactory, we may consider using ChatGPT to summarize the disputes of the litigants for cases that do not have itemized disputes, thus alleviating the problems of low resources for our approach.

We observed encouraging results when using GPT-3.5 and GPT-4. In the remainder of this paper, we offer a formal definition of our research problem in Section 2. We provide more background information about our data in Section 3, and explain how we used ChatGPT to summarize the litigants’ claims for us in Section 4. We elaborate on the design of the experiments in Section 5, and report the experimental results in Section 6. We then wrap up with some discussions in Section 7.

2 Problem Definition

We evaluate the potential contribution of ChatGPT in comparing the similarity of two labor and employment cases in Chinese.

In a previous work [10], we evaluated clustering-based methods for recommending similar labor and employment cases. Since we have labeled some case pairs by their similarity, we could train some classifiers with these annotated cases, and use the trained classifiers to guess whether two future cases are similar.

The real challenge is how we and the annotators could have determined whether the two cases were similar. It is not easy to define “similar” even in everyday life, let alone in judicial cases. In our work, we focused on whether the disputes between the employees and employers in the two cases are similar. This is certainly not the only way to define “similarity” between two cases. For instance, one might be more interested in which party won the cases or in the industry and years of the cases.

Based on the narrowed perspective of similarity, we can define our work in the following way. Assume that we have collected the judgment documents of m previous cases that explicitly included the disputes between the employees and the employers. We denote this collection as $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$. Let $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$, where K_i

denote the descriptions of disputes listed in the judgment documents for case C_i . We annotated a collection of case pairs, C_i and C_j , $i \neq j$, with their degrees of similarity $S_{i,j}$, where $S_{i,j}$ can be a label in $\mathcal{S} = \{\text{"similar"}, \text{"barely similar"}, \text{"not similar"}\}$. We denote the collection of annotated case pairs and their similarity as $\mathcal{D} = \{(C_i, C_j, S_{i,j}) | C_i \in \mathcal{C}, C_j \in \mathcal{C}, S_{i,j} \in \mathcal{S}\}$. Assume that we have n pairs of labeled cases. We may denote the annotated collection as $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$. Namely, a D_k represents a $(C_i, C_j, S_{i,j})$ in $\mathcal{D}$ for certain i and j .

We split $\mathcal{D}$ into two mutually exclusive subsets, one for training and one for testing. We denote the training and testing subsets as $\mathcal{R}$ and $\mathcal{E}$, respectively, and $\mathcal{D} = \mathcal{R} \cup \mathcal{E}$. We trained classifiers with machine learning methods and with instances in $\mathcal{R}$, and evaluated the quality of the trained classifiers with the instances in $\mathcal{E}$. Let f denote the classification function of a trained classifier. When given a case pair, f will return the similarity of the case pair. More specifically, if $(C_i, C_j, S_{i,j}) \in \mathcal{E}$, we hope that $f(C_i, C_j) \equiv f(K_i, K_j) = S_{i,j}$.

3 Data Source, Selection and Preprocessing

We provide information about the data sources, discuss some basic statistics of our data, and introduce an important step in the preprocessing procedure in this section. We have described the major steps for data preparation in previous papers in JURIX [8] and JURISIN [9] and the legalAIIA workshop in ICAIL [10]. We will not repeat all of the details to avoid the concerns of self-plagiarism.

3.1 Data Source: Taiwan Judicial Yuan

We obtained the judicial documents from an open repository that is maintained by the Judicial Yuan. The Judicial Yuan, the highest governing body for Taiwan's judicial system, oversees the publication of judgment documents from various courts, including local, high, supreme, and special courts. These documents are typically released on the TWJY website three months after the judgment date, with February judgments, for example, becoming available in May. Some documents may not be published due to legal reasons, such as protecting minors or litigants, and their contents are anonymized for privacy.

As of July 2023, the TWJY website hosts approximately 18.7 million documents, dating back to January 1996. Initially, only documents from a limited number of special courts were available in the first few years, with broader coverage starting from 2000 onward. The Judicial Yuan updates the website monthly, with a three-month lag, ensuring that users can access and download new judgment documents from various courts in a compressed file. However, the number of available documents may fluctuate due to retractions based on legal reasons. Anonymization of published documents is mandated by law, and the government takes responsibility for safeguarding the privacy of individuals involved in lawsuits.

3.2 Data Selection: Cases with Listed Disputes

Each document in the TWJY is a JSON file and adopts a common top-level structure. The structure consists of seven fields: JID is the long identification number; JYEAR is the year when the

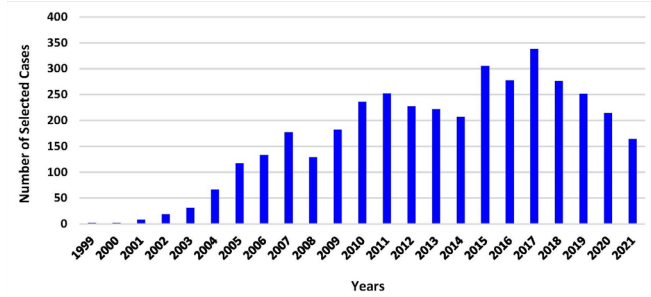


Figure 1: The temporal distribution of the selected cases [10]

case occurred in terms of Taiwan calendar; JCASE is the abbreviated code for the type of the lawsuit; JNO is the short identification number for the lawsuit; JDATE is the date for the current judgment in terms of the Western calendar; JTITLE is the category of the lawsuit, and JFULL is the full text for the judgment document.

Therefore, using the contents of JCASE and JTITLE fields to find judgment documents of labor and employment litigations, among a myriad of case categories, is a basic step. We focus on the judgments of the local courts, where the judges would consider the factual parts of the lawsuits, and using the codes in the JCASE and the JTITLE fields could help us exclude appeal cases.

The filtering of relevant and usable documents needs more steps. Sometimes, even when the JCASE and JTITLE fields seem to qualify a document, we may find clues in the JFULL field that indicate the case does not meet our needs.

Most importantly, in the current study, we look for cases in which the courts explicitly recorded the disputes between the plaintiffs and the defendants in the judgment document. We look into the JFULL field to ensure the documents meet this requirement. The listed disputes provide important information about the lawsuits, and help the lawsuits to proceed more effectively. Nevertheless, not all of the judgment documents would record the disputes.

At this moment, we found 3835 cases from 21 local courts in Taiwan. Figure 1 shows the distribution over the years when the cases took place. The horizontal axis shows the years, and the vertical axis shows the number of cases. The long-term trend is that the number of cases has increased. In addition, most of the selected cases came from the top five courts, which happened to be located in metropolitan or industrialized areas [10].

3.3 Disputes in Labor and Employment Cases

For each of the judgment documents we selected in Section 3.2, there is a section that itemizes the disputes between the plaintiffs and the defendants. The format looks like the following, although the exact formats may vary, and they are actually in Chinese. (See Appendix A for real examples.)

The disputes are as follows.

dispute-1 statement

dispute-2 statement

...

The challenge is that the statements contain specific information about their belonging cases, e.g., person names, place names, and time expressions. Comparing these named entities between two litigations may not make very much sense.

For this reason, our programs would recognize using the NER techniques (named entity recognition) to identify the named entities. They would replace those specific nouns (or noun phrases) with more general terms, e.g., someone, somewhere, and some time. This would make the dispute statements more comparable. We refer to this as a “blurring” step. (See Appendix A for real examples.)

4 Summarizing Disputes with ChatGPT

Large language models and generative AI have shown their impressive applicability in a wide range of tasks in the past year, including tasks that are in the legal domain [16]. ChatGPT of OpenAI is perhaps the most famous leader for this ground-breaking development. One main concern of applying the generative technology was the hallucination problem. There is a case in which a legal practitioner used the ChatGPT to generate a legal document that cited a non-existing document [15]. The research community has been very concerned about this annoying problem very intensively.

As users, we can avoid the hallucination problem using prompts with specific constraints. In a typical judgment document, we can find statements of the plaintiffs and the statements of the defendants. A list of disputes between the plaintiffs and the defendants will follow, if the document has the list. We extracted and used the statements of the plaintiffs and the statements of the defendants from the judgment documents in our prompts.

Since our goal was to ask the ChatGPT to summarize the disputes between the plaintiffs and the defendants, we must try to confine the ChatGPT to finding the disputes only from the statements of the plaintiffs and the defendants. To this end, our prompts consist of three steps. First, we ask ChatGPT to summarize the plaintiffs' statements with a list of items. Second, we ask the ChatGPT to summarize the defendants' statements with a list of the items. Then, we asked ChatGPT to itemize the disputes between the plaintiffs and the defendants based on the two lists it returned in the previous two steps. Upon the request of the reviewers, we provide the segment of our Python code for this three-step operation in Appendix B.

The disputes that we received would look like the examples that we explained in Section 3.3, and we could use them in our experiments as if they were the disputes that were listed in the original judgment documents.

We have implemented the ideas with GPT-3.5 and GPT-4 (0613). We set the parameters temperature to 0.7 and 0.3 for GPT-3.5 and GPT-4, respectively, to avoid over-generation. Currently, both ChatGPT versions limit the number of tokens for the conversations, so we confine the number of tokens of our prompts to 11500 and 6000 for

GPT-3.5 and GPT-4, respectively. We would drop the cases if the lengths

Source	Cases	Sent.	Labeled Pairs	Labeled Cases
TWJY	3835	12385	2288 (1360, 928)	1543
GPT-3.5	3771	14278	2277 (1355, 922)	1541
GPT-4.0	3826	16878	2288 (1360, 928)	1543

Table 1. Basic statistics of the source corpora.

of the statements of the plaintiffs or the defendants exceeded these limits.

In addition, we might not receive the summaries or the disputes from the ChatGPT sometimes. When this unpredictable problem occurred, we resubmitted the prompts at most three times. If the problem persisted, we would drop the case as well. As a result, we started with 3835 judgment documents (cf. Section 3.2), and obtained disputes of 3771 and 3826 documents from GPT-3.5 and GPT-4, respectively.

5 Classification for Similar Case Recommendations

We reported some results of applying clustering-based methods to recommend similar cases in [10]. In this presentation, we reported the results of using classification-based methods for the recommendation task. We will report the resulting differences in the quality of the recommendations in Section 6.

5.1 Data for Training, Validation, and Test

Table 1 provides more statistics about the judgment documents used in the current work. We have 12,385 dispute statements in the selected cases from the TWJY. We have annotated 2288 pairs of similar cases: 1360 not-similar pairs and 928 similar pairs [10]. Only 1543 of the 3835 cases were part of the labeled pairs of cases. We did not have $2 \times 3835 = 7670$ labeled cases because some labeled pairs shared a common case.

Since all of these 1543 cases were not used in finetuning the BERT models (cf. Section 5.2), there would not be data leakage problems for embedding. In addition, we used cosine similarity in the classifiers, so using the same case in different pairs in the training and test would not cause the problem of data leakage.

Due to the limitations of GPT-3.5, we did not obtain the dispute statements for cases when we used GPT-3.5 (cf. Section 4). Hence, we had fewer labeled pairs and cases when we used GPT-3.5 in our experiments.

5.2 Text Embedding and Fine-Tuning the Pretrained BERT models

We vectorized the statements of disputes only with the TfidfVectorizer of scikit-learn [10]. Now, we vectorized the statements with the Sentence-BERT [13] with either the Lawformer [17] or with the Chinese RoBERTa [1].

The BERT models are both for Chinese, only that the Lawformer is specifically pre-trained with legal documents of China that were recorded with simplified Chinese. Hence, in our experiments, we may choose to fine-tune the BERT models with the legal documents of Taiwan, which were recorded in traditional Chinese. There are some

variations between the simplified and traditional Chinese and intricate differences between the legal terms used in China and Taiwan.

We fine-tuned the Lawformer and RoBERTa along with the Sentence-BERT. We first put aside the sentences in the 1543 labeled cases from the sentences in the 3835 TWJY cases for training and testing our classifiers. We then clustered the remaining 5321 sentences with a typical density-based clustering method, and we disregarded all the sentences in small clusters that contained less than ten sentences. The result is 3031 sentences.

We created pairs of sentences for fine-tuning the Sentence-BERT as follows. First, we created pairs of sentences that belonged to the same clusters, and assigned this type of pairs to the category “same”. Then, we created pairs of sentences that belonged to different clusters, and assigned this type of pairs to the category “diff”. Ultimately, we created a database of 3875115 “diff” pairs and 716850 “same” pairs. We randomly sample only 50000 from each category to fine-tune the Sentence-BERT for efficiency and effectiveness.

5.3 Transforming the Text for Convolutional Neural Networks

Without loss of generality, we will consider the task to determine whether two cases C_i and C_j are similar in the following deliberation. Following the notation that we introduced in Section 2, we compute the cosine similarity between the disputes in K_i and K_j . Let $K_i = \{k_{i,1}, k_{i,2}, \dots, k_{i,\alpha}, \dots, k_{i,u}\}$ and $K_j = \{k_{j,1}, k_{j,2}, \dots, k_{j,\beta}, \dots, k_{j,v}\}$. The cosine similarity $s_{\alpha,\beta}^{i,j}$ between a given pair of disputes, $k_{i,\alpha}$ from K_i and $k_{j,\beta}$ from K_j , is defined in (1), where $v_{i,\alpha}$ and $v_{j,\beta}$ denote the Sentence-BERT vectors of $k_{i,\alpha}$ and $k_{j,\beta}$, respectively. Theoretically, the range of $s_{\alpha,\beta}^{i,j}$ is $[-1,1]$, but the majority were in the range of $[0,1]$, and $s_{\alpha,\beta}^{i,j} = 1$ only when $k_{i,\alpha} = k_{j,\beta}$.

$$s_{\alpha,\beta}^{i,j} = \text{cosine_similarity}(v_{i,\alpha}, v_{j,\beta}) \quad (1)$$

We construct a matrix $M_{i,j}$ for C_i and C_j , and an element $m_{\alpha,\beta}^{i,j}$ in $M_{i,j}$ is set to $1 - s_{\alpha,\beta}^{i,j}$. Namely, the element $m_{\alpha,\beta}^{i,j}$ in $M_{i,j}$ at the position (α, β) is the similarity between the α th and β th disputes of C_i and C_j , respectively.

We still clustered all of the disputes of the cases in $\mathcal{C}$, as we did and explained in [10], except that we switched to using the HDBSCAN of scikit-learn for clustering.¹ Let $m_{max}^{i,j}$ and $m_{min}^{i,j}$ denote the largest and the smallest values in $M_{i,j}$, respectively. We set the parameter `cluster_select_epison`, ϵ , for HDBSCAN to $m_{min}^{i,j} + 0.8 \times (m_{max}^{i,j} - m_{min}^{i,j})$ to control the number of clusters that HDBSCAN may produce. Let $\mathbb{C} = \{c_1, c_2, \dots, c_\gamma\}$ denote the resulting clusters that contained all of the disputes in $\mathbf{K}$. The actual values of γ varied in different experiments and were determined automatically by HDBSCAN. For convenience of communication, we set the elements in $\mathbb{C}$ to integers. Namely, $c_1 = 1, c_2 = 2, \dots$, and $c_\gamma = \gamma$. After the clustering step, each dispute

¹ <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.HDBSCAN.html>

$k_{i,\alpha}$ in a $K_i \in \mathbf{K}$ will be assigned to a certain cluster code $c_{i,\alpha} = c \in \mathbb{C}$. Let $\mathbf{K} = \{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_m\}$, where $\mathbf{K}_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,|K_i|}\}$ and $|K_i|$ is the number of disputes in K_i . We evaluated the idea that whether we may judge two cases C_i and C_j were similar given the size of $\mathbf{K}_i \cap \mathbf{K}_j$ in [10].

We then created a square matrix $Z^{i,j}$ for C_i and C_j in the following way. Notice that we let $K_i = \{k_{i,1}, k_{i,2}, \dots, k_{i,\alpha}, \dots, k_{i,u}\}$ and that each $k_{i,\alpha} \in K_i$ has been assigned a cluster code $c_{i,\alpha}$. Hence, we could reorder the disputes in K_i by their cluster codes. Let K'_i in (2) denote such a reordered version of K_i . We reordered the disputes in $K_j = \{k_{j,1}, k_{j,2}, \dots, k_{j,\beta}, \dots, k_{j,v}\}$ as well, so we have the reordered K'_j in (3).

$$K'_i = \{k'_{i,1}, k'_{i,2}, \dots, k'_{i,\alpha}, \dots, k'_{i,u}\} \quad (2)$$

$$K'_j = \{k'_{j,1}, k'_{j,2}, \dots, k'_{j,\beta}, \dots, k'_{j,v}\} \quad (3)$$

We concatenate K'_i and K'_j to form $K^{i,j}$ as shown in (4). By this concatenation step, we could create a squared matrix that would facilitate our operations with the convolutional neural networks. The value of the element $z_{x,y}^{i,j}$ in $Z^{i,j}$ at position (x, y) is the cosine similarity between the x th and y th elements in $K^{i,j}$, counting from the left to the right. Therefore, the size of $Z^{i,j}$ is $(|K_i| + |K_j|) \times (|K_i| + |K_j|)$, in general.

$$K^{i,j} = \{k'_{i,1}, k'_{i,2}, \dots, k'_{i,u}, k'_{j,1}, k'_{j,2}, \dots, k'_{j,v}\} \quad (4)$$

Finally, let $z_{max}^{i,j}$ and $z_{min}^{i,j}$ denote the largest and the smallest values in $Z^{i,j}$, respectively. We projected a value of $z_{x,y}^{i,j}$ in $Z^{i,j}$ to the range $[0, 255]$ with formula (5). This step practically created a squared grey-level image. Since the selected ϵ is closer to $z_{maz}^{i,j}$ than to the average of $z_{min}^{i,j}$ and $z_{maz}^{i,j}$, giving a half of $[0, 255]$, i.e., $[128, 255]$, to the range of $[\epsilon, z_{maz}^{i,j}]$ is to monitor the range $[\epsilon, z_{maz}^{i,j}]$ with a higher resolution.

$$\begin{aligned} z_{x,y}^{i,j} &\leftarrow \frac{z_{x,y}^{i,j} - z_{min}^{i,j}}{\epsilon - z_{min}^{i,j}} \times 127, \text{ if } z_{x,y}^{i,j} \leq \epsilon \\ z_{x,y}^{i,j} &\leftarrow 127 + \frac{z_{x,y}^{i,j} - \epsilon}{z_{maz}^{i,j} - \epsilon} \times 128, \text{ if } z_{x,y}^{i,j} > \epsilon \end{aligned} \quad (5)$$

We would evaluate whether the grey-level images would be useful for predicting whether or not cases C_i and C_j are similar. The reordering operations, i.e., the steps to create (2) and (3), allow us to figure out the clusters that included more dispute statements, thereby creating contextual information. Figure 2 shows a sample image for similar and a sample image for dissimilar case pairs. (The bright diagonal from the upper left to the lower right corners was a natural result of disputes being compared with themselves.)

5.4 Training the Classifier

Figure 3 shows the flow of using a simple procedure with the convolutional neural network (CNN) layers and the max-pooling layers for the classification task. We scaled a given $Z^{i,j}$ to the size of 32×32 . The selection of 32 was based on the number of

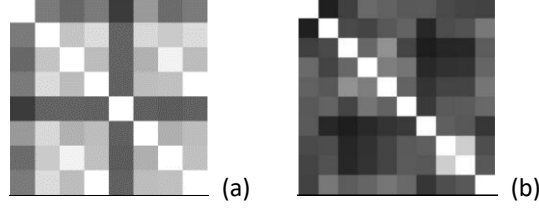


Figure 2: The grey-level images offer a special perspective for the similarity between the cases: (a) a similar pair (b) a dissimilar pair

disputes listed in the judgment documents. This image would go through two iterations of CNN and max-pooling. In each iteration, we used 32 CNN units, and the convolution mask was 3×3 . We used a drop-out layer of 0.2 in between the CNN and the max-pooling layers. Then, the image would be shrunk by the 2×2 max-pooling step. Then, after the two iterations, an 8×8 matrix would be flattened and used as the input to a fully connected layer to produce the classification results.

We trained the classifier with the data that we listed in Table 1. We used 64% of the labeled pairs for training and 16% for validating the classification models. We reserved 20% for the final test. We split the data with a stratification process.

6 Empirical Evaluations

We conducted experiments to answer two questions. The first is whether we can use ChatGPT to summarize and itemize the disputes for the labor and employment litigations. The second is to examine whether or not and how to use the Lawformer in our studies. More specifically, should we use an ordinary Chinese RoBERTa or Lawformer for embedding? What if we fine-tune both BERT models?

Since random numbers play important roles in the operations of artificial neural networks (deep learning), we repeated each of our experiments 30 times to draw a boxplot for the experiment. The training, validation, and test data were resampled every time. Since the BERT models were fine-tuned with data that were completely different from the data that we listed in Table 1, we did not repeat the finetuning step.

After completing the training process, we asked the classifiers to predict whether or not the pairs of cases in the reserved 20% of data were similar. We only fine-tuned our BERT model once, but re-split the labeled pairs of cases 30 times so that we could make boxplots for the results. We can create boxplots for both the F1 measure and the traditional accuracy.

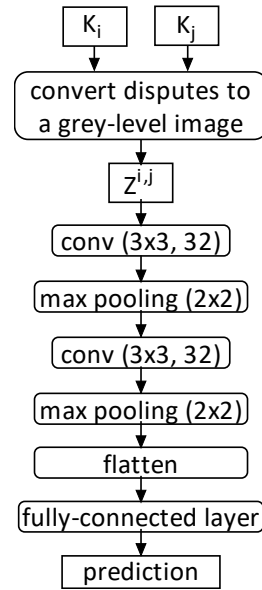


Figure 3: The CNN classifier

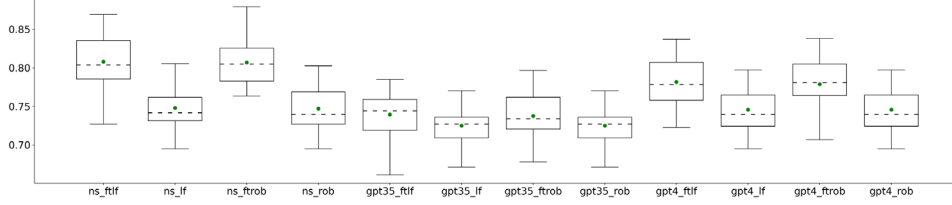


Figure 4: F1 measures of the classification with the CNN classifier

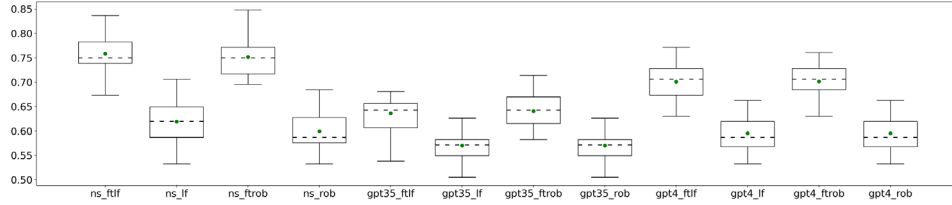


Figure 5: Accuracy of the classification with the CNN classifier

Figure 4 shows the boxplots for the F1 measures of 12 experiments that we have completed. The horizontal axis shows the codes for the experiment. The prefix “ns” indicates that the disputes were from the TWJY, and “gtp35” and “gtp4” indicate that the disputes were from GPT-3.5 and GPT-4, respectively. The suffix “ft” indicates that the BERT model was fine-tuned, “lf” indicates that the Lawformer was used for embedding, and “rob” indicates that the RoBERTa was used for embedding. The green dots and the orange segments in each boxplot mark the means and medians of the classification results, respectively. The vertical axis shows the F1 measures of the experiments. Putting the prefix and suffix together, “ns_ftlf” and “ns_ftrob” indicate that we fine-tuned the Lawformer and RoBERTa, respectively.

