Proceedings of the Eigth International Competition on Legal Information Extraction/Entailment (COLIEE 2021)

in association with
the 18th International Conference
on Artificial Intelligence and Law

COLIEE 2021 Organizers

Randy Goebel, University of Alberta, Canada
Yoshinobu Kano, Shizuoka Univesity, Japan
Mi-Young Kim, University of Alberta, Canada
Juliano Rabelo, University of Alberta, Canada
Ken Satoh, National Institute of Informatics, Japan
Masaharu Yoshioka, Hokkaido University, Japan

June 21, 2021

Preface

This volume contains the papers accepted for presentation at COLIEE 2021, the Eighth Competition on Legal Information Extraction/Entailment, held on June 21, 2021 on line in conjunction with ICAIL 2021, the 18th International Conference on Artificial Intelligence and Law.

The aim of COLIEE is to formulate a challenging legal informatics competition that would engage researchers around the world, and build a community that would consider all manner of computer science and Artificial Intelligence methods to tackle the problems of legal reasoning.

This year there are 12 teams from 10 countries and each team participated in some of 5 COLIEE tasks. After sending competition results, we received 14 submissions from 10 teams to explain methods used. Each submission was reviewed by at least 3 program committee members. The committee decided to accept 3 winning papers among 5 tasks for the presentation at the ICAIL main conference and to accept 11 papers for the workshop. In addition, the organizers provided a summary paper for the COLIEE 2021 tasks and solutions. Papers for COLIEE are focused on an international legal information processing competition, which includes automatically understanding both statute law and case law.

We have had seven competitions, all of whose selected contributions have been peer-reviewed, published in the ICAIL proceedings or at the Springer Lecture Notes in Artificial Intelligence series, and – most importantly – developed a community of researchers, lawyers, judges, and related communities to discuss the impact and future of technology adoption in for legal systems.

The COLIEE organizers would like to acknowledge the continued support of people and organizations around the planet, including Colin Lachance from Compass Law/Vlex in Canada, who has been particularly support in his work to help develop and extend the case law data for COLIEE, and to Young Yik Rhim of Intellicon in Seoul, who has been our advocate since the beginning of COLIEE. In addition, a number of Japanese colleagues (in addition to the organizing team partciipants of Ken Satoh, Yoshinobu Kano, and Masaharu Yoshioka) have contributed to the extension and curation of the statute law data for the COLIEE competition.

June 21, 2021

Randy Goebel, University of Alberta, Canada
Yoshinobu Kano, Shizuoka Univesity, Japan
Mi-Young Kim, University of Alberta, Canada
Juliano Rabelo, University of Alberta, Canada
Ken Satoh, National Institute of Informatics, Japan
Masaharu Yoshioka, Hokkaido University, Japan
COLIEE 2021 organizers

Program Committee

Randy Goebel	University of Alberta
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
María Navas-Loro	Universidad Politécnica de Madrid
Juliano Rabelo	AMII
Julien Rossi	Amsterdam Business School
Ken Satoh	National Institute of Informatics and Sokendai
Jaromir Savelka	Carnegie Mellon University
Yunqiu Shao	Tsinghua University
Akira Shimazu	JAIST
Satoshi Tojo	JAIST
Vu Tran	Japan Advanced Institute of Science and Technology
Josef Valvoda	University of Cambridge
Sabine Wehnert	Otto-von-Guericke-Universität Magdeburg
Hannes Westermann	University of Montreal
Hiroaki Yamada	Tokyo Institute of Technology
Masaharu Yoshioka	Hokkaido University

Table of Contents

Summary of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021

Juliano Rabelo
Randy Goebel
rabelo@ualberta.ca
rgoebel@ualberta.ca
Alberta Machine Intelligence Institute, University of Alberta
Edmonton, Alberta, Canada

Yoshinobu Kano
kano@inf.shizuoka.ac.jp
Faculty of Informatics, Shizuoka University
Hamamatsu, Shizuoka, Japan

Mi-Young Kim
miyoung2@ualberta.ca
Dept. of Science, Augustana Faculty, University of Alberta
Camrose, Alberta, Canada

Masaharu Yoshioka
yoshioka@ist.hokudai.ac.jp
Faculty of Information Science and Technology, Hokkaido University
Sapporo-shi, Hokkaido, Japan