Using the Lawformer may provide better results than using the ordinary Chinese RoBERTa for us, but the differences were only obvious when we used the data in TWJY. The differences between the boxplots of “ns_lf” and “ns_rob” are more sensible than those between the boxplots of “gtp35_lf” and “gtp35_rob” and between “gtp4_lf” and “gtp4_rob”.

We were surprised that the results of using the Lawformer did not necessarily outperform those of using the Chinese RoBERTa, no matter whether or not we fine-tuned them. We believe this is related to the fact that the Lawformer was built upon roberta-wwm-ext [17], and we used a larger and more recent version of RoBERTa (chinese-roberta-wwm-ext-large). After being fine-tuned with our data, the results of classifications improved significantly when we used the disputes in the TWJY in the experiments, as indicated by the boxplots “ns_ftlf”, “ns_lf”, “ns_ftrob”, and “ns_rob”.

The boxplots “ns_ftlf”, “gtp35_ftlf”, and “gtp4_ftlf” suggested two observations. The first is that the range of classification results shown in “ns_ftlf” is much better than the results we reported in [10]. The second is very interesting: although the results of using the itemized disputes generated by GPT-3.5 and GPT-4 in the classifiers were inferior to those of using the disputes that were recorded in the original judgment documents. The differences were not very bad, and the gap shrank when we changed from GPT-3.5 to GPT-4. The same observations were repeated when we replaced the

Lawformer with the RoBERTa for embedding. Hence, it may become reasonable for us to use ChatGPT to summarize the litigants’ disputes in our future work.

Figure 5 provides the boxplots for the accuracy of the 12 experiments. The qualitative observations remain the same.

7 Concluding Remarks and Further Discussions

We reported two lines of work in this paper. The first is about the classification procedure that consisted of a clustering component and a convolutional neural network component. This hybrid design outperformed our previous recommender for similar labor and employment cases [10].

Our second exploration may be more interesting for the JURISIN. We relied on the itemized disputes that the courts prepared for clustering and classification. Not all judgment documents we could download from TWJY contained such itemized disputes. The availability of such itemized disputes confines the scale of our experiments, so we tried to use the ChatGPT to summarize the claims and to itemize the disputes of the litigants. We designed our prompts to avoid possible hallucinations of the ChatGPT. We evaluated the GPT-generated disputes by using them in place of the court-prepared disputes. The results, as depicted in Figures 4 and 5, were quite encouraging. We observed that GPT-4 was better than GPT-3.5 in the final results. As we read the GPT-generated disputes in person, we also observed the superiority of GPT-4. As the technology of large language models has advanced so much recently, we look forward to the days when they become practically useful.

The world of LLMs, including ChatGPT, is changing at an extremely high speed. Since ChatGPT was launched in November 2022, GPT-4, GPT-4 Turbo (or GPT-4-1106-preview)², and GPT-4-0125-preview³ followed within 14 months. The experimental results reported in this manuscript were based on GPT-3.5 and GPT-4. We have completed some preliminary evaluations of GPT-4 Turbo and GPT-3.5 Turbo for our tasks, but we have not evaluated GPT-4-0125-preview completely yet. Using GPT-4 Turbo and GPT-3.5 Turbo offered better results than using the previous GPT-4 and GPT-3.5. That was partially due to the increases in allowed tokens in the conversations. The improvements are not surprising and are certainly welcomed, as we have discussed in Section 6. The evaluation of the LLMs is not a trivial task, especially when the LLMs evolve so quickly and so aggressively [12]. We plan to investigate the factors that led to the improvements and analyze the influences of the factors on the effectiveness and efficiency of our clustering and classification components.

Reviewers of this paper recommended us to calculate the ROUGE scores [6] of the disputes that were generated by the LLMs unanimously, using the disputes that were listed in judgment documents as the references. ROUGE scores were used for evaluating the algorithmically generated summaries in the literature. However, there are different ways to the ROUGE scores, ROUGE-L in particular, when the generated summaries and the references have different numbers of sentences (itemized disputes in our work). In addition, we have to choose a segmentation tool for Chinese text. The task of

² <https://openai.com/blog/new-models-and-developer-products-announced-at-devday>

³ <https://openai.com/blog/new-embedding-models-and-api-updates>

	macro precision			macro recall			macro F ₁		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
GPT-3.5	0.530	0.275	0.373	0.321	0.149	0.190	0.372	0.175	0.221
GPT-4	0.423	0.185	0.252	0.444	0.196	0.262	0.405	0.173	0.225

Table 2. Macro averages of the ROUGE scores for GPT-3.5-turbo and GPT-4 (0613)

evaluating the itemized disputes with ROUGE scores is not trivial, especially, given that an LLM may not generate the same disputes for repeated conversations, we should observe the distributions of the ROUGE scores.

We employed open-source tools for segmenting Chinese text⁴ and calculating the ROUGE scores.⁵ Since each judgement will have a set of ROUGE scores, so we computed the macro averages of the ROUGE scores for all of our judgment documents. At the time of writing, we have seen different versions of GPT-3.5 and GPT-4. The statistics in Table 2 are for the version of GPT of 13 June 2023.⁶ Note that there are two possible definitions of macro F₁, and we adopted the maF₁ in [14].

We have evaluated more recent GPTs, although we do not show their statistics here. Given that we observed that using GPT-4 to generate the itemized disputes for us generally led to better final results than using GPT-3.5 in Figure 4, we may tend to believe that the macro recall rates to evaluate the output of the GPTs is a better metric for our tasks.

Acknowledgments

We thank the reviewers for their valuable comments. This research was partly supported by project 110-2221-E-004-008-MY3 of the National Science and Technology Council of Taiwan. Po-Hsien Wu proposed and implemented the classifiers. Wei-Jie Lee handled the GPT-related tasks. Chao-Lin Liu led this and other cited projects, and wrote this paper. The authors discussed and cooperated on all research issues for the reported work.

References

1. Cui Y, Che W, Liu T, Qin B, Wang S, and Hu G. Revisiting pre-trained models for Chinese natural language processing. *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing*. 2020; p. 657–668.
2. Dan J, Xu L, and Wang Y. Integrating legal event and context information for Chinese similar case analysis. *Artificial Intelligence and Law*. October 2023; published online.
3. Fang J, Li X, and Liu Y. Low-resource similar case matching in the legal domain. *Lecture Notes in Computer Science 13530*. 2022; p. 570–582.
4. He T, Lian H, Qin Z, Zou Z, and Luo B. Word embedding based document similarity for the inferring of penalty. *Proc. of Int’l Conf. on Web Information Systems and Applications*. 2018; p. 240–251.
5. Kumar S, Reddy P K, Reddy V B, and Suri M. Finding similar legal judgements under

⁴ These tools include `hfl/chinese-roberta-wwm-ext-large`, `AutoTokenizer`, and `SentenceTransformersTokenTextSplitter`.

⁵ rouge-chinese: <https://pypi.org/project/rouge-chinese/>

⁶ <https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo>

- common law system, *Proc. of the Eighth Int'l Workshop on Databases in Networked Information Systems*. 2013; p. 103–116.
6. Lin C-Y. ROUGE: A package for automatic evaluation of summaries, *Text Summarization Branches Out*, 2004; p. 74–81.
 7. Liu C-L and Hsieh C-D. Exploring phrase-based classification of judicial documents for criminal charges in Chinese. *Lecture Notes in Artificial Intelligence (LNAI)* 4203. 2006; p. 681–690.
 8. Liu C-L, Lin H-R, Liu W-Z, and Yang C. Functional classification of statements of Chinese judgment documents of civil cases (alimony for the elderly), *Proc. of the Thirty-Fifth Int'l Conf. on Legal Knowledge and Information Systems*, 2022; p. 206–212.
 9. Liu C-L, Liu W-Z, Wu P-H, Huang S-c, and Huang H-C. Modeling the judgments of civil cases of support for the elderly at the district courts in Taiwan, *Proc. of the Seventeenth Int'l Workshop on Juris-Informatics*. 2023; p. 163–176. (An extended version will appear in an LNAI book)
 10. Liu C-L and Liu Y-F. Some practical analyses of the judgment documents of labor litigations for social conflicts and similar cases, CEUR Workshop Proceedings 3423: *Proc. of the Third Int'l Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workplace*, the Nineteenth Int'l Conf. on Artificial Intelligence and Law. 2023; p. 100–109.
 11. Mandal A, Ghosh K, Ghosh S, Mandal S. Unsupervised approaches for measuring textual similarity between legal court case reports. *Artificial Intelligence and Law*. 2021; 29:417–451.
 12. Ma M and Mandal J. Overcoming Turing: Rethinking evaluation in the era of large language models, Stanford Law School Blogs, Nov. 16, 2023. <https://law.stanford.edu/2023/11/16/overcoming-turing-rethinking-evaluation-in-the-era-of-large-language-models/>
 13. Reimers N and Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks, *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the Ninth Int'l Joint Conf. on Natural Language Processing*. 2019; p. 3982–3992.
 14. Takahashi K, Yamamoto K, Kuchiba A, and Koyama T. Confidence interval for micro-averaged F1 and macro-averaged F1 scores, *Applied Intelligence*, 2022; 52:4961–4972.
 15. Weiser B and Schweber N. The ChatGPT lawyer explains himself. *New York Times*. June 8, 2023. <https://www.nytimes.com/2023/06/08/nyregion/lawyer-chatgpt-sanctions.html>
 16. Weiss D C. Latest version of ChatGPT aces bar exam with score nearing 90th percentile. *ABA Journal*. March 16, 2023. <https://www.abajournal.com/web/article/latest-version-of-chatgpt-aces-the-bar-exam-with-score-in-90th-percentile>
 17. Xiao C, Hu X, Liu Z, Tu C, and Sun M. Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open*. 2021; 2:79–84.
 18. Xiao C, Zhong H, Guo Z, Tu C, Liu Z, Sun M, Feng Y, Han X, Hu Z, Wang H, and Xu J. CAIL2018: A large-scale legal dataset for judgment prediction. 2018; arXiv:1807.02478.
 19. Zhang G, Lillis D, and Nulty P. Can domain pre-training help interdisciplinary researchers from data annotation poverty? A case study of legal argument mining with BERT-based transformers. *Proc. of the Workshop on Natural Language Processing for Digital Humanities*. 2021; p. 121–130.

Appendix A

This appendix offers Chinese examples for what we discussed in Section 3.3. We show two dispute lists in Chinese. This appendix was copied from [10].

Example 1

The source is CHDV,92,勞訴,32,20040102,1.json.

1. 系爭夜點費應否列入平均工資核發退休金？
2. 原告己○○、丙○○、乙○○、戊○○、辛○○、庚○○於具領退休金時，所簽立之收據之效力為何？

Example 2

The source is CHDV,98,勞訴,37,20100409,1.json.

1. (一)長森醫院於 97 年 7 月 31 日是否有歇業之事實？
2. (二)兩造間是否因歇業而終止勞動契約？

After the blurring step (cf. Section 3.3), the anonymized person names in the second dispute in Example 1 were changed to “someone” in Chinese, which is listed below.

原告某人、某人、某人、某人、某人於具領退休金時，所簽立之收據之效力為何？

The place name and the time expression in the first dispute in Example 2 were changed to “somewhere” and “sometime” in Chinese as well.

(一)某地於某時是否有歇業之事實？

Appendix B

This appendix provides the Python code mentioned in Section 4. We translated the Chinese directives to English in the following illustration. The values of the two variables, i.e., `p_point` and `d_point`, would be replaced by the actual statements of the plaintiffs and the defendants for individual cases, respectively.

`#step1(summarizing the claim of plaintiff)`

```
prompt = f'''List the key points of the following article
using {'p_point': []} as the template\n :{plaintiff_claim}'''
```

`#step2(summarizing the claim of defendant)`

```
prompt = f'''List the key points of the following article
using {'d_point': []} as the template\n :{defendant_claim}'''
```

`#step3(dispute extracting)`

```
prompt = f'''The following are the main points argued by
the claimant and the counterparty. Based on this, list
the dispute points between the two parties, and use
{'dispute': []} as the template\n The following is the
plaintiff's claim:\n {p_point} \n The following is the
defendant's claim:\n {d_point} '''
```

Towards General Requirements for Norm Representation Languages Using Competency Questions: The Case of Comparing Flint and ODRL[★]

Jeroen Breteler¹[0000–0002–8919–1231] and
Thom van Gessel¹[0000–0002–4686–0490]

¹TNO, The Hague, The Netherlands

Abstract. The field of normative representation needs a way of evaluating whether representation languages are fit-for-purpose, and needs a way of understanding how representation languages can be compared or related to one another. We propose to use the methodology of competency questions, where model requirements are phrased as questions that should be answerable through the model’s data structure, as a basis for meta-level discussions about norm representation language design. To work towards this goal, we present an initial set of 52 competency questions intended for requirements testing of any norm representation language. We exemplify the usage of the competency questions by evaluating and comparing ODRL and Flint. We note that ODRL restricts itself to state-level information, while Flint additionally expresses state transitions; we identify competency questions that pinpoint these differences. We suggest that ODRL is primarily suitable for the purpose of policy enforcement, while Flint is better suited to the purpose of normative coordination between stakeholders. Thus, we demonstrate that subsets of the competency question set relate to various types of use cases, and we argue for a use case typology based on competency questions. We conclude that a widespread agreement on requirements, pursued through formulating competency questions, allows the study of normative representation languages to focus on its ultimate purpose: To understand and serve the variety of needs of all people who engage with norms.

Keywords: Norms · Norm representation · ODRL · Flint · Competency questions · Normative coordination

[★] We are grateful to Giovanni Sileno, Tom van Engers, and Thomas van Binsbergen for comments on early outlines of this paper. We thank Robert van Doesburg for validating the competency question set for Flint. We thank Cornelis Bouter for general discussions about the topic of competency questions. We gratefully acknowledge the detailed and constructive comments from three anonymous reviewers of the JURISIN 2024 workshop.

1 Introduction

In 2009, Gordon et al. [7] noted the “extensive effort for modeling many aspects of legal rules and regulations” that was underway in the AI & Law community. As they surveyed the field, they also suggested many initiatives were perhaps held back by “[the lack of] an overall and systematic effort to establish a general list of requirements for rule interchange languages”, as well as that “there is not yet an agreement in particular among the practitioners working in this field.” Gordon et al. go on to suggest a number of aspects that might be relevant for determining the requirements on normative (especially legal) representations; among others they mention isomorphism between norm and source, metadata about jurisdiction, authority, and temporal properties, and the values that norms might be intended to promote.¹ Relatedly, Guido Boella and colleagues wrote on requirements on representations and applications in the domain of legal compliance checking. For instance, in [4, 3] they suggest that “[n]orms and the interpretation of norms need to have a status, active or inactive, and to be linked to explanations and sources for clarification as needed” ([3], p.18). More recently, Leone and colleagues [11] in their survey of legal knowledge ontologies note that “criteria used to evaluate the proposed models” are conspicuously absent for most models.²

In this paper, we follow the call of previous literature to work towards a typology of use cases and a common view on the requirements made on representation languages supporting a given use case type. We propose to structure the discussion through the practice of defining competency questions, which is a method for formulating requirements used in ontology design. Competency questions “represent functional requirements, in the sense that the developed ontology [...] should be able to answer them; hence the ontology [...] can be considered as containing all relevant knowledge” ([22], p. 1). Full-fledged ontologies can see tens or hundreds of competency questions testing various aspects of the model [15]. Consequently, we believe competency question methodology allows for a concrete and fine-grained way of specifying requirements, thus enabling a fruitful discussion about which requirements apply to which use case types for norm representation languages. This can help researchers understand each other’s work better, and identify where there are gaps either in our understanding or in the solutions we are offering to the public.

¹ We thank a reviewer for pointing out to us that many of Gordon et al.’s suggestions on types of requirements featured in the design process for LegalRuleML [2]. For our present contribution, we believe that Gordon et al.’s observations continue to be relevant as a motivation for finding agreement on fine-grained, testable requirements on norm representations and the practical motivations for (sets of) those requirements.

² Leone et al. note several examples of models that do specify their evaluation criteria, particularly with the use of competency questions. While we welcome such efforts, we note that such competency questions will be tailored to the specific goals of the ontology’s design context. In this paper, we argue for the additional step of using competency questions to understand the general picture of norm representation needs and solutions.

The above considerations surfaced in our experiences while developing the Flint language for representing normative systems [20, 5]. We needed a means of requirements testing firstly for our internal purposes. Defining competency questions also allowed us to make sure our team was in agreement on the development direction we were taking. We additionally realized we needed a way to compare our solution to other solutions in the field. We wanted a structured, comprehensive way of understanding the differences between our solutions and others. In this paper, we will take an example case to study differences between Flint and the popular Open Digital Rights Language (ODRL, [10]). ODRL is a standardized language used to express permissions, prohibitions and obligations associated with digital content and services. It allows for communication of how digital content can be accessed, used, and distributed. Flint is a language that is used to express interpretations of normative sources, while also serving as a basis for technical implementation of compliance checking. It does so by characterizing norms in terms of normative acts, and detailing the preconditions for an act’s validity and the postconditions resulting from an act’s execution.

The next section discusses our competency question set, and their relation to use case types. Section 3 then sets up a small example case to demonstrate Flint and ODRL, and evaluates the differences between the two languages in terms of competency questions. Finally, in section 4 we raise various points for discussion and future work.

2 Competency Questions Set

In order to support requirements testing for a wide range of use case types, the ideal set of competency questions should consider many aspects of normative sources and their interpretations. In addition, the questions should be formulated in a maximally neutral, pre-theoretical way, so that the discussions can be unhindered by the particular theoretical point of departure that researchers might happen to take, such as deontic logic or a Hohfeldian perspective [9, 21].³ In this paper, we present an early step towards that ideal, as our competency questions so far grew mainly out of our own practical context and ontological commitment. We do believe that the competency question set we will present below has one essential property: it can be constructively disagreed with. Thus, we hope that any reader who sees an incompleteness or other fault in this proposal will also see a path to improving on it. In this sense, disagreement with specific choices we made here suggests an agreement with our overall thesis that the competency question methodology facilitates a concrete meta-level discussion with a fine-grained structure.

³ We thank a reviewer for the suggestion to point out that both of these schools of thought, as well as the connections between them, are subject to continuing debate and have benefited from further formalization efforts since their inception; see e.g. [6, 12].

2.1 Pre-theoretic questions

Our initial interest with the Flint language was to represent how actors might navigate what we’ll loosely term “normative worlds” (or, the states of a normative system [1]). Thus, our competency questions relate to broad questions such as the following:

- What possible normative worlds might an agent be in?
- What means are available for an agent to influence their normative world?
- What are the consequences of an agent using a particular means of influence?
- How might the agent be affected by the world (especially other agents)?
- What expectations might agents have about others’ behavior?

Flint is aimed at supporting multiple stakeholders to come to a shared interpretation of a set of sources [19, 20, 5]. Thus, the questions above relate to Flint’s aim of clarifying the normative positions that different types of agents can hold in different states of the world. As we will discuss below, different purposes will require additional competency questions in order to capture all requirements.

The pre-theoretic questions above underspecify, by design, the form that the answer can take. Answering the question reveals an ontology’s particular ontological commitment. For example, in developing Flint we have used the foundation of Hohfeldian relations theory [17, 9]. Thus, for Flint, a “means of influencing the normative world” would be when an agent holds a power in a power–liability relation. This is in line with [22], who caution against overfitting the competency questions to the ontology and instead advocate for competency questions phrased in “natural language”. However, we suggest that this need for generality can conflict with a need for specificity. That is, we might have a need to pose competency questions at a level of specificity that requires us to make some ontological commitments. For Flint, our current attempts balance between these needs; while our competency questions promote generality by staying away from Flint-specific concepts such as a “normative act”, we have so far chosen to specify the questions within the context of Hohfeldian relations theory.

We refer readers interested in applying our CQs for other ontological commitments to literature on fundamental aspects of norms. In particular, we suggest that our CQs can be “translated” to a deontic logic perspective by applying the insights from Giovanni Sartor [18]. Briefly: Leaning on the insight that both the Hohfeldian and deontic logic perspective use a notion of action, Sartor suggests that power–liability relations (and generalizations about them) might be construed as permissions for the power holder to take action specifically against the liable party. A similar mapping is possible between duty–claim relationships and the notion of an obligation.

2.2 Specific questions

Table 1 shows an overview of the topics included so far in our competency questions set, as well as the counts of questions relating to each topic. We also

distinguish between topics that were in scope for the development of Flint, on the left side of the table, and those that were not, on the right side. We return to the notion of question subsets and relations to use case types further below.

In the following, we give examples of competency questions for all in-scope topics.⁴ The out-of-scope topics are, however, essential for a full discussion about requirements on norm representations. Moreover, we expect that a full competency set will cover an even broader range of topics; as discussed in section 1, we can draw inspiration from previous work here [7, 4]. One note that applies to all competency questions, with the exception of the final topic on instances, is that Flint is meant for describing *generalized* norms. That is, Flint is meant for agreement about interpretations in the general case, by making statements about *kinds of* powers, agents, etc. To apply an interpretation in Flint to a particular case, additional statements are needed that identify the atoms making up the case, as well as the instantiation of the generalized statements through an assignment of those atoms to the general categories [13].

Topic (in Flint scope)	Count	Topic (out of Flint scope)	Count
Power–liability relations	4	Non-agentive state changes (time, natural events, ...)	2
Agents & actions	4	Derivable normative relations	3
Conditions	8	Omitted actions	4
Duty–claim relations	6	Duty violations	2
Duties and agents	6	Functions on facts (arithmetic)	3
Facts & boolean structure	7	Scenarios & instances	3
Total	35	Total	17

Table 1. Topics in the competency questions set.

Power–liability relations As stated above, powers in power–liability relations are the Hohfeldian implementation of a “means of influence” that an agent might hold against other agents in their normative world. An example of a question about such relations is shown below; as stated we phrase the question using “kind of” to indicate its generalized nature. Where before we used the pre-theoretic term “normative world”, here we use “normative system”. We also give each competency question a name for easier referencing. Finally, we use “this” to imply that the question is being asked in a certain context where we expect the information to be available.

cq-power–liability: What kinds of power–liability relations can exist in this normative system?

⁴ The full set of questions is available at <https://gitlab.com/normativesystems/knowledge-modeling/flint-ontology/-/blob/main/competency-questions.md>. The latest commit SHA at time of writing was `dbf79446ef274eb5b8f9e25913d8410bbbd7df8b`.

Agents & actions Power norms are typically described for certain classes of agents, actions, etc. We formulate questions to identify those aspects, such as the one below. Here, we try to phrase the question as neutrally as possible on the matter of how facts about the normative state are structured, so we allow either bare facts or expressions over facts.

cq-act-actor: Given a kind of power, what kinds of facts or logical expressions over facts must hold about an agent in order to be a valid holder of this power?

Conditions Norms can apply conditionally on some aspects of the state. In addition, norms can contain information about how normative states can be changed. We refer to these considerations as preconditions and postconditions. For Flint's purpose of normative coordination among stakeholders, both types of conditions are relevant, so we use questions such as the following ones to identify these matters:

cq-act-precondition: Given a kind of power, what kinds of facts must hold about the circumstances in order for the exercise of this power to be valid?

cq-act-create: Given a kind of power, what kinds of facts will continue to hold or be brought about after its exercise?

Duty–claim relations Similarly to the power–liability relation, we wish to identify possible duty–claim relations that might exist in normative states. In addition to basic competency questions as exemplified above, we also ask questions about the ways that duty–claim relations might come about or cease to apply, as exemplified below:

cq-duty-creation: Given a kind of duty, what kinds of power's exercise could bring about such a duty?

Duties and agents The considerations above can also be applied at the agent level; we want to be able to identify how agents involved in a power–liability relation might be involved in a duty–claim relationship connected to it, which is possible with questions such as the following:

cq-power-duty-actor-holder: Given a kind of power whose exercise creates a kind of duty, does the agent exercising the power become the duty holder?

Facts & boolean structure We are also interested in all the different forms that a given normative world might take. Consequently, we ask questions such as the following:

cq-fact: What are all aspects on which two states in this normative system can differ?

2.3 Use case types

As we described above, there is a motivation for each topic for which we define competency questions. Consequently, we suggest to study the relationship between sets of competency questions and types of use cases. We hope this will allow for a typology of use cases, each with its own set of requirements, to which norm representation languages can then be tailored. In Table 2, we suggest some types of use cases, each having a superset of requirements compared to the previous entry.

As we will see in the next section, for use cases about policy enforcement, where typically a digital implementation of a “gatekeeper” can determine matters such as right of access to or usage of some digital property, it might be sufficient to model the normative relations (or their deontic counterparts), the conditions on exercising powers, and some information about the state of the normative world.

The addition of requirements on postconditions makes it possible to support normative coordination use cases, such as drafting legislation, finding agreement between stakeholders, or communicating about norms to stakeholders of various technical, legal, or non-expert backgrounds. It is because of the postconditions, and their coupling to the preconditions, that one can express potential changes to the current state; it is this that allows one to express the consequences of kinds of actions for future kinds of conduct. Normative coordination cases further benefit from requirements on the provenance of norms, i.e. by formulating competency questions about the relations between sources of norms and their interpretation.

Further use case types can be supported by moving from the generalized view on norms to include the instance level. That is, by adding requirements about concrete scenarios and the assignment of atoms in these scenarios to instantiate generalized norms, one can determine how behaviors by agents in the scenarios lead to certain concrete state transitions. Lastly, by coupling the normative model to a domain ontology, it becomes possible to apply the norms to real-world cases; this enables e.g. normative advisors [14] and digitally supported decision-making.

Not all use case types will fit in a hierarchy. For some use cases, it will be necessary to model certain kinds of general knowledge. For example, use cases relating to tax law might require support for arithmetical functions or operators. Arguably, other use cases might benefit from a modeling approach that combines normative information with some non-normative information, such as natural events.

3 Example: Library Regulations

We illustrate the competency of Flint and ODRL with a simple example: A set of rules for borrowing books at a library. Let us assume that our library has formulated its regulations in the following way:

1. Library members can borrow books

Use case type	Competency questions	Examples
Core	Powers, power components, Duties, duty components	
Policy enforcement	+Preconditions, +Facts & booleans	Digital rights management, access control*,
Normative coordination	+Postconditions +Source	Policy making, Policy explaining, interdisciplinary communication
Normative reasoning	+Scenarios +Instances	Policy simulation
Norm implementation	+Domain ontology	(Digital) normative advisors, digital decision-making

Table 2. A hierarchy of use case types and their relation to competency questions.

2. If a library member borrows a book, the book should be returned within 28 days.
3. The membership of a library member who did not return a borrowed book in time, is suspended.

3.1 Flint

Flint interpretations revolve around acts: Flint’s way to represent power–liability relations. The interpretation of the library regulations consists of three of these acts: borrowing a book, returning a book and suspending a membership (see Figure 1).

Using the properties *hasActor* and *hasRecipient*, it is indicated what the requirements are to be an actor or an interested party of these acts, respectively: only members of the library can borrow books, and only librarians can suspend a membership. The *hasObject* property is used to indicate that only library books can be borrowed. Some acts also have other preconditions. For instance, when suspending a membership, the duty to return a book should still be open.

The properties *creates* and *terminates* indicate how acts affect the normative state, namely by creating or terminating a fact. The duty to return a book, for example, is not assumed to hold in all states, but only in states that are the result of a book being borrowed. The duty can also be taken away by returning the book.

The red hexagons indicate how the roles in an act carry over to a fact. In this way, the duty that is created by borrowing a book can be created for a particular individual library member with reference to a particular book.

3.2 ODRL

We now move on to defining the same set of library rules in ODRL. Unlike actions in Flint models, *rules* take center stage in ODRL models. In the model for

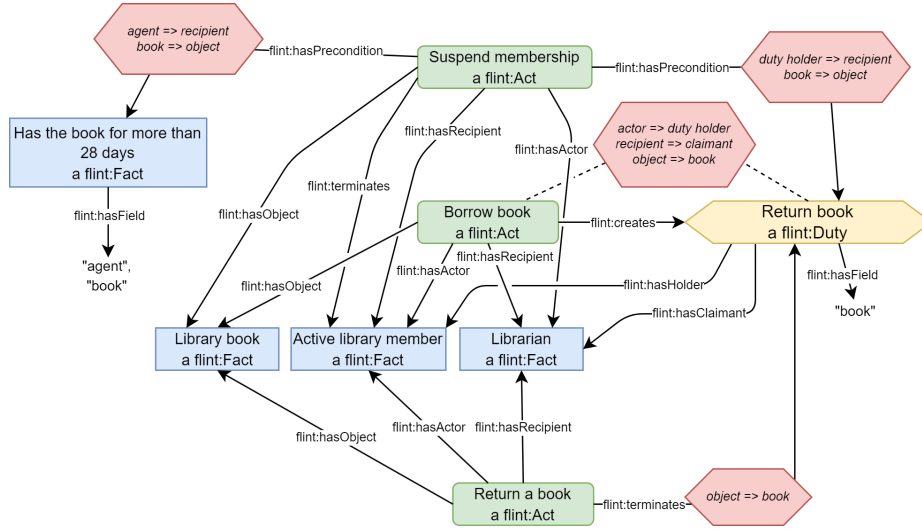


Fig. 1. Library book borrowing rules in Flint

borrowing a book (Figure 2), this rule takes the form of a permission, namely the permission to perform the action of borrowing. The assignee and assigner properties indicate the interested parties: Any individual from the collection of library members can be an assignee, given that their membership has not been suspended. This is indicated using a constraint on the party collection. Constraints could also be applied to assets or actions. The assigner of the permission is the library itself. The target of the permission is the collection of books in the library.

Returning a book is defined as a duty in ODRL. Unlike the permission to borrow books, a duty to return a book does not apply to the whole collection of library members and books, but to a specific member and the specific book they borrowed. Because of this, the model has to be constructed with reference to these individuals. ODRL does not provide a way to indicate that performing the action of borrowing a book results in a duty for that specific member about that specific book. It does allow for direct consequences of not fulfilling a duty, namely by creating a new duty, in this case the duty to suspend the member. However, this renders it *obligatory* for the library to suspend members with overdue books, arguably against the intuition that the library is permitted but under no obligation to do so. Moreover, ODRL does not provide a way to indicate that the library member to be suspended is the person who did not fulfill their duty to return the book.

3.3 Differences between Flint and ODRL

The example above illustrates the differences between Flint and ODRL with respect to which competency questions can be answered based on the data. The

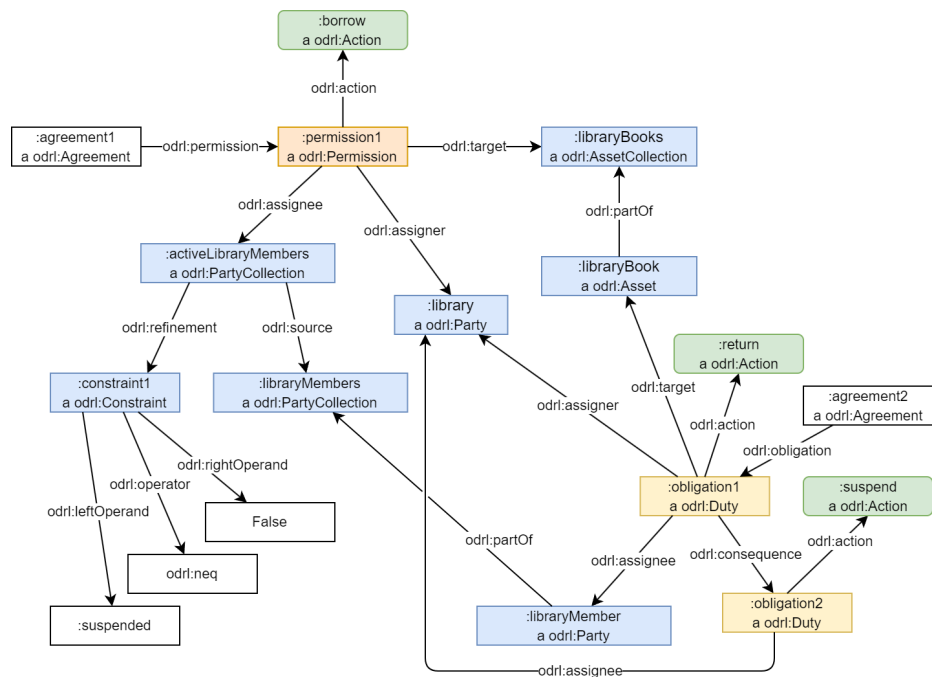


Fig. 2. Library book borrowing rules in ODRL

Flint interpretation allows us to derive all possible normative states and shows how these states can be reached. For example, a new library member could verify in advance that they will be obliged to return a book when they borrow one, even if they haven't done so yet.

ODRL does allow us to determine what normative state we are in; given what we know about reality, we can determine whether any constraints for permissions, obligations and prohibitions are met, and therefore which rules currently apply. But it does not allow us to determine what counterfactual normative states might arise later. In practice, this means that an ODRL representation of the library rules cannot tell a library member what will happen when they borrow a book. Another way to say this is that all competency questions about state change are out of scope. Consider the question **cq-act-create**, repeated below:

cq-act-create: Given a kind of power, what kinds of facts will continue to hold or be brought about after its exercise?

The interpretation in Flint can answer this question in the following way: Taking the act of borrowing a book, we can ask what facts are related to it by the property *creates*, and we find the duty to return the book (*Duty* is a subclass of *Fact* in Flint). In the ODRL model, no such relation obtains between the permission to borrow a book (or the action itself) and the duty to return it.⁵

Crucially, the differences pointed out here can help to clarify the suitability of ODRL and Flint for various purposes. As alluded to in section 2.3, ODRL's focus on individual states could be perfectly adequate for many policy enforcement use cases, where interactions might typically involve a user giving a one-time proof of the relevant facts to gain access or usage rights to an asset.

Another nuanced distinction between Flint and ODRL is the one between assigner and recipient. In many cases, those roles are filled by the same agent, but not always: It is possible in ODRL to assign a permission to perform an action against some third party that is not the assigner themselves. This shows that while Flint emphasizes who is affected by actions through the *recipient* role, ODRL emphasizes who assigns actions. The competency questions set might therefore be expanded by including a question about action assignment, which would highlight that Flint has no built-in support for use case types where action assignment plays a role.

4 Discussion

Firstly, we reiterate that the competency question set presented here is a starting point, influenced by the context of our own work, and therefore by no means complete or final. One way to improve the question set would be to compare and

⁵ This difference can also be stated in terms of translations of competency questions into queries, for instance in SPARQL. In Flint, **cq-act-create** can be translated as `SELECT ?fact WHERE $this flint:creates ?fact`, while no such translation is available for ODRL.

combine it with competency questions formulated for other norm representation languages, as done for example for UFO-L [8].

Secondly, we note that to realize the vision of wide-spread agreement on general requirements will involve also an operational effort. We hope our work will encourage other development efforts to publish their competency questions sets, and perhaps to clarify where their choices differ from ours. Moreover, future work could investigate what infrastructure would facilitate sharing, discussing, and iterating on a set of competency questions. Such infrastructure could also involve automated testing and other tooling.

Lastly, we wish to dispel the notion that competency questions are the solution to all challenges. Competency questions do not help to test for non-functional requirements, such as considerations of computational efficiency (e.g. [16]) or learnability of the language.

5 Conclusion

In this paper, we have argued that the study of norm representation languages is best conducted by understanding the variety of use case types which the representation languages are intended to serve, and the expressiveness needed in each such context. We have argued that formulating competency questions is a helpful method for doing so, because it offers a concrete and fine-grained basis for discussion about what are the relevant requirements for a use case type.

Our example case on library book borrowing rules has highlighted the relation between competency questions and use case types. Unlike Flint, ODRL does not support answering questions about postconditions, i.e. about the relation between who borrows a book and who pays a fine. Rather than interpret this as a defect or extravagance on the side of either language, we can understand these findings as relating to the different contexts where the two languages can add value, namely policy enforcement and normative coordination. It is our hope that future work can expand on this typology of use cases and their associated requirements, for an ideal positioning of all the solutions that the field of norm representation has to offer to the public.

References

1. Alchourrón, C.E., Bulygin, E.: Normative systems. Springer-Verlag (1971)
2. Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: Oasis legalruleml. In: ICAIL '13: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 3–12 (2013)
3. Boella, G., Humphreys, L., Muthuri, R., Rossi, P., van der Torre, L.: A critical analysis of legal requirements engineering from the perspective of legal practice. In: 2014 IEEE 7th International Workshop on Requirements Engineering and Law (RELAW). pp. 14–21. IEEE (2014)
4. Boella, G., Janssen, M., Hulstijn, J., Humphreys, L., Van Der Torre, L.: Managing legal interpretation in regulatory compliance. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 23–32 (2013)

5. Breteler, J., van Gessel, T., Biagioni, G., van Doesburg, R.: The FLINT ontology: An actor-based model of legal relations. In: Knowledge Graphs: Semantics, Machine Learning, and Languages, pp. 227–234. IOS Press (2023)
6. Gabbay, D., Horty, J., Parent, X., Van der Meyden, R., Van der Torre, L. (eds.): Handbook of Deontic Logic and Normative Systems. College Publications (2013)
7. Gordon, T.F., Governatori, G., Rotolo, A.: Rules and norms: Requirements for rule interchange languages in the legal domain. In: International Workshop on Rules and Rule Markup Languages for the Semantic Web. pp. 282–296. Springer (2009)
8. Griffo, C., Sales, T.P., Guizzardi, G., Almeida, J.P.A.: Legal power-subjection relations: Ontological analysis and modeling pattern. In: Conceptual Modeling: 41st International Conference, ER 2022, Hyderabad, India, October 17–20, 2022, Proceedings. pp. 65–81. Springer (2022)
9. Hohfeld, W.N.: Some fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal* **23**(1), 16 (1913)
10. Iannella, R., Villata, S.: Open Digital Rights Language Information Model 2.2 (2018), <https://www.w3.org/TR/odrl-model/>
11. Leone, V., Di Caro, L., Villata, S.: Taking stock of legal ontologies: a feature-based comparative analysis. *Artificial Intelligence and Law* **28**(2), 207–235 (2020)
12. Markovich, R.: Understanding hohfeld and formalizing legal rights: the hohfeldian conceptions and their conditional consequences. *Studia Logica* **108**(1), 129–158 (2020). <https://doi.org/https://doi.org/10.1007/s11225-019-09870-5>
13. Minsky, M.: A framework for representing knowledge (1974)
14. Parizi, M.M., van Binsbergen, L.T., Sileno, G., van Engers, T.: A modular architecture for integrating normative advisors in mas. In: European Conference on Multi-Agent Systems. pp. 312–329. Springer (2022)
15. Potoniec, J., Wiśniewski, D., Ławrynowicz, A., Keet, C.M.: Dataset of ontology competency questions to sparql-owl queries translations. *Data in Brief* **29**, 105098 (2020). <https://doi.org/https://doi.org/10.1016/j.dib.2019.105098>, <https://www.sciencedirect.com/science/article/pii/S2352340919314544>
16. Robaldo, L., Batsakis, S., Calegari, R., Calimeri, F., Fujita, M., Governatori, G., Morelli, M.C., Pacenza, F., Pisano, G., Satoh, K., Tachmazidis, I., Zangari, J.: Compliance checking on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies. *Artificial Intelligence and Law* pp. 1–51 (2023)
17. Salmond, J.W.: *Jurisprudence: Or, The Theory of the Law*. Stevens and Haynes (1907)
18. Sartor, G.: Fundamental legal concepts: A formal and teleological characterisation. *Artificial Intelligence and Law* **14**, 101–142 (2006)
19. Van Doesburg, R., Van Der Storm, T., Van Engers, T.: *Calculemus: Towards a formal language for the interpretation of normative systems*. AI4J workshop at ECAI 2016, The Hague, Netherlands (2016)
20. Van Doesburg, R., Van Engers, T.: Explicit interpretation of the dutch aliens act. In: *Proceedings of the Workshop on Artificial Intelligence and the Administrative State (AIAS 2019)*. pp. 27–37 (2019)
21. Von Wright, G.H.: Deontic logic. *Mind* **60**(237), 1–15 (1951)
22. Wiśniewski, D., Potoniec, J., Ławrynowicz, A., Keet, C.M.: Analysis of ontology competency questions and their formalizations in sparql-owl. *Journal of Web Semantics* **59**, 100534 (2019). <https://doi.org/https://doi.org/10.1016/j.websem.2019.100534>, <https://www.sciencedirect.com/science/article/pii/S1570826819300617>

Development of a Multilingual Model for Mapping Japanese and Foreign Laws

Daichi YAMADA¹ and Makoto NAKAMURA¹

Niigata Institute of Technology,
Fujihashi 1719, Kashiwazaki City, Niigata Pref., 945-1184, Japan
`mnakamur@niit.ac.jp`

Abstract. An automated system for mapping similar provisions between Japanese and foreign laws can significantly benefit comparative legal studies. Previous research has been done using BERT to map legal provisions, but it relied on translating everything into English and using a monolingual BERT. In this study, we utilize a multilingual model to map foreign laws directly using the original legal texts. From our experiments, we confirmed that it is possible to replicate traditional methods using a multilingual model. Additionally, when mapping between Japanese laws, performance improved through fine-tuning with a legal corpus. Notably, supervised learning demonstrated superior performance.

Keywords: comparative law · mapping similar provisions · BERT

1 Introduction

In conducting business overseas or when actually going abroad, it is necessary to fully understand the laws of the destination country. Focusing on the similarities between one's own country's laws and those of the destination country can make it easier to understand what is common and what differs. Comparative law is the discipline that compares laws belonging to different legal systems to clarify their differences (see Figure 1). Comparative law requires expertise in the language and legal knowledge of the region of interest. However, depending on the laws being compared, the combinations can be extensive, and considering the social background of the laws makes it a very labor-intensive task even for experts.

On the other hand, the digitization of legal data in Extensible Markup Language (XML) format is progressing, and the laws of some countries can now be handled collectively using the same format. Therefore, this study considers the use of publicly available legal XML data to computationally map similar provisions across multiple laws. If the mapping of similar provisions could be automated through computation, it would be expected to facilitate comparative law research and be useful for general business expansion and visits abroad.

Research on the automatic mapping of provisions between laws has already been conducted by researchers such as Cho [3,2] and Koseki [5], and various computational methods have been tried. One such method proposed the use of the language model BERT [4], but because it used a pre-trained model, its

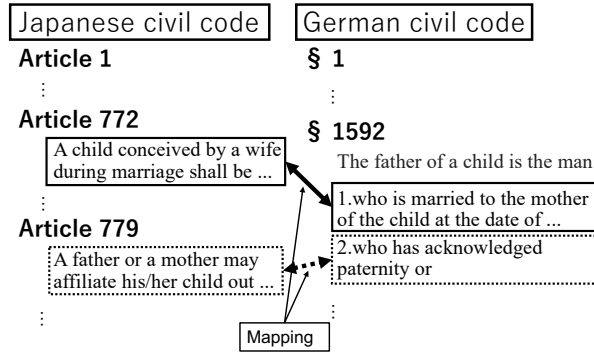


Fig. 1. Example of mapping between Japanese and Foreign Laws

performance was inferior to word-based methods like the Jaccard coefficient and BM25 when mapping between Japanese laws. General BERT models, which are trained on common texts like Wikipedia, have been reported to perform poorly in specific domains like medicine [6].

Thus, this study aims to fine-tune the BERT model with a legal corpus. It seeks to verify the effectiveness of such fine-tuning in mapping similar legal provisions. Additionally, as a trial, multilingual models are introduced to investigate whether they can map provisions in their original text.

In this study, considering the impact of translation accuracy when using monolingual models, we first performed mapping between Japanese laws that have some degree of correlation. Here, unsupervised learning using Transformer-based Sequential Denoising Auto-Encoder (TSDAE) [13] and supervised learning using labeled data were conducted for fine-tuning, and these were compared. Then, mapping was performed with foreign laws. Whereas previous research unified the language to English for mapping, this study used multilingual models to map the original legal texts. The results showed that, like traditional BERT, several provisions could be mapped. However, at the current stage, there remain various issues with creating correct data and mapping methods, so the results are provisional.