Ken Satoh
ksatoh@nii.ac.jp
National Institute of Informatics
Chiyoda-ku, Tokyo, Japan

ABSTRACT

This paper summarizes the 8th Competition on Legal Information Extraction and Entailment. In this edition, the competition included five tasks on case law and statute law. The case law component includes an information retrieval task (Task 1), and the confirmation of an entailment relation between an existing case and an unseen case (Task 2). The statute law component includes an information retrieval task (Task 3), an entailment/question answering task based on retrieved civil code statutes (Task 4) and an entailment/question answering task without retrieved civil code statutes (Task 5). Participation was open to any group based on any approach. Eight different teams participated in the case law competition tasks, most of them in more than one task. We received results from 6 teams for Task 1 (16 runs) and 6 teams for Task 2 (17 runs). On the statute law task, there were 8 different teams participating, most in more than one task. Six teams submitted a total of 18 runs for Task 3, 6 teams submitted a total of 18 runs for Task 4, and 4 teams submitted a total of 12 runs for task 5. We describe in this paper the approaches, our official evaluation, and analysis on our data and submission results.

CCS CONCEPTS

• **Information systems** → **Content analysis and feature selection**; **Similarity measures**; **Clustering and classification**; *Document topic models*; *Information extraction*; Specialized information retrieval.

KEYWORDS

legal textual entailment, legal information retrieval, text classification, imbalanced datasets

COLIEE '21, June, 2021, Sao Paulo, Brazil

ACM Reference Format:
Juliano Rabelo, Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, and Ken Satoh. 2021. Summary of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. In *COLIEE'21,June, 2021, Sao Paulo, Brazil*. ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

The objective of the Competition on Legal Information Extraction/Entailment (COLIEE) is to develop the state of the art for information retrieval and entailment using legal texts. It is usually co-located with JURISIN, the Juris-Informatics workshop series, which was created to promote community discussion on both fundamental and practical issues on legal information processing, with the intention to embrace various disciplines, including law, social sciences, information processing, logic and philosophy, including the existing conventional "AI and law" area. In alternate years, COLIEE is organized as a workshop the International Conference on AI and Law (ICAIL), which was the case in 2017 and 2019, and again in 2021. Until 2017, COLIEE consisted of two tasks: information retrieval (IR) and entailment using Japanese Statute Law (civil law). Since COLIEE 2018, IR and entailment tasks using Canadian case law were introduced, and the 2021 edition included a fifth task (entailment in statute law text without relying on previously retrieved data).

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

Task 5 is similar to Task 4, but competitors do not rely on previously retrieved statute data.

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

2 TASK 1 - CASE LAW RETRIEVAL

2.1 Task Definition

The Case Law Retrieval task consists in finding which cases should be "noticed". "Notice" is a legal technical term that denotes a legal case description that is considered to be relevant to a query case. More formally, given a set of cases C, a set of query cases Q, a set of the true noticed cases N, and a set of false noticed cases F, such as $C = \{Q \cup N \cup F\}$, the task is to find the set of answers $A = \{A_1 \cup A_2 ... \cup A_n\}$, such as $n = |Q|$ and each $A_i \subset N$ contains all the true noticed cases and only the true noticed cases with respect to the query case $q_i \in Q$.

2.2 Dataset

The dataset is comprised by 4,415 case law files. A labelled training set is given with 650 cases and a total of 3,311 true noticed cases. At first glance, the task may seem simple as one could think competitors need to identify the 3,311 cases among the 4,415 total cases. However, in reality the task requires competitors to identify the noticed cases for each given query case. In average, there are 5 noticed cases per query case, which should be identified among the 4,415 cases. A test set is given with 250 query cases. Initially, the golden labels for that test set is not provided to competitors. The test set has 900 true noticed cases.

2.3 Approaches

Li et al. [6] (team name: siat) propose a pipeline method based on statistics features and semantic understanding models, which enhances the retrieval method with both recall and semantic ranking.

Schilder et al. [13] (team name: TR) applies a two-phase approach for task 1: first, they generate a candidate set which tentatively contain all true noticed cases but eliminate some of the false candidates (i.e., it is optimized for recall). The second step is a binary classifier which receives as input the pair $(querycase, candidatecase)$ and predicts whether they represent a true noticed relationship.