This paper is organized as follows: Section 2 introduces related research. Section 3 explains the method for mapping similar provisions between laws. Section 4 describes the details of the experiments. Section 5 presents and analyzes the experimental results. Section 6 concludes the paper.

2 Related Work

2.1 Mapping Similar Provisions

The mapping of similar provisions can be seen as a kind of similar document search, where each provision is treated as a single document. This has already

been researched [3]. Initially, the similarity is calculated for every combination of provisions from different laws. Then, for a given provision, the most similar provision is referenced, and if that provision's most similar one turns out to be the original provision, the two are mapped together.

Various methods have also been tried for calculating the similarity between documents used in mapping. Initially, documents were treated as sets of words, with the Jaccard and Dice coefficients being used to calculate similarity [3]. However, these methods had drawbacks, such as not considering the importance of words and treating synonyms as entirely different words. Therefore, mapping using cosine similarity of embedding representations by BERT was performed [2]. Additionally, the BM25, which ranks documents based on the importance of words, and Doc2Vec, which does not have input length limitations, have also been experimented with [5].

2.2 Sentence-BERT

Sentence-BERT (SBERT) is a model proposed by Reimers and Gurevych [10] and fine-tuned using a Siamese Network on a pre-trained BERT model. Compared to traditional BERT, it improves the embedded representation of sentences and is more effective when dealing with multiple documents. Among the objective functions used in its training, they adopt the Regression Objective Function. Initially, the model receives two sentences and calculates the cosine similarity of their sentence embeddings. Then, it calculates the mean squared error (MSE) with the correct label (similarity) and adjusts the embeddings so that documents that are similar have a higher similarity score. The mean squared error is defined by the following formula. In this context, '*label*' is the correct label, and $\mathbf{u}$ and $\mathbf{v}$ are the sentence vectors for Document *A* and Document *B*, respectively.

$$MSE = \|\text{label} - \cos(\mathbf{u}, \mathbf{v})\| \quad (1)$$

2.3 TSDAE

TSDAE is a Transformer model for learning unsupervised sentence embeddings [13]. TSDAE operates by adding noise to corrupt the input sentence, converting this corrupted sentence into a fixed-length vector, and then reconstructing the original input sentence from this vector to learn sentence representations. Wang et al. [13] adopted token deletion as the noise process and showed that an optimal deletion rate for input tokens is 0.6.

TSDAE differs from existing Transformer models [12] in that the key and value in Cross-Attention are limited to sentence vectors. The Cross-Attention is calculated using the following formula:

$$H^{(k)} = \text{Attention}\left(H^{(k-1)}, [s^T], [s^T]\right) \quad (2)$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3)$$

Here, $H^{(k)} \in \mathbb{R}^{t \times d}$ is the hidden state of the decoder at the k -th layer for the t -th decoding step, where d is the size of the sentence vector, $[s^T] \in \mathbb{R}^{1 \times d}$ is a one-row matrix containing the sentence vector, and Q , K , V respectively represent the query, key, and value.

2.4 Multilingual Models

The multilingual model employed in this study leverages knowledge distillation to facilitate understanding across different languages. Reimers et al. [11] believed that sentences with the same meaning in different languages should be mapped to the same position in the vector space, and they achieved the multilingualization of monolingual models through knowledge distillation (see Figure 2). The student model is trained to mimic the teacher model. Furthermore, to enable the student model to function in other languages, training is conducted with parallel sentences. In Figure 2, ‘Hello World’ and its German translation ‘Hallo Welt’ are mapped to the vector of (‘Hello World’) from the teacher model. This is achieved by training the student model using the MSE Loss.

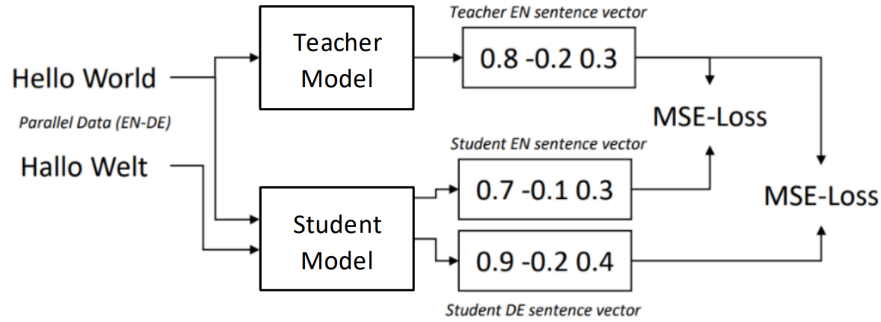


Fig. 2. Training procedure (cited from [11])

3 Proposed Method

Following previous research [2], we perform the mapping of similar legal provisions. Initially, the similarity is calculated for every combination of legal provisions. Then, the provision with the highest similarity to a given provision is referenced, and if the provision that appeared to be the most similar from its perspective matches the original provision, the two provisions are mapped together. Figure 3 shows an example of mapping provisions between Law A and Law B on a per-provision basis. The BERT model used in prior research was

pre-trained on the Japanese Wikipedia corpus as of September 2019. By fine-tuning the BERT model using legal text, an improvement in the performance of mapping is expected. Therefore, in this study, before mapping the provisions, we conducted fine-tuning of the BERT model using a legal corpus in two ways: unsupervised learning and supervised learning.

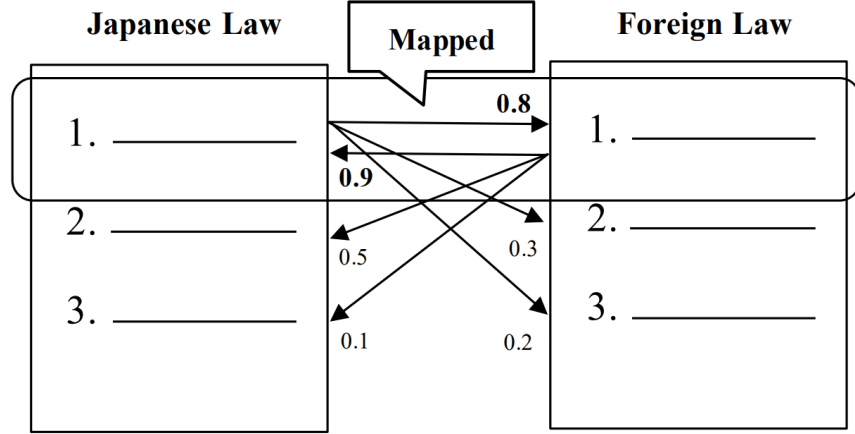


Fig. 3. Mapping similar articles

4 Experiments

4.1 Purpose of the Experiments

In this section, we map similar provisions between the Japanese Civil Code (Family Law) and the German Civil Code. The aim of this experiment is to verify whether fine-tuning the BERT model and using multilingual models are effective for mapping provisions between Japanese and foreign laws.

The experiment in this study is conducted in the following steps:

1. Unsupervised Approach (JP - JP)
2. Supervised Approach (JP - JP)
3. Using Multilingual Models (JP - DE)

Due to the difficulty in creating labeled data with legal texts and multiple languages, we focus on testing the mapping of provisions between Japanese laws to evaluate the effectiveness of fine-tuning. Here, we consider laws with similar content, such as the Electric Business Act (Law No. 170 of 1964) and the Gas

Business Act (Law No. 51 of 1954). Fine-tuning is performed in two ways: unsupervised learning and supervised learning, and the results are compared with the model before fine-tuning.

Next, we use multilingual models to map foreign laws. Here, we map the Japanese Civil Code to the German Civil Code. Previous research [2] unified the language to English and conducted mapping using an English pre-trained BERT model. In this experiment, we use a multilingual model to map each law in its original text and compare the results with the findings of the previous research [2].

4.2 Obtaining Legal Texts

In all experiments, the Japanese laws were obtained in XML format from e-Gov Legal Document Search¹, and the German laws were obtained from Gesetze im Internet². For the Japanese Civil Code, the text under the Article tag was treated as a single provision. The format of the German Civil Code’s XML differs from that of the Japanese Civil Code, so the text under the Content tag was treated as a single provision.

4.3 Experiment 1: Unsupervised Approach

Gathering a large amount of labeled training data for Semantic Textual Similarity (hereafter; STS) tasks on legal texts is not an easy task. Therefore, unsupervised learning with TSDAE is conducted using a legal corpus. The training data for TSDAE is very simple, requiring only a list of sentences.

Datasets: For unsupervised learning, only legal texts are used. The Electricity Business Act and the Gas Business Act, which are being mapped in this instance, both belong to the industrial category in the e-Gov Legal Document Search. Therefore, all 485 laws in the industrial category were downloaded in XML format, and texts were extracted on a per-paragraph basis, resulting in a total of 42,147 paragraphs.

Training Details: The base model used for training was cl-tohoku/bert-base-japanese-whole-word-masking. The batch size was set to 8, the optimizer used was AdamW, the learning rate was 3e-5, the loss function was DenoisingAuto-EncoderLoss, and the training was conducted for 1 epoch.

4.4 Experiment 2: Supervised Approach

To conduct supervised learning, a labeled dataset like the STS benchmark [1] is necessary. However, no dataset specifically tailored for legal texts has been created. Therefore, in this study, we construct a provisional legal STS dataset and use it for training, evaluating the model’s performance through 5-fold cross-validation.

¹ <https://elaws.e-gov.go.jp/>

² <https://www.gesetze-im-internet.de/>

Datasets: Based on the correct data used in this experiment, labels ranging from -1 to 1, representing similarity, were assigned to all 65,205 pairs of provisions in the ‘Electricity Business Act - Gas Business Act’ combination. There were actually 141 pairs of provisions with a corresponding relationship, for which the label was set to 1. For pairs with no corresponding relationship, the similarity score calculated using the model created in Section 4.3 was reduced by a factor of 0.95 and labeled as a numerical value.

Training Details: The base model used for training was cl-tohoku/bert-base-japanese-whole-word-masking. The batch size was set to 8, the optimizer used was AdamW, the learning rate was 2e-5, the loss function was CosineSimilarity-Loss, and the training was conducted for 3 epochs. During the 5-fold cross-validation, the split of training and validation data was done in an 8:2 ratio and sampled randomly. Ultimately, the training data consisted of 52,614 items, and the validation data consisted of 13,041 items.

4.5 Experiment 3: Using Multilingual Models

In this experiment, to verify whether the same results as in previous research can be achieved using multilingual models, we use the three models shown in Table 1 from Hugging Face without fine-tuning.

Table 1. Multilingual models

LaBSE
sentence-transformers/LaBSE
Supports 109 languages.
Max Sequence Length: 256
Dimensions: 768
Pooling: cls token
MiniLM
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
Trained on parallel data for 50+ languages.
Teacher: paraphrase-MiniLM-L12-v2
Student: microsoft/Multilingual-MiniLM-L12-H384
Max Sequence Length: 128
Dimensions: 384
Pooling: mean tokens
XLM-r
sentence-transformers/paraphrase-xlm-r-multilingual-v1
Trained on parallel data for 50+ languages.
Teacher: paraphrase-distilroberta-base-v1,
Student: xlm-r-base
Max Sequence Length: 128
Dimensions: 768
Pooling: mean tokens

In addition to these models, we compare the results with the pre-trained BERT from previous research [5]. For the mapping, the laws are unified into English, and the model used is bert-base-uncased. The translated Japanese Civil Code was obtained from the Japanese Law Translation Database System, and the German Civil Code was used in the English translation provided by the government.

The length of the input tokens was set to 512, and the average of each token was used as the document vector.

5 Evaluation

5.1 Evaluation for Mapping Japanese Laws

The results of the mapping were compared with the correct data created manually by a legal expert. A true positive (TP) was defined as a case where a mapping was predicted and it matched the correct answer, a false positive (FP) was when a mapping was predicted but did not match the correct answer, a true negative (TN) was when no mapping was predicted and this matched the correct answer, and a false negative (FN) was when no mapping was predicted but this did not match the correct answer. Consequently, Accuracy, Precision, Recall, and the F1 score were calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (7)$$

Accuracy is the overall rate of correctness, Precision is the proportion of provisions that were correctly mapped, Recall represents the proportion of correctly predicted corresponding provisions, and the F1 score is the harmonic mean of Precision and Recall.

5.2 Evaluation for Mapping Japanese and Foreign Laws

This paper focuses on family law. The correct data used is the same as in previous research and is based on the following commentaries on the law:

New Edition Commentary on the Civil Code [7, 9]: This is a sequel series to the ‘Commentary on the Civil Code’ published in 1988 in response to legal revisions. However, this new edition is now over 30 years old.

New Commentary on the Civil Code [8]: The New Commentary on the Civil Code is a series that began publication in 2016. This book was planned as an independent publication, so its content differs from the above series.

(Prohibition of bigamy)

Article 732 A person who has a spouse may not enter into a double marriage.

[Contrast] French Civil Code 147, German Civil Code 1306, Swiss Civil Code 96

[Amendment] 766

Fig. 4. Example of an entry in the New Annotated Civil Code [8]

The New Commentary on the Civil Code includes a notation at the beginning of each article, as shown in Figure 4. Article 732 of the Japanese Civil Code is described as corresponding to Section 1306 of the German Civil Code. Note that several section (article) numbers may be written here. In this paper, we extract this ‘Contrast’ part and list it by article as the correct data. Since only the first half of the New Commentary on the Civil Code is published, the second half has been created from the New Edition Commentary on the Civil Code. For this reason, there is a possibility that some items do not correspond to current law or that the comparative law approach is not consistent.

Moreover, the experiments in this section revealed that one-to-one mapping was difficult, and it was not possible to obtain results sufficient for evaluation. Therefore, we have changed the mapping method as shown in Figure 5. Initially, the top five German laws with the highest similarity from the perspective of Japanese law are output. Then, if any of the top five matches the correct data, it is considered correct. The results are evaluated by the accuracy rate. The Family Law of the Japanese Civil Code contains 183 articles from Articles 725–881, which correspond to 469 sections (Sections 1297–1888) of the German Civil Code Buch 4 Familienrecht. There are 106 corresponding pairs in the correct data.

6 Experimental Results

6.1 Experiments 1 and 2: Unsupervised and Supervised approaches for Mapping Japanese Laws

Table 2 presents the evaluation results for each method. The pre-trained BERT model is referred to as BERT-pretrained, the model trained unsupervised is SBERT-TSDAE, and the model trained supervised is SBERT-STs. The evaluation value for SBERT-TSDAE is the average of 50 experimental runs and performance evaluations. The evaluation for SBERT-STs is from the 5-fold cross-validation. In addition to these models, we also compare the results with the Jaccard coefficient and BM25, which were used in the previous research.

Firstly, mentioning the traditional methods such as Jaccard, BM25, and BERT-pretrained, it is evident that the Jaccard coefficient performs the best in mapping Japanese laws to each other. This is due to the presence of many common words when provisions are in a corresponding relationship. As for BM25

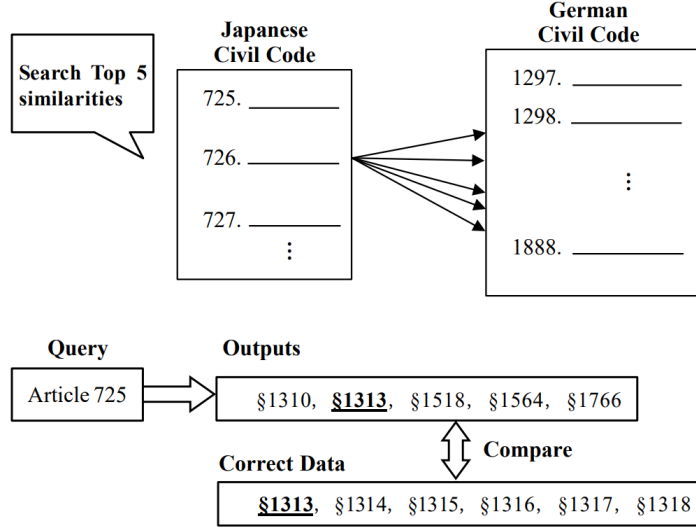


Fig. 5. Mapping between Japanese and German laws

Table 2. Results for experiments 1 and 2

Method	Accuracy	Precision	Recall	F1-score
Jaccard	0.870	0.871	0.812	0.840
BM25	0.841	0.856	0.754	0.802
BERT-pretrained	0.838	0.867	0.731	0.794
SBERT-TSDAE	0.875	0.877	0.817	0.846
SBERT-STS	0.895	0.911	0.836	0.872

and BERT, BM25 was slightly ahead with an F1 score of 0.8pt. Focusing on the BERT model, SBERT-TSDAE, which was fine-tuned using a legal corpus, significantly improved the F1 score by approximately 5.2pt and outperformed BERT-pretrained significantly ($p < 0.05$). Particularly, Recall was greatly improved due to a decrease in FN and an increase in TP, successfully mapping more similar provisions. SBERT-STC surpassed all methods, further improving the F1 score by about 2.6pt compared to SBERT-TSDAE. Similar to SBERT-STC, an increase in Recall was observed, but this model also showed an increase in Precision, indicating that supervised learning outperforms unsupervised learning.

6.2 Experiment 3: Using Multilingual Models for Mapping Japanese and German Laws

Table 3 shows the accuracy rates for each model. bert-base-uncased is the pre-trained English version of BERT, while the others are the multilingual models presented in Section 4.5. CorrectAnswers denotes the accuracy rate. The models paraphrase-multilingual-MiniLM-L12-v2 (MiniLM) and paraphrase-xlm-r-multilingual-v1 (XLM-r) achieved results that slightly exceeded BERT. On the other hand, LaBSE scored lower than BERT. This may be due to LaBSE supporting over 100 languages, while MiniLM and XLM-r are trained on about 50 languages, suggesting that the larger number of supported languages could have impacted the performance negatively.

Table 3. Results for experiment 3

Model	Correct Answers
bert-base-uncased	33/106 = 0.311
LaBSE	30/106 = 0.283
paraphrase-multilingual-MiniLM-L12-v2	34/106 = 0.321
paraphrase-xlm-r-multilingual-v1	38/106 = 0.358

Figure 6 shows an example of successful mapping by all multilingual models. It illustrates provisions from both laws regarding the commencement of legal effects of marriage, showing that they correspond to each other. Thus, it is suggested that the use of multilingual models can enable mapping directly from the original legal texts. However, the current scores are low and not yet practical, so further improvement in performance remains a challenge.

The term ‘paraphrase’ in the names of the models used indicates that they have been trained on paraphrase expressions. It is believed that if a dataset of paraphrase expressions specific to legal texts can be created, performance could be improved through fine-tuning.

Japan: Civil Code (Act No. 89 of 1896) Article 739 Marriage shall take effect upon notification in accordance with the Family Registration Act (Act No. 224 of 1947). The notification set forth in the preceding paragraph shall be made in writing signed by both parties and two or more adult witnesses, or orally by them.
Germany: Civil Code Section 13122 At the marriage ceremony, the registrar shall ask the spouses individually whether they wish to marry each other and, after the spouses have answered this question in the affirmative, pronounce that they are now legally joined spouse. The marriage may take place in the presence of one or two witnesses if the spouses so wish.

Fig. 6. Example of successful mapping in all multilingual models

7 Conclusion

This study set out to develop an automated system for mapping similar provisions across Japanese and foreign laws, with a keen focus on leveraging multilingual models. By avoiding the traditional reliance on translation and instead directly comparing original legal texts, we aimed to preserve the nuanced meanings that are often pivotal in legal discourse. Our experiments demonstrated the feasibility of replicating traditional mapping methods using a multilingual approach, particularly highlighting the enhanced performance in mappings within the Japanese legal framework through the fine-tuning of models with a dedicated legal corpus.

Importantly, our findings reveal that supervised learning significantly outperforms other methods, showcasing the potential of tailored approaches in legal text analysis. This work stands as a substantial contribution to comparative law studies, offering a novel methodology that can simplify the labor-intensive process of legal comparison and potentially streamline the preparation for international legal practices and business expansions.

Reflecting on the journey of this research, it is evident that while multilingual models hold promising capabilities, their practical application faces hurdles, including the creation of accurate data and effective mapping methods. However, the successes and challenges encountered pave the way for future explorations. We advocate for the development of multilingual and legal parallel data to refine the performance of these models further. The creation of such datasets, particularly those that capture the subtleties and variations in legal terminology across languages, will be an invaluable resource for advancing this field.

In conclusion, this study not only highlights the practicality of using multilingual models for mapping legal provisions but also emphasizes the significant potential for improving the efficiency and accuracy of comparative law studies.

As we look forward, the vision for future research includes not just enhancing model performance but also expanding the scope of legal systems and languages examined. Through continued innovation and collaboration, we aim to foster a more interconnected and comprehensible global legal landscape.

Acknowledgment

This research was supported by a grant from the Uchida Energy Science Promotion Foundation (Grant Number R05-1008).

References

1. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S.M., Cer, D., Jurgens, D. (eds.) *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pp. 1–14. Association for Computational Linguistics, Vancouver, Canada (Aug 2017). <https://doi.org/10.18653/v1/S17-2001>, <https://aclanthology.org/S17-2001>
2. Cho, H., Koseki, R., Shima, A., Nakamura, M.: Mapping similar provisions between Japanese and foreign laws. In: Takama, Y., Yada, K., Satoh, K., Arai, S. (eds.) *New Frontiers in Artificial Intelligence*. pp. 36–50. Springer Nature Switzerland, Cham (2023)
3. Cho, H., Nakamura, M.: The relationship between translation accuracy and mapping of similar provisions to foreign laws in Comparative Law Study (in Japanese). In: *Proceedings of IEICE Shin-Etsu Section Conference* (2021)
4. 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)
5. Koseki, R., Cho, H., Nakamura, M.: Mapping similar provisions in comparative law studies using Doc2Vec and BERT (in Japanese). In: *Proceedings of the 29th Annual Conference on Natural Language Processing (NLP2023)*. pp. 182–186 (2023)
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)

10. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) 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>
11. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4512–4525. Association for Computational Linguistics, Online (Nov 2020). <https://doi.org/10.18653/v1/2020.emnlp-main.365>, <https://aclanthology.org/2020.emnlp-main.365>
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
13. Wang, K., Reimers, N., Gurevych, I.: TSDAE: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 671–688. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). <https://doi.org/10.18653/v1/2021.findings-emnlp.59>, <https://aclanthology.org/2021.findings-emnlp.59>

A hybrid approach for accessible rule-based reasoning through large language models ^{***}

Marco Billi¹[0000–0002–6807–073X], Alessandro Parenti^{1,2}[0000–0002–9855–7792]
Giuseppe Pisano¹[0000–0003–0230–8212], and Marco Sanchi^{1,3}[0009–0005–3264–2168]

¹ University of Bologna

{marco.billi3,alessandro.parenti3,g.pisano,marco.sanchi4}@unibo.it

² University of Luxembourg

alessandro.parenti.001@student.uni.lu

³ University of Pisa

marco.sanchi@phd.unipi.it

Abstract. The paper advocates for LLMs to enhance the accessibility, usage and explainability of rule-based legal systems, contributing to a democratic and stakeholder-oriented view of legal technology. A methodology is developed to explore the potential use of LLMs for translating the explanations produced by rule-based systems, from high-level programming languages to natural language, allowing all users a fast, clear, and accessible interaction with such technologies. The study continues by building upon these explanations to empower laypeople with the ability to execute complex juridical tasks on their own, using a *Chain of Prompts* for the autonomous legal comparison of different rule-based inferences, applied to the same factual case.

Keywords: Large Language Models · GPT-4 · CrossJustice · Explainable AI · Accessibility · Chain of Thought

1 Introduction

Rule-based systems are one of the most common type of knowledge-based system used for automated legal reasoning, and they have found application in many different socio-juridical domains, such as credit approval, insurance policy determination, and public organisational structures (healthcare, welfare, pensions, etc.)[7, 10, 8, 16]. These systems contain a knowledge base made up of rules, often traceable to if-then statements, paired with an inference engine which applies them to factual data related to specific cases, in a transparent and explainable way.

In the domain of law, rule-based systems historically hold a well-known series of problems and limitations [14, 13], including that of communicating the output resulting from their expert legal reasoning to laypeople. Such an issue is directly related to how

* The work has been supported by the “CompuLaw” project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement No. 833647).

** All authors contributed equally to this work.

the knowledge base is created: to represent the domain rules in a computable way, a human expert must use specific, high-level programming languages, which ties into how the system then presents its answers to users. The difference between natural language and these programming languages is stark, and as such it requires a way to communicate the output to stakeholders with no computer science background. This creates a critical problem of accessibility. As the aforementioned programming languages are not comprehensible by everyday users, they may not appreciate the output of the rule-based model, nor understand the justification of its legal reasoning.

The problem of communicating the complex syntax and specialised terms used in legal provisions to laypeople is not a novelty, and has been the focus of discussion ever since the sixties [9, 2], under the *plain language movement*. As a matter of fact, improving access to justice has been achieved by working on the lexical aspect of the juridical language, analyzing *layperson ontologies* [15, 3], and applying natural language processing (NLP) to simplify legal documents [11, 4].

A different approach towards the solution of the same problem focuses instead on developing understandable programming languages, such as *Logical English* [6]. As a Controlled Natural Language, *Logical English* resembles natural language in wording, thus increasing the intelligibility of the system to the user and the programmer alike. However, this solution rarely takes into account the possibility for users to directly interact with the system, as these methods appear as static, and do not provide more meaningful information to different users.

In the present paper we tackle this set of issues by developing a methodology focused on employing LLMs for reprocessing the outcomes of rule-based systems in a form that is accessible to laypeople. Large Language Models (LLMs) are a kind of generative artificial intelligence system that leverages deep learning methodologies trained on Big Data, to achieve the processing and creation of human-like text. These models not only hold the ability to successfully generate and manipulate natural language, but also create and model programming languages, including coding script, and as such are being implemented in various fields, including the legal domain. One of the most known and used LLMs at the state of the art is GPT-4, which “*exhibits human-level performance on various professional and academic benchmarks*” according to testing and research conducted by Open AI⁴.

We argue that LLMs, by relying on the legal reasoning provided by rule-based systems, can carry out different legally relevant tasks and present them in a form that is more accessible to end-users, compared to the one produced by the expert system. Our goal is to aid everyday users, lacking both juridical and programming skills, in appreciating the output of rule-based systems and, through the same means, in making more complex legal activities available to them. To test these hypothesis, we provide a case study where we apply the GPT-4 model to the legal reasoning of an established rule-based system, developed using the Prolog language: CrossJustice[1]. In particular, the experiments will focus on providing an accessible natural language explanation of the output of the expert system and operating a comparison between the applications of two different legal sources to the same specific case.

⁴ <https://openai.com/research/gpt-4>

2 Methodology

In spite of their capabilities as natural language processors, LLMs have proven to struggle when they are employed for reasoning about legal norms and their applications. In these cases, such systems have shown to be plagued by hallucinations, lack of coherence and misinterpretation of the norms and the specific context [12, 18]. For this reason, we distance ourselves from these approaches and, instead, bring about a new way to explore the relationship between Large Language Models and rule-based systems that can enhance the access of everyday users to law.

We therefore introduce a hybrid approach where the rule-based system is employed as the legal reasoner, automatically applying the relevant norms to the specific case provided. The rule-based system’s output, presented in the form of a logic programming language, is then fed into GPT-4, for it to be reprocessed and used to carry out different operations.

In particular, our approach implements the *Chain of Thought* [17][5] methodology to create a complex architecture where legal tasks are divided into different prompts, with each of those becoming a link in a series of operations, called chain. In this structure, we use the output of the first prompt as input to the second, as seen in Figure 1, achieving the step-by-step reasoning of the CoT method, while additionally improving its accuracy and flexibility. Moreover, in what we define as a *Chain of Prompts* (CoP) approach, modularity and versatility are introduced in the hybrid system, and users are able to interact autonomously with the model, interrupting the chain of operations at the desired task.

Through our case study, we will show how LLMs can successfully use this solid foundation to operate a natural language explanation of Prolog rules, and then proceed with the comparison between two rule-based inferences applied to the same specific case, highlighting contrasts between them and analysing the potential consequences brought about by their differences, therefore realizing a chain of inferences in line with our methodology.

2.1 Objectives

In developing this system, our main goal is to allow the LLM to output a result in a form that is as accessible as possible to end users, as well as providing all the necessary legal elements to enable a clear understanding of the situation at hand.

To achieve this, we focus on one hand on giving instructions on the structure and lexicon it should be used in the answer, and on the other on ensuring that the direct relation between the specific case provided and the final outcome of the rule-based system is clearly stated in the explanation. In other words, we want the LLM to report and explain which specific conditions of the norm were applied, and exactly which facts triggered this application.

With regards to the more complex legal tasks, we aim at performing a legal analysis on different texts, in order to support both citizens and legal professionals. Our main focus is to ensure that the contrasts between legal sources were clearly highlighted and that the system could explain how these variations may be relevant for the user.

We believe all these information to be crucial for allowing the user to have a clear understanding of their legal position.

Finally, A fundamental step in our methodology is represented by repeating the experiment several times, using the same prompts on the same specific case. Given the intrinsic non-deterministic nature of LLMs, such a procedure allows us to verify whether the approach can provide correct results in a stable and consistent way. To decrease inconsistencies, we set the model's temperature to its minimum, limiting the creativity and inventiveness of GPT-4, thus forcing it to focus on the extraction of legal inferences made by the expert system and the identification of relevant, case-based facts, to be presented in natural language.

2.2 Evaluation Criteria

The results of the approach described above are evaluated according to criteria capable of validating the accessibility and legal soundness of the output. The criteria appear as follows:

- **Correctness**: accuracy in grasping key points, legal issues and essential information by the LLM. This criteria is used to exclude all output which does not match the meaning and legal argumentation of the source provided, overcoming any misinterpretations and misapplications of juridical norms, under a lenses of *juridical validation*.
- **Form**: coherence, readability and simplification of legal vocabulary (*legalese*), to maximise the accessibility of everyday users to the output. Moreover, it verifies the correspondence between input and output in terms of structure and presentation, under the lenses of *formal validation*.
- **Completeness**: inclusion of all the elements requested by the prompt, with particular emphasis on those necessary to evaluate the success of operation. This criteria is used to exacerbate output that did not consider key facts about the overall process, under a lenses of *substantial validation*.

3 Case Study

The CrossJustice platform is a rule-based system capable of automatic legal reasoning in the domain of criminal law, which provides its users with meaningful information about their rights and freedoms as suspects, or accused, of criminal conduct. CrossJustice was identified as suitable as it holds all the main characteristics we ought to look for in order to ground our approach: it explains its inferences, it uses the high-level Prolog programming language to do so, it reasons about extremely different categories of rights, and finally it applies EU law as well as Member State law to justify its inferences.

To introduce the factual scenario provided in Listing 1.1, let us imagine that a person, named *Mario*, is involved in criminal proceedings taking place in Poland (line 10), and does not speak the polish language (line 13). Also, he has been presented with a document charging him of his crime (line 12). According to Article 3, paragraph 2, such a document is to be considered essential (line 11). Thus, according to Article 3, paragraph 1, of the European Directive 2010/64, Mario has the right to receive a translation

of this document, which is essential to ensure that he is able to exercise his right of defence and to safeguard the fairness of the proceedings. This is the main right, presented in the CrossJustice platform in lines 8-9, followed by a recap of the specific conditions needed for that right to be granted.

In the CrossJustice system we also express the relation between a primary right and the further rights that expand the meaning or the implementation of the main.

Auxiliary rights do not directly regulate the legal sphere of the defendant, but are depending upon, or reasonably linked to a primary right. This connection can either be a temporal one, where the right exists only after the primary one has been applied, or a subjective one, which implies that the defendant has particular needs. In this case the defendant, Mario, has the right to have the costs of the translator covered by the State under Article 4 (lines 15-23).

The property norms are used to explicitly define certain characteristics, or details, of a right. The difference between auxiliary rights and properties consists in the fact that the latter exists irrespective of the presence of the defendant, because it attains directly to the main right. In this case, according to Article 3, paragraph 7, an oral translation of essential documents may be provided instead of a written translation on condition that such oral translation does not prejudice the fairness of the proceedings (lines 25-34).

Listing 1.1. Right to Translation - EU Directive

```

1 directive_2010_64 - art3_1
2
3 Article 3
4 Option: essentialDocument
5
6 Explanation:
7
8 has_right(right_to_translation, dir, art3_1, mario, essentialDocument)
9   has_right(art3_1, mario, right_to_translation, essentialDocument)
10     proceeding_language(mario, polish) [FACT]
11     essential_document(art3_2, mario, documents)
12       person_document(mario, charge) [FACT]
13       not(person_understands(mario, polish))
14
15 Auxiliaries:
16
17 art4 - cost - state
18
19 Article 4
20 Explanation:
21
22 auxiliary_right(art4, art3_1, mario, cost, state)
23   auxiliary_right(art4, mario, cost, state)
24
25 Properties:
26
27 art3_7 - form - oral
28
29 Article 3.7
30 Explanation:
31
32 right_property(art3_7, art3_1, mario, form, oral)
33   right_property(art3_7, mario, form, oral)
34     not(proceeding_event(mario, prejudice_fairness))

```

We will now explore the outcome of the same factual scenario, as applied in the Polish legal system (Listing 1.2). Starting from the same basic facts - Mario is involved in criminal proceedings taking place in Poland (line 10), and does not speak the Polish language (line 11) - the Polish legislator states that, according to Article 204, paragraph 2, of the Polish Code of Criminal Procedure, Mario has the right to translation (lines 8-9) if there is a need to translate a document drawn up in a foreign language (line 12). A document presenting a charge is one such document (line 13).

Furthermore, according to Article 618, paragraph 1, part 7, Mario has the right to have the costs of the translator covered by the State (lines 15-23).

There is no applicable article regarding the form of the translation.

Listing 1.2. Right to Translation - Polish Law

```

1 directive_2010_64_pl - article204_2
2
3 Article 204.2 code of criminal procedure
4 Option: documents
5
6 Explanation:
7
8 has_right(right_to_translation, pl, article204_2, mario, documents)
9   has_right(article204_2, mario, right_to_translation, documents)
10     proceeding_language(mario, polish) [FACT]
11     not(person_understands(mario, polish))
12     person_document(mario, translation_needed)
13     person_document(mario, charge) [FACT]
14
15 Auxiliaries:
16
17 article618_7 - cost - state
18
19 Article 618.1.7 code of criminal procedure
20 Explanation:
21
22 auxiliary_right(article618_7, article204_2, mario, cost, state)
23   auxiliary_right(article618_7, mario, cost, state)

```

Given the European nature of the project, the prolog transposition of the Polish legal source has been written using English terms. For the purpose of the present experiments, we did not processed rules expressed in Polish language.

Having established our platform of choice and case study, we are now going to focus on prompt engineering for the achievement of two different tasks following the pipeline described in Figure 1. The first one involves providing a natural language explanation of the pieces of legislation returned by the Crossjustice system in a Prolog form. In the second one, we will use the results obtained in the previous step to compare the two legislations and highlight legally relevant situations.

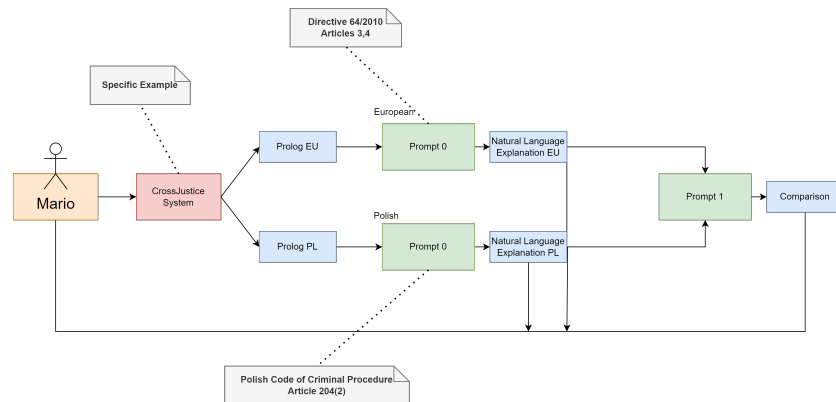


Fig. 1. Pipeline

4 Natural Language Translation

For the first task, our goal was for GPT-4 to be capable of extracting from the Prolog trace the following pieces of information:

- A simplified summary of the norms relevant to the inference;
- The list of rights granted to the user according to the input facts representing the case;
- A description of the inference process that led the system to its solution.

With these objectives in mind, we directed our efforts towards achieving the most flexible and versatile prompt, able to be applied to any of the inferences from the Crossjustice system.

There were however several problems we encountered. An initial challenge was to obtain a fixed structure in the output, which was often presented in different formatting (e.g. bullet lists, numbered lists, free-form text). We found that it was necessary to give a fixed structure for the LLM to follow, in order to decrease the degree of variance between answers and maximise repeatability and reliability.

Second, we required the LLM to take into consideration all legal terms and facts from the scenario, and to apply those to build the explanation, as it had a propensity to miss key facts if not well prompted.

Finally, the most challenging aspect was to overcome the tendency of the model to ascribe meaning to legal text where not explicitly mentioned nor provided. In particular, the meaning was often guessed based on the most general significance, disregarding the specificity of juridical norms and legal lexicon.

Through trial and error, we achieved the following prompt:

```
1 You have been provided a Prolog inference tree using a legal norm in a specific case (Prolog Tree).
2 Provide the following info according to the given structure:
3
4 Summary: simplified text of the legal norm. Use everyday language with a serious register. Add any
5 information you have on the source of law;
6 What Rights do You Have: what rights do you have according to the Prolog explanation;
7 Why do You Have Them: inference steps and reasoning that led to the the rights. Use all the Prolog terms in
8 the explanation explicitly referencing the original Prolog when needed.
9
10 Use enumerations in the 'What Rights do You Have' and 'Why do You Have Them' sections if needed.
```

First, we introduced the object and the form of the input data (line 1). Second, we provided a fixed structure to follow (Summary - What Rights do You Have - Why do You Have Them). This is aimed at guaranteeing that the model follows the given formal criteria, in order to maximise repeatability and reliability.

Furthermore, we specifically requested the system to use all the Prolog terms in the explanation, with explicit references to the original (line 6). This is aimed at providing an easier way to read and evaluate the answer, both from a substantial and juridical point of view.

Finally, an example of the answers we obtained using the prompt follows:

```
1 Summary: According to Directive 2010/64, Article 3.1, if you are involved in a legal proceeding and do not
2 understand the language of the proceeding, you have the right to translation of essential documents.
3 The cost of the translation will be covered by the state, as stated in Article 4. Additionally, under
4 Article 3.7, you may have the right to oral translation if it does not prejudice the fairness of the
5 proceeding.
6
7 What Rights do You Have:
8 1. Right to translation of essential documents.
```

```

5 2. State covering the cost of translation.
6 3. Right to oral translation, if it does not prejudice the fairness of the proceeding.
7
8 Why do You Have Them:
9 1. You have the right to translation of essential documents because:
10 - Your proceeding language is Polish (proceeding_language(mario, polish)).
11 - You have essential documents related to the charge (person_document(mario, charge)).
12 - You do not understand Polish (not(person_understands(mario, polish))).
13 - This right is granted under Directive 2010/64, Article 3.1 (has_right(right_to_translation, dir, art3_1
14 , mario, essentialDocument)).
15 2. The state will cover the cost of translation because:
16 - This auxiliary right is granted under Directive 2010/64, Article 4 (auxiliary_right(art4, art3_1, mario
17 , cost, state)).
18 3. You have the right to oral translation if it does not prejudice the fairness of the proceeding because:
19 - This right is granted under Directive 2010/64, Article 3.7 (right_property(art3_7, art3_1, mario, form,
20 oral)).
21 - The fairness of the proceeding is not prejudiced (not(proceeding_event(mario, prejudice_fairness))).

```

It appears clear how GPT-4 was able to present its output in an accessible and readable way, upholding the instructions given about structure and formatting, even after much repetition (*formal validation*). However, we cannot say that the next two criteria have been fully satisfied. GPT-4 did not include all the relevant Prolog facts used to apply juridical reasoning to the specific case (*substantial validation*), as it failed to grasp and correctly represent one of the legal inferences of CrossJustice, based on the application of the the sub-rule (*juridical validation*).

We noticed how GPT-4 struggles in giving the exact meaning to Prolog terms when these may be open to different interpretations. In particular, in line 11, GPT reports the fact that Mario has an essential document related to the charge as a condition of the Prolog rule. The predicate has been misinterpreted, and a better version would have been: “*a document containing a charge*”, or “*a document which states that the person has been charged*”. Furthermore, this has lead the LLM to mix up two facts, of which one is a condition for the application of the other. In this case, the correct solution would have been to identify that a document would be considered essential **if** that document was a charge.

```

1 Summary: According to Article 204.2 of the Code of Criminal Procedure, if you are involved in a legal
2 proceeding and do not understand the language of the proceeding (Polish in this case), you have the
3 right to translation of necessary documents. Additionally, Article 618.1.7 states that the cost of
4 this translation will be covered by the state.
5
6 What Rights do You Have:
7 1. Right to translation of necessary documents.
8 2. The cost of translation will be covered by the state.
9
10 Why do You Have Them:
11 1. You have the right to translation because:
12 a. The proceeding language is Polish (proceeding_language(mario, polish)).
13 b. You do not understand Polish (not(person_understands(mario, polish))).
14 c. You have a document that needs translation (person_document(mario, translation_needed)).
15 d. The document in question is a charge (person_document(mario, charge)).
16 e. Therefore, you have the right to translation according to Article 204.2 (has_right(
17 right_to_translation, pl, article204_2, mario, documents)).
18
19 2. You have the right to have the cost of translation covered by the state because:
20 a. The auxiliary right in Article 618.1.7 is connected to the right in Article 204.2 (auxiliary_right(
21 article618_7, article204_2, mario, cost, state)).
22 b. Therefore, you have the right to have the cost of translation covered by the state (auxiliary_right(
23 article618_7, mario, cost, state)).

```

Here, all 3 criteria have been fully satisfied. The performance of the LLM visibly improved, as both the sub-condition that the person Mario is presented with a document containing a charge (line 12), and that because of this Mario has a document that needs translation (line 11), have been correctly identified. This highlights the contrast between the Natural Language translation of the same Prolog fact (“*person_document(mario, charge)*”) applied to the two corresponding legal sources, which has been interpreted differently, for no apparent reason. These small but substantial mistakes might be a

consequence of the limited context provided to the model. However, even when experimenting by providing the LLM the full text of the relevant legal norms, we found that it did not cause a substantial improvement in performances, nor in the language and terminology used.

Overall, we could not find a way to reliably prompt the system to correctly identify and present all sub-rules and conditions, although mistakes were significantly lowered throughout the experimenting process.

5 Comparison of legal sources

Building upon the results of Task 1, we followed by instructing GPT-4 to enact legal comparison between two sources. To reach a successful result, we experimented with several prompts. We also tested beforehand the capacity of the LLM to produce legal comparison directly on the text of the norm; however, the results were extremely poor.

We noticed that, especially for more complex tasks, employing a *Chain of Thought*[17, 5] approach decreases the probabilities of mistakes in the final answer. *Chain of Thought* prompting consists of having an LLM generate a series of intermediate reasoning steps necessary to get to the final answer.

To implement this method, we first tried to have a single prompt describing multiple logical steps, ranging from extracting the information to the analysis of the differences. However, we found inconsistency in the answers provided by the LLM, possibly because of the length of the step by step process and the high number of actions required. We therefore decided to divide the prompt into two sub-prompts, each related to a specific task. The first one – described in Section 4 – requiring an explanation of both the legal sources provided, and the second one requiring their comparison based on the output of the previous step. Doing so, we created a *Chain of Prompts* capable of enabling laypeople not only to better access and understand legal reasoning, but also to execute complex juridical operations such as comparison in full autonomy.

Our goal was for the model to identify the following:

- Differences and contrasts in the legal reasoning and application of the norms to the case between the sources.
- Diverging interpretations of the rights present in the legal sources and how those may lead to different applications of the law to the specific case.

This subsequent operation resulted in the application of the following prompt to the output obtained from the first step:

```

1 You have received two legal sources. Compare them according to the following steps:
2 1 - COMPARE the differences in the reasonings. USE all the inference steps and describe how they influence
   the final outcome;
3 2 - analyse potential consequences of the differences from step 1 only based on the data provided.
```

Here we followed the same principles as per the initial prompt. First, we introduced the object of the text given as input, although now in natural language, so no clarification regarding the form was needed. Second, we fixed the structure to be followed when providing an answer. Again, we required the system to use all the inference steps in order to minimise the risk of misinterpretation and hallucinations.