Rosa et at. [12] (team name: NM) presents a vanilla application of BM25 to the case law retrieval problem. They do that by first indexing all base and candidate cases present in the dataset. Before indexing, each document is split into segments of texts using a context window of 10 sentences with overlapping strides of 5 sentences (the 'candidate case segments'). BM25 is then used to retrieve candidate case segments for each base case segment. The relevance score for a $(basecase, candidatecase)$ pair is the maximum score among all their base case segment and candidate case segment pairs. The candidates are then ranked according to threshold-based heuristics.

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

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

Table 1: Task 1 results

Team	File	F1
TLIR	run1.txt	0.1917
NM	NM_Run_task1_BM25.txt	0.0937
TLIR	run3.txt	0.0456
DSSIR	run_test_bm25.txt	0.0411
TLIR	run2.txt	0.0330
siat	siatEMB_result-task1.txt	0.0300
siat	siatEMB2_result-task1.txt	0.0291
DSSIR	run_test_vanillabert.txt	0.0279
DSSIR	run_test_bm25_dpr.txt	0.0272
MAN01	[MAN01] task1 run0.txt	0.0073
TR	TR_run1.csv	0.0046
JNLP	JNLP.taks1.BM25SD_3_7.txt	0.0019
JNLP	JNLP.taks1.BM25SD_7_3.txt	0.0019
JNLP	JNLP.taks1.SD.txt	0.0009
TR	TR_run2.csv	0.0000

2.4 Results and Discussion

Table 1 shows the results of all submissions received for task 1 in COLIEE 2021. A total of 15 submissions from 7 different teams have been received. It can be seen the f1-scores were lower than in previous editions, reflecting the fact the task is now more challenging than its previous formulation. Most of the teams applied traditional IR techniques such as BM25, transformer based methods such as BERT, or a combination of both. The best performing team was TLIR, with an f1-score of 0.1917. Also worth mentioning is the NM team, whose approach was a vanilla application of BM25 and got the second place.

3 TASK 2 - CASE LAW ENTAILMENT

3.1 Task Definition

Task 2 is a legal case entailment task and it involves the identification of a paragraph from existing cases that entails the decision of a new case. Given a decision Q of a new case and a relevant case R, a specific paragraph in R that entails the decision Q needs

to be identified. The organizers have confirmed that the answer paragraph cannot be identified merely by information retrieval techniques using some examples. Because the case R is a relevant case to Q, many paragraphs in R can be relevant to Q regardless of entailment. This task requires one to identify a paragraph which entails the decision of Q, so a specific entailment method is required that compares the meaning of each paragraph in R and the decision in Q in this task. The data is drawn from an existing collection of predominantly Federal Court of Canada case law. The evaluation measure will be precision, recall and F-measure.

In Task 2, the training and testing sets contain 326 and 100 base cases respectively. Training data consists of triples of a query, a noticed case, and a paragraph number of the noticed case by which the decision of the query is entailed. Here, 'noticed case' means the relevant case of the query. The example is shown in Table 2.

Table 2: Training data Example in Task 2

base case	B232 arrived in Canada with 491 other persons aboard the MV Sun Sea...
decision	Given that the Respondent remains a security risk whom the Minister has...
p#1 in noticed case	Previous decisions to detain the individual must be...
p#2 in noticed case	The Ministers are requesting an order...
...	...
p#32 in noticed case	THIS COURT ORDERS that the stay motion be granted until the final ...
entailing paragraph	#27

3.2 Approaches

Seven teams participated in Task 2, and total 17 results were submitted (average 2.43 results per team). Each team was allowed to submit maximum three results. Table 3 shows the approahes that teams used in Task 2. Althammer et al. [1] (team name:DSSIR) used either BM25 or DPR model as the first two results, trained on the entailing paragraph pairs in order to rank each paragraph in the noticed case given the query paragraph. They also combined the ranking of BM25 and DPR as the third result.

Schilder et al. [13] (team name: TR) used hand-crafted similarity features and applied a classical random forest classifier. Using n-gram vectors, universal sentence encoder vectors, and averaged word embedding vectors, they computed the similarity between each paragraph in the noticed case and the decision fragment in the query. after selecting most similar k paragraphs, they trained a random forest classifier.