An example of the final results gathered using the *Chain of Prompts* in discussion follows:

```

1 1. Comparison of differences in reasonings:
2
3 Directive 2010/64:
4 - The right is granted by Article 3.1 of Directive 2010/64.
5 - The right is specifically for the translation of "essential" documents.
6 - The state covers the costs of translation as per Article 4.
7 - There is an additional right to oral translation if it does not prejudice the fairness of the proceeding,
   as per Article 3.7.
8
9 Code of Criminal Procedure:
10 - The right is granted by Article 204.2 of the Code of Criminal Procedure.
11 - The right is specifically for the translation of "necessary" documents.
12 - The state covers the costs of translation as per Article 618.1.7.
13 - There is no mention of an additional right to oral translation.
14
15 The differences in the reasonings lie in the terminology used for the documents that require translation (
   essential vs. necessary) and the presence of an additional right to oral translation in Directive
   2010/64. The sources of the rights also differ, with one being a European directive and the other
   being a national code of criminal procedure.
16
17 2. Potential consequences of the differences:
18
19 - The difference in terminology (essential vs. necessary) might lead to different interpretations of which
   documents qualify for translation. This could potentially affect the scope of the right to translation
   and the number of documents that are translated for the individual.
20 - The presence of an additional right to oral translation in Directive 2010/64 could provide more
   comprehensive language assistance to the individual during the legal proceeding, ensuring better
   understanding and participation in the process. This right is not mentioned in the Code of Criminal
   Procedure, which might limit the language assistance provided to the individual.
21 - The difference in the sources of the rights (European directive vs. national code) could lead to potential
   conflicts or discrepancies in the application of these rights, depending on the jurisdiction and the
   specific case. This might affect the individual's access to these rights and the overall fairness of
   the legal proceeding.

```

With regards to the *formal validation* criteria, GPT-4 was able to present its output in an accessible and readable way, upholding the instructions given about structure and formatting, even after much repetition.

Before going into the details of the next two criteria, a clarification is necessary. One of the issues we faced when using a single prompt lied in the LLM's understanding of the words "*documents*" and "*needed*". We observed a tendency to ignore the context of the two words when together, and replace them with the expression "*all documents*", regardless of our insistence on the use of the correct terms as extracted by the Prolog trace. This created multiple issues from both a substantial and a juridical point of view, as it missed inference steps in its comparison, as well as misrepresented legal terms and inferences. After changing our approach, we saw a steep increase in the quality of the outcome, as can be seen from the example above.

Currently, one of the few remaining limitations is due to the Prolog representation of the norm. In this case, GPT-4 cannot reliably infer that the Code of Criminal Procedure belongs to Poland. This is achieved in the CrossJustice platform through the use of a visual interface, while in the Prolog representation this is done through the use of the suffix *.pl*. Furthermore, the LLM still has trouble highlighting the fact that the same document has two different interpretations according to the two applicable legal sources.

To conclude with the *juridical validation* criteria, the LLM has correctly grasped the relevant terms and compared them without changing their meaning. However, it does not include all the relevant inference steps used to apply juridical reasoning to the specific case (*substantial validation*).

6 Conclusions

This paper explores the opportunities and limitations regarding the use of LLMs for the autonomous generation of accessible natural language explanations, within the context of rule-based systems in the legal domain. Moreover, it tackles the possibility of building upon these explanations to empower stakeholders with the ability of enacting autonomous legal tasks, such as comparing the application of different norms to the same specific case.

To reach our first objective, we provided a methodology for the engineering of flexible prompts, able to process juridical inferences in a stable, repeatable and simplified way. We followed by successfully applying our hybrid approach to the CrossJustice platform - a system based in Prolog language for the domain of criminal law - showing our method to be effective in making the rule-based reasoning accessible, while preserving its substantial, juridical and formal validity.

After establishing such sound foundation, we moved to our second objective by creating a chain of prompts able to process different rule-based outputs and their explanations. Our goal was to produce legal comparison by identifying the relevant juridical and factual differences present amongst inferences relating to the same specific case. The methodology proved to be once again successful, showing the potential of a hybrid approach based on the expert reasoning of rule-based systems, paired with the versatility of LLMs, which opens the door to various legal operations, as shown in the case study.

On this note, future works would expand our methodology by further developing the *Chain of Prompts* used in our trials, exploring the potential *modularity* of such an approach. By this, we mean the creation of a versatile and flexible initial prompt for the natural language translation and explanation of rule-based inferences, followed by the engineering of multiple different and subsequent prompts, each dedicated to a different legal operation. Those would be applied accordingly to the output of the first, creating chains going beyond normative comparison, enabling more complex and differentiated operations.

Finally, our methodology could easily embrace the multilingual nature of European Law given the capabilities of state-of-the-art LLMs, thus contributing to overcome language barriers in the fruition of legal technology, as well as to bolster access to European and Member State law.

References

1. Billi, M., Calegari, R., Contissa, G., Pisano, G., Sartor, G., Sartor, G.: Explainability through argumentation in logic programming. In: Proceedings of the International Conference on Logic Programming 2021 Workshops co-located with the 37th International Conference on Logic Programming (ICLP 2021), Porto, Portugal (virtual), September 20th-21st, 2021. CEUR Workshop Proceedings, vol. 2970. CEUR-WS.org (2021), <https://ceur-ws.org/Vol-2970/causalpaper2.pdf>
2. Charrow, R.P., Charrow, V.R.: Making legal language understandable: A psycholinguistic study of jury instructions. *Columbia law review* **79**(7), 1306–1374 (1979)

3. Fernández-Barrera, M., Casanovas, P.: From user needs to expert knowledge: mapping laymen queries with ontologies in the domain of consumer mediation. In: *AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal Language and Legal Ontologies, Argumentation and Software Agents: International Workshop AICOL-III, Held as Part of the 25th IVR Congress, Frankfurt am Main, Germany, August 15-16, 2011. Revised Selected Papers 3*. pp. 286–308. Springer (2012)
4. Garimella, A., Sancheti, A., Aggarwal, V., Ganesh, A., Chhaya, N., Kambhatla, N.: Text simplification for legal domain: Insights and challenges. In: *Proceedings of the Natural Legal Language Processing Workshop 2022*. pp. 296–304 (2022)
5. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. *Advances in neural information processing systems* **35**, 22199–22213 (2022)
6. Kowalski, R.: Logical english. *Proceedings of Logic and Practice of Programming (LPOP)* (2020)
7. Lymer, A., Richards, K.: A hybrid-based expert system for personal pension planning in the uk. *Intelligent Systems in Accounting, Finance and Management* **4**(1), 71–88 (1995)
8. Masri, N., Sultan, Y.A., Akkila, A.N., Almasri, A., Ahmed, A., Mahmoud, A.Y., Zaout, I., Abu-Naser, S.S.: Survey of rule-based systems. *International Journal of Academic Information Systems Research (IJASIR)* **3**(7), 1–23 (2019)
9. Mellinkoff, D.: *The Language of the Law*. Little, Brown & Co (1963)
10. Michaelsen, R.H.: An expert system for federal tax planning. *Expert Systems* **1**(2), 149–167 (1984)
11. Paquin, L.C., Blanchard, F., Thomasset, C.: Loge-expert: from a legal expert system to an information system for non-lawyers. In: *Proceedings of the 3rd international conference on Artificial intelligence and law*. pp. 254–259 (1991)
12. Savelka, J., Ashley, K.D., Gray, M.A., Westermann, H., Xu, H.: Explaining legal concepts with augmented large language models (gpt-4). *arXiv preprint arXiv:2306.09525* (2023)
13. Schauer, F.: *Playing by the rules: A philosophical examination of rule-based decision-making in law and in life*. Clarendon Press (1991)
14. Susskind, R.: *Expert systems in law*. Oxford University Press, Inc. (1987)
15. Uijttenbroek, E.M., Lodder, A.R., Klein, M.C., Wildeboer, G.R., Van Steenberghe, W., Sie, R.L., Huygen, P.E., Van Harmelen, F.: Retrieval of case law to provide layman with information about liability: Preliminary results of the best-project. Springer (2008)
16. van Melle, W.: Mycin: a knowledge-based consultation program for infectious disease diagnosis. *International Journal of Man-Machine Studies* **10**(3), 313–322 (1978). [https://doi.org/https://doi.org/10.1016/S0020-7373\(78\)80049-2](https://doi.org/https://doi.org/10.1016/S0020-7373(78)80049-2), <https://www.sciencedirect.com/science/article/pii/S0020737378800492>
17. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: *NeurIPS* (2022), http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html
18. Westermann, H., Meeüs, S., Godet, M., Troussel, A.C., Tan, J., Savelka, J., Benyekhlef, K.: Bridging the gap: Mapping layperson narratives to legal issues with language models. In: *Proceedings of the 6th Workshop on Automated Semantic Analysis of Information in Legal Text co-located with the 19th International Conference on Artificial Intelligence and Law (ICAIL 2023), Braga, Portugal, 23rd September, 2023. CEUR Workshop Proceedings, vol. 3441, pp. 37–48. CEUR-WS.org* (2023), <https://ceur-ws.org/Vol-3441/paper5.pdf>

A Hybrid Approach to Legal Textual Entailment

Cor Steging¹ and Ludi van Leeuwen¹

¹Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence,
University of Groningen

Abstract. Large language models and other data-driven natural language processing models are black boxes that cannot guarantee sound decision-making. Within the context of legal textual entailment, providing arguments for a decision is essential. We therefore propose a hybrid legal textual entailment system that combines knowledge representation techniques with large language models. As part of the COLIEE2024 competition, our system is tasked with answering yes-no bar exam questions, based on provided relevant articles of the Japanese Civil Code. As team Hybrid Intelligence (HI), we represent these articles as Abstract Dialectical Frameworks using the ANGELIC methodology and use a large language model to ascribe factors for each question. In a set of two experiments, we show that our hybrid system can perform as well as our baseline large language model and can provide explanations for its decisions.

Keywords: Hybrid Intelligence · Explainable AI · Natural Language Inference

1 Introduction

Natural Language Inference (NLI) deals with determining whether a hypothesis is true for a given premise in a text. For humans, this task requires us to not only understand the text, but to also reason about it. Previous research in the field of NLI has developed a variety of approaches to create systems that aim to solve these types of task [15]. Currently, Large Language Models (LLM) are the state of the art in most natural language tasks, including NLI. Yet, studies have shown that there are limitations of using LLMs in NLI tasks [10]. Not only do LLMs often make the wrong predictions, but we cannot explain their decision-making either; they are black boxes. While LLMs can be prompted to provide explanations for their decisions, there is no guarantee that these are faithful. We as humans have the capability to explicitly reason about why something should be true or not, but LLMs cannot guarantee sound decision-making. Data-driven models have been shown to make the right decisions for the wrong reasons [6,13], which can lead to irresponsible behavior. Without a method to evaluate their internal rationale, the black box remains unopened.

Symbolic AI methods, on the other hand, do not have this issue, as their behavior is explicitly defined in their design. These symbolic methods, however,

do not possess the same level of performance as the LLMs when it comes to natural language tasks. We therefore propose a hybrid system, that combines symbolic AI with data-driven LLMs. We apply this system to Task 4 of the 2024 version of the Competition on Legal Information Extraction/Entailment (COLIEE), which deals with legal textual entailment.

2 Background

In Task 4 of the COLIEE, artificial systems are asked to solve a set of yes-no questions from the Japanese bar exam. Each exam question contains one or multiple relevant legal articles S from the Japanese Civil Code and a statement Q , which makes a claim C . The goal is to determine whether or not the relevant articles S legally entails the claim C in statement Q . As an example, we will look at bar exam question *H18-2-2*, shown in Figure 1. In this example, the claim C in statement Q is that the rescuer does not need to pay damages. If we examine the story in Q using relevant Article S (Article 698), we can see that this is the right conclusion: the individual engaged in a benevolent intervention in order to allow the other individual to escape imminent danger to their person, and the former is therefore not liable to compensate for damages resulting from the act. There is therefore a legal entailment.

Relevant Articles S	Statement Q
Article 698 If a manager engages in benevolent intervention in another’s business in order to allow a principal to escape imminent danger to the principal’s person, reputation, or property, the manager is not liable to compensate for damage resulting from this unless the manager has acted in bad faith or with gross negligence.	In cases where an individual rescues another person from getting hit by a car by pushing that person out of the way, causing the person’s luxury kimono to get dirty, the rescuer does not have to compensate damages for the kimono.

Fig. 1: H18-2-2: example of a bar exam question.

Last year in the same Task 4 of the competition, most approaches used a large language model (LLM) to predict legal entailment [9]. These models use prompt engineering to find the best prompt for their model. Some of the approaches used prompts that contained a number n examples of bar exam questions (n -shot prompting), whereas others used no examples in their prompts (zero-shot prompting).

While these LLMs performed well, some concerns were raised by the COLIEE organisation [9]. First of all, in some cases, the training data of the LLM is not disclosed, which means that we cannot know whether it contains texts that are

similar to the bar exam questions of the COLIEE, giving an unfair advantage. Secondly, some of the external resources used can be difficult to replicate, which is why it is not allowed to use models such as OpenAI’s GPT models for predicting entailment at the competition, just as in the previous year. Lastly, the COLIEE organisation notes that LLMs cannot explain their decision making and cannot guarantee that their decisions are based on sound logic. In this study, we address this third concern.

Previous research focused on sound decision-making in data-driven models has shown that these models can perform well for the wrong reasons [6, 13]. In the field of law, however, arguments for a decision are as important as the decision itself. Various formal theories exist within the domain of AI & law that aim to create knowledge representations of legal principles [2, 3]. These normative representations of the law allow us to explicitly reason about cases based on expert knowledge of the domain. In this study, we take a hybrid approach, aiming to integrate the adaptivity of machine learning techniques with the sound reasoning and interpretability of symbolic knowledge [1].

In our system, we represent the legal articles of the Japanese Civil Code as Abstract Dialectical Frameworks (ADF) using the ANGELIC methodology [4]. ANGELIC was created specifically for representing and reasoning about legal domains as ADFs. The ADF of a legal domain is a hierarchical structure, in our case representing a single legal article of the Japanese Civil Code. The root node of the ADF is the verdict, which is determined by a set of issues. Each issue is made up of a set of factors, which represent legally relevant fact patterns. We distinguish between two types of factors: abstract factors and baselevel factors. Abstract factors represent intermediary legal concepts and are determined by other abstract or baselevel factors. Baselevel factors are the leaf nodes of the ADF. Each baselevel factor has an associated value that represents whether that factor applies to the given case. The values of the baselevel factors can be ‘True’, ‘False’, or ‘Unknown’, as the ADF can deal with unknown values using default logic. Using this approach, we can create ADF representations of legal articles of the Japanese Civil Code and we provide examples of such ADFs later on in the study.

In earlier research, the ANGELIC approach has been used to model Article 6 of the European Convention on Human Rights as an ADF [8], which in turn was used in a set of experiments investigating sound reasoning [14]. The ADF of Article 6 of the European Convention on Human Rights has also been combined with NLP models (BERT), laying the foundation for a hybrid approach for predicting court cases [11]. Unlike data-driven models, such as LLMs, ADFs are knowledge representations that are explainable by design and follow a predefined decision process. This means that they do not only provide a final decision, but also sound argumentation for that decision. ADFs are, however, formal models that cannot process natural language. We therefore propose a hybrid legal entailment system that combines the natural language processing of large language models with the sound reasoning of formal knowledge representations.

3 Method

Our hybrid legal entailment system combines knowledge and data by creating ADFs for legal articles and ascribing the values of the factors of those ADFs using a large language model. Ascribing factors is the process of determining whether the baselevel factor of the apply to the given statement or not, or whether this is unknown. For each bar exam question (S, Q) , finds the ADF associated with article S . Using a large language model, we answer questions about the statement Q in order to ascribe the baselevel factors of ADFs. We then evaluate the verdict of the ADF and compare this verdict to the claim C made in statement Q . If these match, there is a legal entailment. We explain these steps in further detail in the upcoming section.

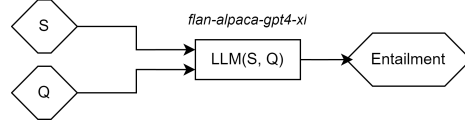
To investigate the effect of including knowledge representations in the legal entailment task, we evaluate three versions of our system, represented in Figure 2a, 2b, and 2c respectively:

1. **Version 1** uses only a large language model to predict the legal entailment with zero-shot prompting, similar to the state of the art [7].
2. **Version 2** uses a small selection of manually crafted ADFs to predict entailment. The large language model is used as a backup classifier when bar exam questions pertain to legal articles outside of our selection of ADFs.
3. **Version 3** uses artificially generated ADFs in addition to the manually crafted ADFs to predict entailment. The large language model is used as a backup classifier.

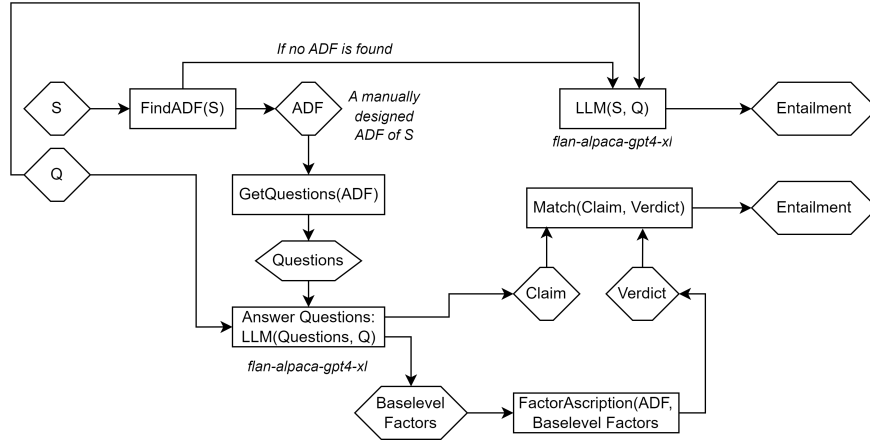
3.1 Large Language Model

The input of our large language model is a prompt that includes the bar exam question (S, Q) , and the output is a yes or no. Our approach follows the same steps as the best performing system of the previous COLIEE competition (team JNLP) [7]. We use the `flan-alpaca-gpt4-xl` model from the HuggingFace Transformer library [17], as `flan-alpaca-xxl` yielded the best performance in previous iterations of the competition. In order to find the best prompt for this model, we used the PromptSource library [5], just as team JNLP did last year. Using PromptSource, we selected all prompts associated with the SuperGLUE dataset [16] and investigated which prompt performed best for Task 4. In last year’s iteration of the competition, JNLP used the prompts of the GLUE dataset, containing NLI tasks. SuperGLUE is an improved version of the GLUE dataset. Since we opt to use zero-shot prompting, we can evaluate each prompts across all training and test data. The best performing prompt is the following, where S represents the articles of the exam question and Q the statement:

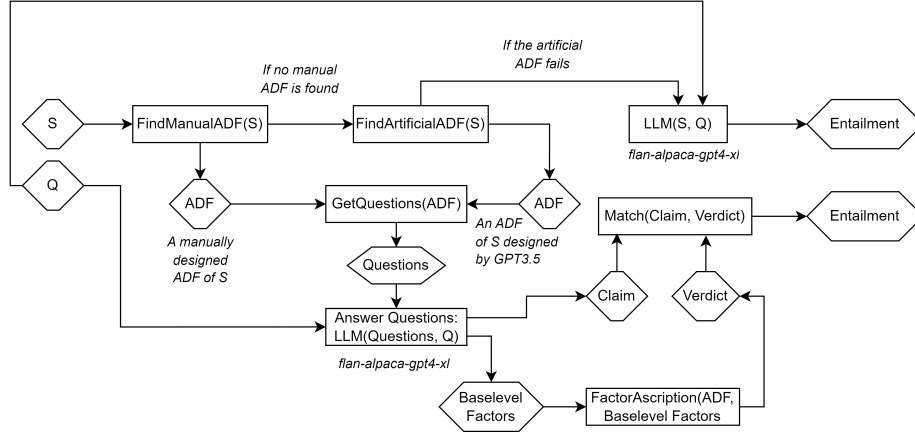
```
Exercise: read the text and answer the question by True or False.  
'S'  
Question: Q?
```



(a) Version 1: Predicting entailment using a large language model.



(b) Version 2: Predicting entailment using manually designed ADFs that represent legal articles, combined with large language models.



(c) Version 3: Predicting entailment using both manually designed ADFs and artificially generated ADFs, combined with large language models.

Fig. 2: The pipeline for the three different versions of our system. We predict legal entailment for a bar exam question (S, Q) , where S represents the legal articles and Q the statement.

We use this prompt for `flan-alpaca-gpt4-xl`. The output of the model should be a binary ‘yes’ or ‘no’ therefore needs to be parsed. We do this by looking at the words in the output. If the output contains positive words, such as ‘yes’, ‘true’, ‘correct’ or ‘right’, the answer will be ‘yes’. Likewise, if the output contains negative words, such as ‘no’, ‘false’, ‘incorrect’ or ‘wrong’, the output will be ‘no’. Additionally, we account for negation as well, such that ‘not true’ evaluates to a ‘no’.

In version 1 of our system, we use this large language model to determine legal entailment. In the other two versions of the system, this large language model is used as a backup classifier.

3.2 Manually crafted ADFs

In the second version of our system, we use human domain knowledge to create Abstract Dialectical Frameworks of legal articles which can be used to determine legal entailment. The system is created with the goal of using sound reasoning and being able explain its behavior. We create an ADF for 25 of the legal articles. These 25 articles were selected based on their frequency in the dataset and ease of implementation. To illustrate, we present the ADF of Article 698, which was used in example question H18-2-2 (Figure 1), graphically in Figure 3, alongside its verdict and factors. Connections ending in an X in Figure 3 represent an attack from one argument to another, whereas connections without an X represent support from one argument to another.

In Figure 3b, we can see that the verdict of Article 698 deals with the liability to compensate for damages. This verdict is determined by a set of factors. Factors can support the verdict or other factors, as indicated by the arrow in Figure 3a. Factors can also attack these support relationships, as indicated by the crosses in Figure 3a. For example, if there were damages resulting from the intervention (*dmg*), this would support the verdict that the manager is liable to compensate for damages. If, in this example, the manager’s actions aimed to allow the escape of imminent danger to the person of the principal (*per*) and the manager’s intervention was benevolent (*ben*), then the support of *dmg* for the verdict would be attacked, and the verdict would change such that the manager is not liable to compensate for damages.

For any given case, each baselevel factor of the ADF is given a value: True, False or Unknown. The values of the abstract factors are in turn determined by the values of their baselevel factors. These abstract factors can be seen as a generalized legal concept. The value of the verdict, True or False, is determined through the values of all factors, following their logical relationships, such as the one shown in Figure 3a.

To use the ADFs to predict legal entailment, we need a way to ascribe the correct values to the baselevel factors of the ADF. For bar exam question (S_x, Q_x) , we have have an ADF A_x that represents article S_x . The values of the baselevel factors of A_x should then be based on the content of statement Q_x . For Article 698, we therefore need to determine the values of *ben*, *dmg*, *per*, *rep*, *pro*, *neg*, and *fth* (See Figure 3b). To determine the values of these baselevel factors, we ask a

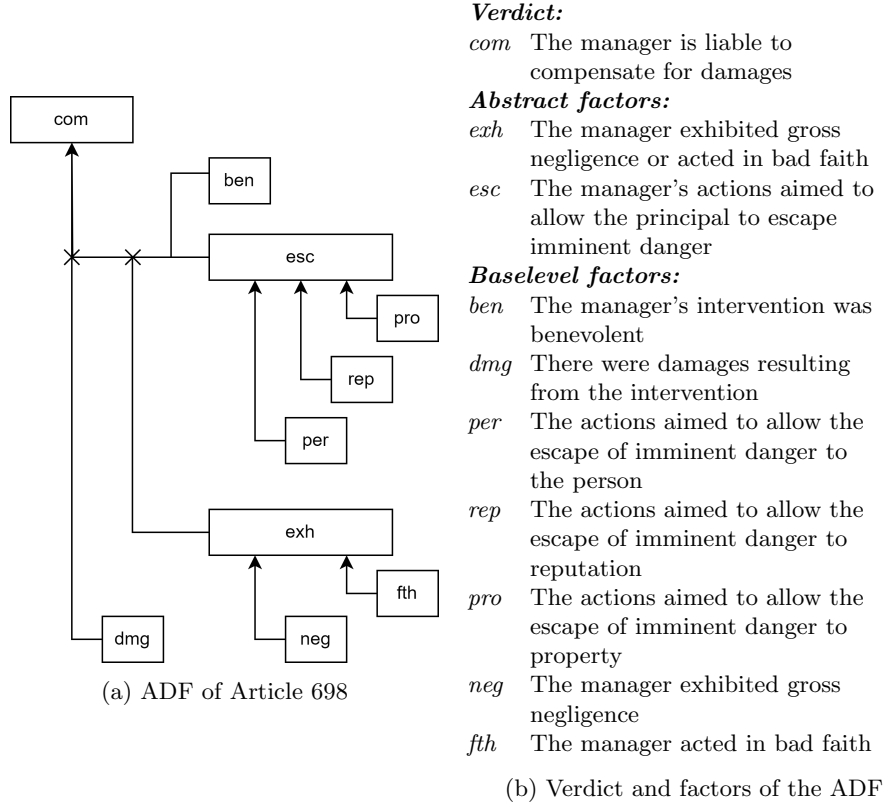


Fig. 3: The Abstract Dialectical Framework of Article 698 (A) and its verdict and factors (B).

large language model (flan-alpaca-gpt4-xl) questions about Q_x . For example, if we want to know the value of baselevel factor *ben* for Q_x , we would provide the following prompt to the LLM:

Passage: ' Q_x '
 After reading this passage, I have a question:
 Was the manager's intervention benevolent?
 Answer only with 'True', 'False', or 'Unknown'

We provide such a prompt to the LLM for each of the baselevel factors of our ADF A_x . Once we have the values of all baselevel factors, we can determine the value of the verdict V_x .

In order to determine legal entailment, we need to find out if the verdict V_x matches the claim made in the question Q_x regarding the verdict. If these match, then there is a legal entailment, and if they do not match, there is no

legal entailment. We therefore need to find out what claim C_x is made in the statement Q_x , which we do by asking our LLM (flan-alpaca-gpt4-xl) what claim was made in Q_x regarding the verdict. For Article 698 and statement Q_x , the prompt would be as follows:

```

Passage: ' $Q_x$ '
After reading this passage, I have a question:
Is the manager liable to compensate for damages?
Answer only with 'True', 'False'

```

This prompt yields us the value of claim C_x . For example case *H18-2-2* (see Figure 1), the value for C be False, as the the claim made in Q is that the manager is not liable to compensate for damages. In the last step of our pipeline, we determine entailment by comparing claim C_x made in statement Q_x to verdict V_x as determined by ADF A_x . There is a legal entailment if C_x and V_x match, and no legal entailment otherwise. The full pipeline can be seen in Figure 2.

Since we only created ADFs for 25 out of the 781 legal articles, we fall back to the LLM and prompts from version 1 of our system as a backup classifier in case the bar exam question pertained to an article for which we do not have an ADF. We also did not account for bar exam questions pertaining to multiple legal articles, and use the LLM from version 1 in these situations as well.

3.3 Artificially generated ADFs

In the second version of our system we were only able to create ADFs for 25 of the legal articles. In the third version of our system, we create an ADF for each of the 781 legal articles by generating them artificially.

For the artificial conversion of legal articles to ADFs we use the GPT3.5-turbo API from OpenAI [12]. We use 1-shot prompting, where we provide one example of an article (Article 698) and its ADF (See in Figure 3). The prompt also includes the legal article for which we want to make the ADF and the command to convert that article into an ADF. Using Article 698 S_{698} and its respective ADF A_{698} , we would write the following prompt for legal article S_x :

```

Read this article: ' $S_{698}$ '
This article can be represented as an ADF as follows:  $A_{698}$ 
Now read this second legal article:  $S_x$ 
Convert this second legal article into an ADF.

```

The resulting output then contains ADF A_x , which represent legal article S_x . A light post-processing was then applied to ensure that the code generated by GPT would work within our pipeline. This includes small tweaks to the code such as ensuring that the right libraries are imported and that the final ADF has the correct variable name.

Not all of the ADFs generated by GPT3.5-turbo result in executable code. The model sometimes outputs an apology rather than an ADF, or code that does

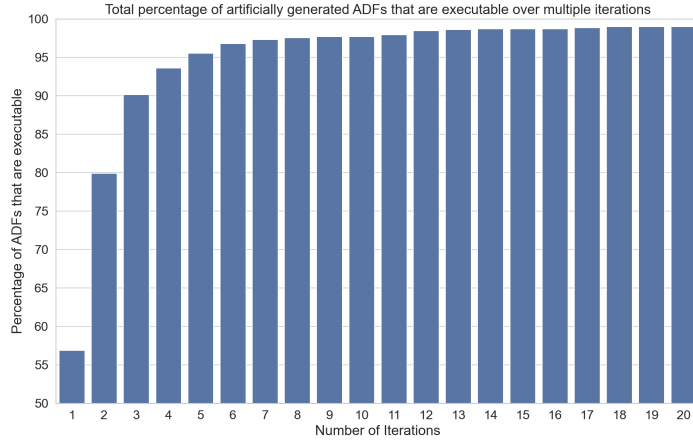


Fig. 4: Due to the non-deterministic behavior of the model, GPT3.5-turbo generates more ADFs that are executable over multiple iterations.

not follow the correct Python syntax. However, due to the non-deterministic nature of the model, running the same prompt again usually fixes this issue. We therefore automatically check whether each artificially generated ADF is executable code and re-run the prompt if it is not. This process is done iteratively, leading to an increasingly larger number of executable ADFs. We plot the number of executable ADFs versus the amount of iterations in Figure 4. After 20 iterations, 773 out of the 781 legal articles were successfully converted to an ADF. The remaining 8 articles only contain the word ‘deleted’ in the provided dataset.

We should note that we do not check whether these ADFs are logically sound or match the legal articles, but only whether they do not cause any errors upon loading their module. Some of the ADFs still cause errors during runtime, for example when the logic is not sound or when variable names do not match. In those cases, the system automatically falls back to the backup classifier from system 1. Additionally, if an exam question pertains to a legal article for which we have a manually crafted ADF, version 3 of the system uses the manually crafted ADF instead of the artificially generated ADF. Version 3 of our system, therefore prefers the manually crafted ADFs, then the artificially generated ADFs and then the backup classifier (flan-alpaca-gpt4-xl).

4 Results

In the COLIEE competition, the performance of a system is evaluated on four test sets: the test set for this year’s iteration of the competition (set 2024) and the test sets of previous iterations (H30, R01, R02). We therefore first report

Table 1: The mean accuracy and standard deviations of the three different versions of our system on each of the four test sets.

Test set	Version 1	Version 2	Version 3
H30	64.23 $\pm$ 4.13	64.23 $\pm$ 4.35	54.00 $\pm$ 3.42
R01	61.77 $\pm$ 3.28	61.01 $\pm$ 3.19	56.50 $\pm$ 3.76
R02	64.44 $\pm$ 4.37	63.65 $\pm$ 3.53	50.50 $\pm$ 3.25
2024	67.08 $\pm$ 3.28	65.03 $\pm$ 3.21	57.69 $\pm$ 2.46
Mean	64.38 $\pm$ 3.77	63.48 $\pm$ 3.57	54.67 $\pm$ 3.22

the performances of our systems on these four test sets. Secondly, in version two and three of our system, the ADFs are not always used, and sometimes the large language model is used as a backup. We therefore also analyse our system and report on how often the system uses the hand-crafted ADFs, the artificially generated ADFs or the the large language model. Finally, to investigate the effect of including knowledge representations in the legal entailment task, we also report the performances of all three of our systems on all of the bar exam questions that pertain to articles for which we have manually crafted ADFs. These questions are a subset that we created of all of the data, which we refer to as the limited domain dataset.

4.1 Performance in COLIEE

We first report on the performance of the systems on the four test sets (H30, R01, R02, 2024). Because of the nondeterministic behavior our large language model, we run our system on each of the four test sets 25 times and report the mean performances and their standard deviations. For each version of the system, and each of the four test sets, we report the accuracy in Table 1 and the Matthew’s Correlation Coefficient (MCC) in Table 2 respectively. The accuracy is used as the official performance metric of the COLIEE competition, whereas the MCC, which is scaled between -100 and 100, accounts for all four quadrants of the confusion matrix, thus giving us more insight in the performance of the system. For the MCC in Table 2, we also report the maximum performance of each version of system for each test set across all 25 runs. For interpreting the accuracy results, we should note that the label distribution of the test sets is balanced.

When we aggregate over all four test-sets, we get a distribution of MCC values for every run, for every version of the system, as seen in Figure 5a. We performed an Analysis of Variance (ANOVA) test which shows that there is a significant difference between the mean MCC score of the versions ($F_{2,297} = 133.6, p < 0.01$). A post-hoc Tukey HSD Test finds that the mean MCC score for version 3 was significantly lower ($M = 12.73, sd = 7.18$) than the mean MCC scores for version 1 ($M = 31.84, sd = 9.94$) and version 2 ($M = 30.60, sd = 10.30$), at $p < 0.01$. There is no significant difference in performance between the MCC scores of version 1 and version 2 of the system.

Table 2: The mean, maximum and standard deviation (SD) of the Matthew’s Correlation Coefficient of the three different versions of our system across the four test sets. The best results are shown in bold.

Test set	H30			R01			R02			2024		
Version	1	2	3	1	2	3	1	2	3	1	2	3
Mean	29.35	28.94	8.16	22.80	21.90	15.18	41.27	42.56	12.28	33.94	28.99	15.32
Max	40.90	43.61	20.33	33.01	38.66	28.10	52.10	54.72	27.07	50.68	43.41	25.19
SD	8.70	8.79	6.91	6.74	6.33	7.63	6.19	5.69	6.87	7.72	7.38	4.92

4.2 System analysis

For each of the four test sets, we report on how often the systems could use the manual ADFs, the artificially generated ADFs and the large language model. These percentages can be found in Table 3. In version 1 of our system, the LLM is used in 100% of the cases. In version 2, the manual ADF is used if it exists and otherwise the backup LLM is used. On average, in 7.67% of the exam questions a manual ADF could be used (see Table 3). That means that when we use version 2 of our system, in 7.67% of the exam questions a manual ADF is used and in the remaining 92.33% the backup LLM is used. Version 3 of the system prefers the manual ADFs, then the artificially generated ADFs and then the backup LLM. So if version 3 is used, on average in 7.67% of the exam questions the manual ADFs are used, in 70.33% the artificially generated ADFs are used, and in the remaining 21.99% the backup LLM is used.

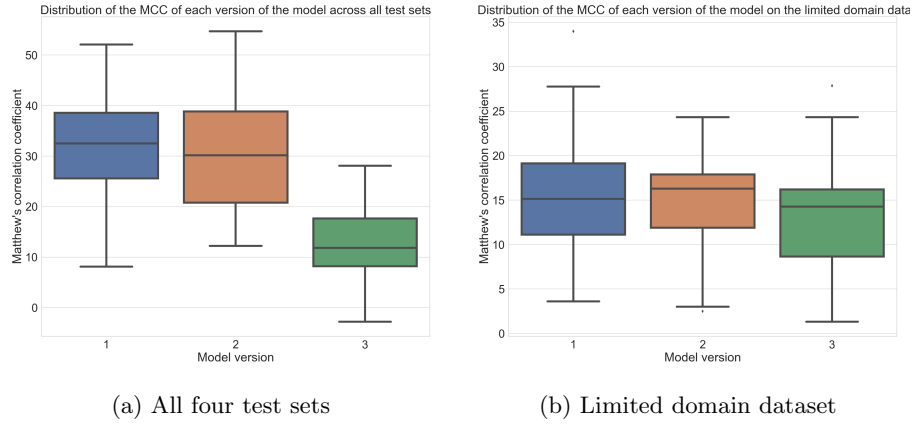


Fig. 5: The distribution of the Matthew’s Correlation Coefficient (MCC) for each version of the system across all four test sets (A) and across the limited domain dataset (B).

Table 3: The percentage of exam questions for which version 3 of our system uses manually crafted ADFs, artificially generated ADFs or the large language model for each of the four test sets.

Test set	Manual	Artificial	LLM
H30	7.14	75.71	17.14
R01	14.41	63.96	21.62
R02	6.93	64.36	28.71
2024	1.83	78.90	19.27
Mean	7.67	70.33	21.99

4.3 Explainability

Version 2 and version 3 of our system use a combination of LLMs and ADFs. The latter are inherently interpretable, and thus can be used to generate explanations. For example case *H18-2-2* (see Figure 1), version 2 and version 3 of our system provide the explanation as seen in Figure 6.

There is a legal entailment.

We examined the question: ‘Is the manager not liable to compensate for damages?’

The text claims that this is True and according to the legal articles it is True.

We believe this to be the case, as we can answer yes to the following questions:

Was the manager’s intervention benevolent?

Did the manager’s actions aim to allow the principal to escape imminent danger

to the principal’s person, reputation, or property?

Was there any damage resulting from the intervention?

And we do not know the answer to the following questions:

Did the manager act in bad faith

Did the manager exhibit gross negligence

Fig. 6: Explanation provided by version 2 and 3 of our system for exam question H18-2-2

4.4 Investigating the effect of knowledge

We have created a total of 25 manual ADFs, each representing a different legal article. In all of the data, excluding test set 2024, 107 exam questions are found that pertain to these legal articles. Exam questions with multiple articles are

excluded. We refer to these 107 questions as the *limited domain dataset*. The distribution of the performance of our three different systems on the limited domain dataset can be found in the boxplot in Figure 5b. Note that the y-axis is scaled differently than in Figure 5a.

Since manual ADFs are available for each exam question in the limited domain dataset and version 3 of our system always prefers to use manual ADFs over artificial ADFs, version 2 and version 3 of our system are effectively identical. We performed an ANOVA test to find that there is no significant difference in performance between the three versions of our system, as measured by the mean MCC score of each of the versions on the limited domain dataset ($F_{2,87} = 0.7703, p > 0.05$).

5 Discussion

We can see a high variation in the performance of our systems, as evident from the standard deviations in Tables 1, 2 and the high variance in Figure 5. This variance is solely due to the non-deterministic nature of our LLM, as the ADFs themselves are deterministic. Lowering the temperature parameter of the LLM, which defines the randomness or creativity of the response, could reduce this variance in performance, but might also have an impact on the performance itself. Fine-tuning this parameter is therefore required for future research. We chose our LLM based on the best performing system of the last iteration of the COLIEE [7], which had yielded an accuracy of 78.22%. Version 1 of our system, which uses only the LLM, only yielded an accuracy of 64.4%. A better performance can thus be gained by either fine-tuning the LLM, doing more prompt engineering, or selecting a different LLM.

On average, there was no statistical difference between the performance of version 1 and version 2 of our system in the experiment across all test sets (Figure 5a) and across the limited domain dataset (Figure 5b). While there is no significant difference in performance, version 2 of our system provides explanations, such as the ones shown in Figure 6. Additionally, the highest performing version 2 system yields a higher MCC than the highest performing version 1 system for three of the four test sets, as can be seen in Table 2. In the right circumstances, version 2 may therefore slightly outperform version 1. On the four test sets, the backup classifier was used in most cases when using version 2 of our system, as seen in Table 3. More and better manually crafted ADFs could improve the performance of version 2 on these test sets. On the limited domain dataset, where the backup classifier is not used in version 2 of our system, we also see no significant difference in performance compared to version 1. Version 2, the hybrid approach, therefore yields the same performance as version 1, the large language model, and can explain its decision-making.

The performance of version 3 was significantly worse than the performance of the other two versions across the four test sets, as seen in Figure 5a and Table 1 and 2. Across these test sets, version 3 used the artificially generated ADFs for 64.36%-78.9% of the exam questions. Our method for generating ar-

tificial ADFs is therefore worse than using a zero-shot LLM. Additionally, the artificially generated ADFs are worse than the manually crafted ADFs. A manual inspection showed that many of the artificially generated ADFs contained structures that are illogical or incorrect. Further research is needed on how to extract logical structures from natural language automatically. At the moment, creation of knowledge representations of the law is therefore best left to human experts.

On the limited domain dataset, there was no significant difference between version 2 and 3, because version 3 always prefers manually crafted ADFs over artificial ADFs, and every case in the limited domain dataset has an associated manually crafted ADF. In that experiment, no artificial ADF was used by version 3, making it identical to version 2. The only variations are due to the non-deterministic nature of the LLM.

While the ADF is inherently explainable, the factor ascription is not, as this is done by the large language model. We do therefore not know why a certain factor was ascribed for a given case. However, due to the explanations of our system, we do have a way of finding out when and where factor ascription has gone wrong. We also should note that our manually crafted ADFs were not created by legal experts. However, the structure of an ADF is human-readable (see Figure 3), and would allow a legal professional to proof-read or check the ADFs that have been created, in contrast to the black-boxes of pure language models. In future research, we would like to collaborate with legal experts to create knowledge representations of legal domains. Additionally, our current implementation cannot handle bar exam questions with multiple articles, which is a feature we would like to incorporate at a later stage.

In this study, we present a proof of concept for a hybrid system for legal textual entailment. This system combines large language models with knowledge representation of the law to make its predictions. We show that the performance of the current implementation rivals that of a baseline large language model. Additionally, the system can provide an explanation for its decision-making, which is essential in the field of law.

Acknowledgements

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, <https://hybrid-intelligence-centre.nl>.

References

1. Akata, Z., Balliet, D., de Rijke, M., Dignum, F., Dignum, V., Eiben, G., Fokkens, A., Grossi, D., Hindriks, K., Hoos, H., Hung, H., Jonker, C., Monz, C., Neerincx, M., Oliehoek, F., Prakken, H., Schlobach, S., van der Gaag, L., van Harmelen, F., van Hoof, H., van Riemsdijk, B., van Wylsberghe, A., Verbrugge, R., Verheij,

- B., Vossen, P., Welling, M.: A research agenda for hybrid intelligence: Augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer* **53**(8), 18–28 (2020)
2. Aleven, V.: Teaching case-based argumentation through a model and examples. Ph.D. thesis, University of Pittsburgh (1997)
3. Ashley, K.D.: Modelling legal argument: Reasoning with cases and hypotheticals. (1989)
4. Atkinson, K., Bench-Capon, T.: Angelic ii: An improved methodology for representing legal domain knowledge. In: ICAIL '23: Nineteenth International Conference for Artificial Intelligence and Law, Braga, Portugal, June 19 - 23, 2023. ICAIL '23, ACM (2023)
5. Bach, S., Sanh, V., Yong, Z.X., Webson, A., Raffel, C., Nayak, N.V., Sharma, A., Kim, T., Bari, M.S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Bendavid, S., Xu, C., Chhablani, G., Wang, H., Fries, J., Al-shaibani, M., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Radev, D., Jiang, M.T.j., Rush, A.: PromptSource: An integrated development environment and repository for natural language prompts. In: Basile, V., Kozareva, Z., Stajner, S. (eds.) *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pp. 93–104. Association for Computational Linguistics, Dublin, Ireland (May 2022). <https://doi.org/10.18653/v1/2022.acl-demo.9>, <https://aclanthology.org/2022.acl-demo.9>
6. Bench-Capon, T.: Neural networks and open texture. In: *Proceedings of the 4th International Conference on Artificial Intelligence and Law*. pp. 292–297. ICAIL '93, ACM, New York (1993)
7. Bui, Q.M., Do, D.T., Le, N.K., Nguyen, D.H., Nguyen, K.V.H., Anh, T.P.N., Nguyen, M.L.: JNLP @COLIEE-2023: Data augmentation and large language model for legal case retrieval and entailment. In: Erich, S. (ed.) *In Workshop of the tenth Competition On Legal Information Extraction/ Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and law (ICAIL)*. pp. 17–27 (2023)
8. Colletette, J., Atkinson, K., Bench-Capon, T.: Explainable ai tools for legal reasoning about cases: A study on the european court of human rights. *Artif. Intell.* **317**, 103861 (2023)
9. Goebel, R., Kano, Y., Kim, m.y., Rabelo, J., Satoh, K., Yoshioka, M.: Overview and discussion of the competition on legal information, extraction/entailment (coliee) 2023. *The Review of Socionetwork Strategies* (01 2024)
10. Gubelmann, R., Kalouli, A.I., Niklaus, C., Handschuh, S.: When truth matters - addressing pragmatic categories in natural language inference (NLI) by large language models (LLMs). In: Palmer, A., Camacho-collados, J. (eds.) *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics*. pp. 24–39. Association for Computational Linguistics, Toronto, Canada (jul 2023)
11. Mumford, J., Atkinson, K., Bench-Capon, T.: Reasoning with legal cases: A hybrid ADF-ML approach. *Legal Knowledge and Information Systems* **362**, 93–102 (2022)
12. OpenAI: Gpt-3.5-turbo (2023), <https://platform.openai.com/docs/models/gpt-3-5>
13. Steging, C., Renooij, S., Verheij, B.: Discovering the rationale of decisions: towards a method for aligning learning and reasoning. In: Maranhão, J., Wyner, A.Z. (eds.) *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*. pp. 235–239. ACM (2021)
14. Steging, C., Renooij, S., Verheij, B.: Improving rationales with small, inconsistent and incomplete data. In: Giovanni Sileno, Jerry Spanakis, G.v.D. (ed.) *Legal*

Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, the Netherlands, 18–20 December 2023. vol. 379, pp. 53–62. IOS Press (2023)

15. Storks, S., Gao, Q., Chai, J.Y.: Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. CoRR **abs/1904.01172** (2019)
16. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Superglue: A stickier benchmark for general-purpose language understanding systems. CoRR **abs/1905.00537** (2019)
17. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Liu, Q., Schlangen, D. (eds.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>

Legal Yes/No Question Answering Through Text Embedding, Fine-Tuning, and Prompt Engineering

Housam Khalifa Bashier Babiker², Md Abed Rahman^{1,2}, Mi-Young Kim³,
Juliano Rabelo¹, and Randy Goebel^{1,2}

¹ Alberta Machine Intelligence Institute, University of Alberta, Canada

² Department of Computing Science, University of Alberta, Canada

³ Department of Science, Augustana Faculty, University of Alberta, Canada
{khalifab,mdabed,miyoung2,rabelo,rgoebel}@ualberta.ca

Abstract. The challenge of information overload in the legal domain increases every day. The International Competition on Legal Information Extraction/Entailment (COLIEE) has created four challenge tasks that are intended to encourage the development of systems and methods to alleviate some of that pressure: a case law retrieval (Task 1) and entailment (Task 2), and a statute law retrieval (Task 3) and entailment (Task 4). Here we describe our methods for Task 3 and Task 4. In Task 3, we use text embeddings to learn better semantic representations. In Task 4, our approach relies on leveraging the power of newer Large Language Models (LLMs) through prompt engineering and fine-tuning. Our method for Task 4 was ranked third among 25 submitted models and nine participating teams in the COLIEE 2024 competition.

Keywords: Legal Information Retrieval · Legal Information Entailment
· COLIEE 2024 · Legal Question Answering.

1 Introduction

Every day, large volumes of legal data are produced by law firms, law courts, independent attorneys, legislators, regulators, and many others. Within that context, the disciplined management of legal information becomes manually intractable and requires the development of tools that automatically or semi-automatically aid legal professionals in managing information overload. The International Competition on Legal Information Extraction/Entailment (COLIEE)⁴ addresses four facets of that challenge: case law retrieval, case law entailment, statute law retrieval, and statute law entailment. Here we summarize our approaches to statute law retrieval and statute law entailment tasks, evaluate the results achieved, and comment on future work to further improve our models.

The COLIEE statute law retrieval task (Task 3) requires identifying and retrieving appropriate Japanese Civil Code Articles relevant to a given query sentence.

⁴ <https://sites.ualberta.ca/~rabelo/COLIEE2024/>

COLIEE Task 4 focuses on legal entailment, i.e. predicting the entailment relationship between a legal query and statutory law articles; this is done by comparing a representation of the meaning of the legal query and the civil law articles. In general, this task requires participants to develop algorithms capable of reading a query and a law article (or multiple law articles) and then determining whether the law article(s) entail the legal query or does not. In other words, the goal of the statute law entailment (Task 4) is to construct yes/no question-answering systems for legal queries, by confirming the entailment of a query from articles. The answer to a question is typically determined by measuring semantic similarity between question and answer. Because the legal bar exam query and articles are complex and varied, we need to carefully determine what kind of information is needed to confirm textual entailment. Here we exploit the power of recent large LLMs’ ability to process large volumes of text as contextual information and then produce an output using zero-shot learning given the appropriate prompt.

Our approach for Task 3 relied on text embeddings as a proxy for semantic representations for articles and queries for the information retrieval task. To do so, we fine-tuned a well-known state-of-the-art text sentence embedding model called AngleE [14]. The fine-tuning stage aims to maximize the similarity between similar articles and queries in the training set. This approach achieved an F2-measure of 0.6526 in the official test dataset.

Our approach for Task 4 relies on the approach to “prompt engineering” on the Large Language Model (LLM) called *flan_t5_xxl* [6] and subsequently fine-tuning the last two layers of the same LLM with a chosen prompt. Finally, we chose three good prompts with varied performance levels and used a simple voting-based ensemble of their predictions. This approach achieved an accuracy of 0.7982 in the official test dataset, which ranked third amongst 25 submissions of the nine competitors in Task 4 of the COLIEE 2024 competition.

Our paper is organized as follows: Section 2 presents a brief state-of-the-art review; Sections 3 and 4 describe our method in more detail. Section 5 provide an analysis of our results. Section 5 include a fault analysis of some models. Finally, Section 6 provides some final remarks and proposes future work.

2 Literature Review

Most current approaches to legal information retrieval rely on traditional information retrieval (IR) methods and more recently, transformer-based large language model (LLM) techniques. Here we briefly summarize some of the most successful approaches in Task 3, proposed in recent editions of the COLIEE competition. CAPTAIN [15] used LLM-based ranking models; Tohoku BERT for Japanese and monoT5 for English. One of their models was ranked first in COLIEE 2023. The best performance system used the ensemble of these two results. JNLP [5] utilized ensembles of BM25 for Japanese and LLM-based ranking models for English and monoT5. Their system was ranked second in COLIEE 2023.

Similarly, we briefly summarize some of the most successful approaches in Task 4, proposed in recent editions of the COLIEE competition.

They [5] used zero-shot models of LLMs, by gathering all the prompts from the GLUE tasks available in the PromptSource library, and selecting 56 prompts. Among their three models, JNLP1 used google/flan-t5-xxl model⁵, JNLP2 used google/flan-ul2 model⁶, and JNLP3 used declare-lab/flan-alpaca-xxl model⁷, respectively, to run the prompts which were the given problem-article pairs inserted. Their model JNLP3 was ranked first, JNLP1 was ranked 2nd, and JNLP2 was ranked third in COLIEE 2023.

KIS [16] extended their previous system which performs data augmentation and uses an ensemble of BERT-based and rule-based models, to integrate LUKE⁸, the named entity-enhanced Transformer. KIS1 uses the pre-trained LUKE model, KIS2 uses a fine-tuned LUKE model for the alphabetical person-included dataset, and KIS3 uses another fine-tuned LUKE model without the alphabetical person-included dataset. KIS2 was ranked fourth in the COLIEE 2023.

The methods for learning text embedding can be classified into two categories: **Unsupervised methods** such as BERT-flow [13], can be used to improve the quality of the embeddings by mapping BERT embeddings to a standard Gaussian latent space as compared to earlier studies [17] and [10]. Other approaches to unsupervised text embeddings focused on using contrastive learning methods to improve the power and quality of text embeddings such as [30].

Supervised methods can also be used to learn text embeddings. For example, SBERT [19] modified the BERT-based model to learn better semantic representations. The SBERT model uses a Siamese network architecture to improve the semantic representation of legal texts. Recent advanced methods rely on ideas from prompt engineering to learn better semantics such as [11] and to define a new unsupervised training objective to enhance BERT embeddings.

Over the last few years, LLMs have been used for Natural Language Inference (NLI) across several benchmarks. Few-shot learning [22] and self-explaining models [29] are used in the state-of-the-art models on the Stanford Natural Language Inference (SNLI) dataset [28]. For the Question Answering NLI (QNLI) dataset [20], BERT [7]-based models seem to perform the best [12, 23]. MultiNLI [27], an NLI dataset made from multiple data sources is another well-known benchmark dataset. T5 [18] and other BERT-based models that perform well on the QNLI dataset perform well on this dataset.

Prompt Engineering and its importance in developing LLMs has garnered recent attention. White et al. [25] added to previous research by investigating prompt engineering patterns and their applicability to software development. Maddigan and Susnjak [1] explore the capabilities of prompt engineering for natural language understanding to generate visualizations. Han et al. [8] researched strate-

⁵ <https://huggingface.co/google/flan-t5-xxl>

⁶ <https://huggingface.co/google/flan-ul2>

⁷ <https://huggingface.co/declare-lab/flan-alpaca-xxl>

⁸ luke-japanese-base-lite

gies for automatic prompt generation to facilitate classification. Wang et al. [21] have looked into the potential of prompts in the generation of effective queries for literature review.

3 Our Method for Task 3

3.1 Task Description

Task 3 requires the retrieval of an appropriate subset ($S_1, S_2, \dots, S_n$) of Japanese Civil Code Articles from the Civil Code texts dataset, used for answering a Japanese legal bar exam question Q . An appropriate subset means the identification of a subset of statutes for which an entailment system can judge whether the statement Q is true or not.

3.2 Dataset Analysis

For the training dataset, we considered training data from previous years. To better learn the representations, we used all available XML files from the COLIEE dataset, except R04, to fine-tune the proposed models. The R04 dataset is used as the evaluation set.

3.3 Details of our Approach

Since task 3 is a retrieval task, we focused on training representations that would make the queries have the highest cosine similarity with the related articles. Thus, our first approach re-purposed (fine-tuned) a text embedding model called UAE-Large-V1 [14] to handle the retrieval task.

UAE-Large-V1 is a sentence embedding model that is based on the bert-large uncased model from Huggingface.⁹ pretrained on embedding data¹⁰. The embeddings were further trained on the multi-nli [26], QRECC [2], SNLI [28], Quora question pairs [24] and MRPC dataset [9]

For Task 3, we finetuned the model on the training data. For each query, we take the articles that are relevant as positive samples and we randomly select 10 articles that are not relevant as negative samples. We followed the same training objective as in [14] and then fine-tuned the model on the COLIEE dataset. For inference, because we used the learned embeddings to represent both articles and queries, we calculated the cosine similarity to find the top-k relevant articles for a given query. We call this model UA-Angle

Our second approach is called “UA-Angle-chunk.” It uses the same model as the first approach, but during testing, we employ a customized chunking approach to increase the priority of individual sentences of each article. For example, given a query and an article, we first split the article into different sentences using a combination of ‘;’, ‘,’ and ‘.’ in addition to considering the full

⁹ <https://huggingface.co/google-bert/bert-large-uncased>

¹⁰ <https://huggingface.co/embedding-data>

article contents. For ranking, we again rank them according to cosine similarity. During evaluation, if multiple sentences from a single article are found to be relevant, the sentence with the highest similarity score is considered the similarity score for the entire article and is ranked accordingly. The reason for employing this approach is that we observed that with some articles, only a small portion of the text was relevant, and using the entire article’s representation added unnecessary noise that hampered the ranking process.

For our third approach, we used a pre-trained sentence transformer model called ‘all-mpnet-base-v2’¹¹. The sentence transformer model maps the sentences and paragraphs to a 768 dimensional dense vector. The mapped vector representation is used with the cosine similarity to find relevant articles for a given query. To improve this model, we fine-tuned the COLIEE dataset.

For all fine-tuned models, we used the COLIEE datasets from previous competitions all XML files are used for training, except R04.

4 Our Method-Task 4

4.1 Task Description

In Task 4, the problem of answering a legal yes/no question can be viewed as a binary classification problem. We assume that a set of questions Q , where each question $q_i \in Q$ is associated with a list of corresponding article sentences $\{a_{i1}, a_{i2}, \dots, a_{im}\}$, where $y_i = 1$ if the answer is “yes” and $y_i = 0$ if “no.” The process proceeds by choosing the most relevant sentence a_{ij} . Therefore, our task is to learn a classifier over these triples so that it can predict the answers of any additional question-article pairs.

4.2 Dataset Analysis

For Task 4, the training set contained 14 files enumerated as H18 to H29. We also included R03 and R04. H30, R01, and R02 were kept aside for additional testing results which were part of the COLIEE 2024 competition submission. Table 1 contains the number of samples as well as the distribution of labels for the various training and testing sets.

Table 1. Number of samples and Distribution of labels in Datasets for Task 4

Dataset Name	#of samples	Percentage (%) of positive samples
Training Set [H18 - H29]	1206	45.9
H30	70	51.4
R01	59	53.2
R02	81	46.9
Test Set [R05]	109	–

¹¹ <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

As we can see both training and test sets are somewhat balanced, albeit the training set slightly favors the "No" [non-entailment] class.

4.3 Details of our Approach

Our approach to the inference task uses zero-shot learning on a well-known LLM through in-context-learning [4] and prompt engineering. As demonstrated through prior work, LLMs can perform well in zero-shot learning. We chose the model `flan-t5-xxl` [6] for our experiments. This model is based on pretrained T5 [18] which was pre-trained on C4¹², a cleaned version of the Common Crawl dataset¹³ which contains hundreds of gigabytes of English text scraped from the web. This model was fine-tuned for better zero and few-shot performance.

For prompts, we chose the PromptSource [3] library. We also augmented the prompts with in-context learning, i.e. adding examples to the prompt. We added one positive and one negative example from the training data as part of each prompt and experimented on the rest of the training set (barring the two examples) to find good prompts. During inference, specific positive phrases such as "yes" were searched in the generated text that indicates entailment. Then we chose the top 3 prompts based on their overall accuracy on the training set, as well as how they differ in terms of predicting the "yes" (entailment) and "no" (non-entailment) classes. Table 2 shows the top 3 prompts chosen. Finally, we performed zero-shot inference with all three prompts, and the final prediction was based on simple voting on the three prompts' results.

Table 2. Captions used for Task 4

Prompt Name	Prompt Text
GPT-3 style	{{premise}} Question: {{hypothesis}} True or False?
Can we Infer	Suppose {{premise}} Can we infer that {{hypothesis}}? Yes or no?
Always/Never	Suppose it's true that {{premise}}. Then, is {{hypothesis}} always, or never true?

We call this model `UA_stack`. We also report results on the top prompt (dubbed `UA_gpt`).

Finally, we took the last two layers from both the encoder and decoder of `flan-t5-xxl` and fine-tuned in on the training data for task 4 on all XML files available except R01, R02, and H30. We set the Encoder sequence length is set to 512 and the Decoder sequence length is set to 5. The top prompt (same used in `UA_gpt`)

¹² <https://huggingface.co/datasets/c4>

¹³ <https://commoncrawl.org>

was used in fine-tuning. Four A100 GPUs were used in the fine-tuning. We call this model UA_encoder_decoder.

5 Results

5.1 Task 3 Results

The Task 3 results for COLIEE 2024 for the retrieval task are summarized in Table 3. The best performance, given our three submissions, is based on fine-tuning the UAE-Angle model which seems to capture better semantic representation as compared to other approaches and achieves a recall score of 0.7339.

Table 3. Information retrieval (Task 3) results on test data considering only the best system in each team

Team	Model ID	Precision	Recall	F2
JNLP.constr-join	JNLP.constr-join	0.6502	0.7982	0.7408
CAPTAIN.bjpAllMonoT5	CAPTAIN.bjpAllMonoT5	0.6713	0.7752	0.7335
TQM-run1	TQM-run1	0.7202	0.7339	0.7706
NOWJ-25mulreftask-ensemble	NOWJ-25mulreftask-ensemble	0.7081	0.6334	0.7081
AMHR02	AMHR02	0.5972	0.7569	0.6887
UA-angleE	UA-angleE	0.5596	0.7339	0.6526
BM24-1	BM24-1	0.2590	0.7294	0.4945
MIG2	MIG2	0.1604	0.1881	0.1665
PSI01	PSI01	0.0826	0.0780	0.0785

5.2 Task 4 Results

Table 4 contains the results on test data for COLIEE 2024. We submitted 3 results (UA_stack, UA_gpt, and UA_encoder_decoder). We report only the best results for each team. Two of our models (UA_stack and UA_encoder_decoder) perform the same. We report only UA_stack in Table 4.

While both UA_stack and UA_encoder_decoder perform the same (i.e. have the same accuracy), their performance is different for the two classes. This warrants future investigation.

6 Conclusion

We have explained our use of various language models for legal information retrieval and question-answering in COLIEE 2024. For the statute law retrieval task (Task 3), we used a state-of-the-art text embedding model. In Task 4, we used in-context learning along with prompt engineering to perform zero-shot classification on an LLM to get results. We also fine-tuned the last two encoder

Table 4. NLI (Task 4) results on test data considering only the best system in each team

Team	Model ID	Correct	Accuracy
–	Baseline	Yes 60/All 109	0.55
CAPTAIN	CAPTAIN2	90	0.8257
JNLP	JNLP1	89	0.8165
UA	UA_stack	87	0.7982
AMHR	AMHR.ensembleA50	84	0.7706
HI	HI1	82	0.7523
NOWJ	NOWJ.pandap46	82	0.7523
OVGU	OVGU1	77	0.7064
KIS	KIS2	76	0.6972
MIG	MIG	69	0.6330

and decoder layers of the LLM on the training set. Both approaches of our system were ranked third among 25 submitted models.

Our results on Task 3 showed that re-purposing (fine-tuning) the model for the legal domain might boost the recall but not the precision. Also, the Angle model outperforms the other model as it is good at handling the vanishing gradient problem in traditional sentence embedding models, by modifying the objective function and by optimizing the angle difference from the cosine function in complex space. In future work, we would like to consider different approaches to augment the data and as well as handle the negative sampling more effectively for the task.

In Task 4, Our reported model UA_stack has a balanced performance for both classes “yes” (entailment) and “no” (non-entailment), i.e. our model is equally likely to perform well on both classes. UA_gpt is slightly better at predicting the “yes” (entailment) class as opposed to the “no” (non-entailment) class. On the other hand, UA_encoder_decoder is better at predicting the “no” (non-entailment) class significantly better than the “yes” (entailment) class. This behavior is consistent with the distribution of positive and negative samples in the training set. As the training set had more negative samples, it stands to reason the fine-tuned model has a slight bias towards the “no” (non-entailment) class. Furthermore, this also gives key insights into making use of various models with various capabilities in tandem for better and more stable performance.

Acknowledgments

This research was supported by Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea(RS-2023-00304286). This research was also supported by the University of Alberta, the Alberta Machine Intelligence Institute in Canada, the Canadian Natural Sciences and Engineering Research Council (NSERC) [including funding reference numbers RGPIN-2022-03469 and DGEER-2022-00369], and Alberta Innovates.

References

1. Achiam, Josh, e.a.: Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models (2023)
2. Anantha, R., Vakulenko, S., Tu, Z., Longpre, S., Pulman, S., Chappidi, S.: Open-domain question answering goes conversational via question rewriting. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 520–534 (2021)
3. Bach, S.H., Sanh, V., Yong, Z.X., Webson, A., Raffel, C., Nayak, N.V., Sharma, A., Kim, T., Bari, M.S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Ben-David, S., Xu, C., Chhablani, G., Wang, H., Fries, J.A., Al-shaibani, M.S., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Radev, D., Jiang, M.T.J., Rush, A.M.: Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models pp. 93–104 (2022)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., 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.M., 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., Dario, A.: Language models are few-shot learners. arXiv preprint arXiv:2104.14690 **33**, 1877–1901 (2022)
5. Bui, Q.M., Do, D.T., Le, N.K., Nguyen, D.H., Nguyen, K.V.H., Anh, T.P.N., Le, M.N.: Jnlp coliee-2023: Data argumentation and large language model for legal case retrieval and entailment. In: Proceedings of the Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE’2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL) (2023)
6. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V., Wei, J.: Scaling instruction-finetuned language models. In: arXiv preprint arXiv:2210.11416 (2022)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Ptr: Prompt tuning with rules for text classification **3**, 182–192 (2022)
9. Dolan, B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Third International Workshop on Paraphrasing (IWP2005) (2005)
10. Hill, F., Cho, K., Korhonen, A.: Learning distributed representations of sentences from unlabelled data. In: Proceedings of NAACL-HLT. pp. 528–540 (2016)
11. Jiang, T., Jiao, J., Huang, S., Zhang, Z., Wang, D., Zhuang, F., Wei, F., Huang, H., Deng, D., Zhang, Q.: Promptbert: Improving bert sentence embeddings with prompts. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 8826–8837 (2022)
12. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. In: arXiv preprint arXiv:1909.11942 (2020)
13. Li, X., Li, J.: On the sentence embeddings from pre-trained language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP. pp. 9119–9130 (2020)

14. Li, X., Li, J.: Angle-optimized text embeddings. In: arXiv preprint arXiv:2309.12871 (2023)
15. Nguyen, C., Nguyen, M.L.: Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks. In: Proceedings of the Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL) (2023)
16. Onaga, T., Fujita, M., Kano, Y.: Japanese legal bar problem solver focusing on person names. In: Proceedings of the Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL) (2023)
17. Pagliardini, M., Gupta, P., Jaggi, M.: Unsupervised learning of sentence embeddings using compositional n-gram features. In: Proceedings of NAACL-HLT. pp. 528–540 (2018)
18. 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 **21**(1), 5485–5551 (2020)
19. 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 (2019)
20. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the International Conference on Learning Representations (2019)
21. Wang, S., Scells, H., Koopman, B., Zuccon, G.: Can chatgpt write a good boolean query for systematic review literature search? In: arXiv preprint arXiv:2302.03495 (2023)
22. Wang, S., Fang, H., Khabsa, M., Mao, H., Ma, H.: Entailment as few-shot learner. In: arXiv preprint arXiv:2104.14690 (2022)
23. Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L., Si, L.: Structbert: Incorporating language structures into pre-training for deep language understanding. In: arXiv preprint arXiv:1908.04577 (2019)
24. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences
25. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. In: arXiv preprint arXiv:2302.11382 (2023)
26. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. 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. 1112–1122 (2018)
27. Williams, A., Nangia, N., Bowman, S.R.: Structbert: Incorporating language structures into pre-training for deep language understanding. In: Proceedings of NAACL-HLT. pp. 1112–1122 (2019)
28. Zeng, Z., Li, X., Gasevic, D., Chen, G.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 632–642 (2015)
29. Zeng, Z., Li, X., Gasevic, D., Chen, G.: Do deep neural nets display human-like attention in short answer scoring? In: Proceedings of the 2022 conference of the north American chapter of the Association for Computational Linguistics: Human language technologies. pp. 191–205 (2022)

30. Zhuo, W., Sun, Y., Wang, X., Zhu, L., Yang, Y.: Whitening-based contrastive learning of sentence embeddings. In: arXiv preprint arXiv:2305.17746 (2023)

Author Index

B	
Babiker, Housam	170
Billi, Marco	142
Breteler, Jeroen	115
C	
Chi, Hongmei	71
Contissa, Giuseppe	55
D	
Davila, Jacinto	55
F	
Fungwacharakorn, Wachara	43
G	
Goebel, Randy	170
H	
Huang, Sieh-Chuen	28
K	
Kim, Mi-Young	170
L	
Lee, Jieh-Sheng	1
Li, Wei-Jie	101
Liu, Chao-Lin	101
M	
May Myo Zin,	43
N	
Nakamura, Makoto	128
Nguyen, Duc-Minh	86
Nguyen, Ha-Thanh	14, 43, 86
Nguyen, Hai-Long	86
Nguyen, Tan-Minh	86
P	
Parenti, Alessandro	142
Pisano, Giuseppe	142
R	
Rabelo, Juliano	170
Rahman, Md Abed	170
S	
Sanchi, Marco	142
Sartor, Galileo	55
Satoh, Ken	14, 43, 86
Shao, Hsuan-Lei	28
Steging, Cor	154

T	
Takeda, Hideaki	43
Theophilus, Oluwafisayo	71
Tsai, Yun-Cheng	28
V	
Van Gessel, Thom	115
van Leeuwen, Ludi	154
Vuong, Thi-Hai-Yen	86
W	
Wu, Po-Hsien	101
Wyner, Adam	55
Y	
Yamada, Daichi	128

ISBN 978-4-915905-96-4