Kim et al. [4] (team name: UA) used BERT pre-trained on a large (general purpose) dataset by fine-tuning on the provided training dataset. If the tokenization step produces more than the 512 token limit, they apply another transformer-based model to generate a summary of the input text, and then process the pair again. Since the input text often includes text in French, they apply a simple language detection model based on naive Bayesian filter to remove those fragments. Usually there are very few actual entailing paragraphs in a case (by far, most of the cases only have one entailing paragraph). So in the post-processing step they establish limits for the maximum number of outputs allowed per case. At the same time,they observe a minimum score in an attempt to reduce the number of the false positives.

Li et al. [6] (team name: siat) proposed a pre-training task on BERT (BERT-base-uncased) with dynamic N-gram masking, to get a special BERT model with legal knowledge (BERTLegal). They utilized n-gram masking to generate masked inputs for "masked language model" targets. The length of each n-gram mask is randomly selected amongst 1,2, and 3. They also did data augmentation and used Fast Gradient method.

Nguyen et al. [8] (team name: JNLP) used the supporting model and lexical model for two submissions, and in the last submission, they used NSFP model.

Table 3: Approaches in Task 2

Team	Approaches
DSSIR	BM25 or DPR model
TR	hand-crafted similarity features and random forest classifier
UA	BERT and naive bayesian filtering
siat	BERT, n-gram masking, data augmentation and Fast Gradient method
JNLP	supporting model, lexical model and NSFP model
NM	BM25, monoT5-zero-shot, and DeBERTa

[11] (team name: NM) used monoT5-zero-shot, monoT5 and DeBERTa. They also evaluated an ensemble of their monoT5 and DeBERTa models. The model monoT5-zero-shot is a sequence-to-sequence adaptation of the T5 model.

We were not able to identify the approach of the team MAN01 because of no paper submission from the team.

3.3 Results and Discussion

Table 4 shows the Task 2 results. NM team's three submissions are all ranked no.1 to no.3. Especially their Ensemble of DeBERTa and monoT5 showed the best performance with the F1 score of 0.6912. Most of the systems combined the traditional BM25 information retrieval algorithm and BERT Trans-former language model. They showed that the traditional BM25 system is still useful in legal information retrieval and entailment. To solve the issue of the dataset imbalance, some teams tried data augmentation. Also, there were some approaches to extract semantic relationships between paragraphs using BERT. In addition, there was an approach to use LEGAL-BERT, a BERT system optimized for the legal domain, but the performance was not promising.

4 TASK 3 - STATUTE LAW INFORMATION RETRIEVAL

4.1 Task Definition

Task 3 is a task to retrieve an appropriate subset (S_1, S_2,..., S_n) of Japanese Civil Code Articles from the Civil Code texts for answering a legal bar exam question statement Q.

Table 4: Task 2 official results

Team	File	F1
NM	Run_task2_DebertaT5.txt	0.6912
NM	Run_task2_monoT5.txt	0.6610
NM	Run_task2_Deberta.txt	0.6339
UA	UA_reg_pp.txt	0.6274
JNLP	JNLP.task2.BM25Sup._Den..txt	0.6116
JNLP	JNLP.task2.BM25Sup._Den._F..txt	0.6091
UA	UA_def_pp.txt	0.5875
JNLP	JNLP.task2.NFSP_BM25.txt	0.5868
siat	siatCLS_result-task2.txt	0.5860
DSSIR	run_test_bm25.txt	0.5806
siat	siatFGM_result-task2.txt	0.5670
UA	UA_loose_pp.txt	0.5603
TR	task2_TR.txt	0.5438
DSSIR	run_test_bm25_dpr.txt	0.5161
DSSIR	run_test_dpr.txt	0.5161
MAN01	[MAN01] task2 run1.txt	0.5069
MAN01	[MAN01] task2 run0.txt	0.2500

Table 5: Number of questions classified by number of relevant articles

number of relevant article(s)	1	2	4	total
number of questions	65	14	2	81

An appropriate subset means, the entailment system can judge whether the statement Q is true $Entails(S_1, S_2, ..., S_n, Q)$ or not $Entails(S_1, S_2, ..., S_n, notQ)$.

4.2 Dataset

For task 3, questions related to Japanese civil law were selected from the Japanese bar exam. Since there are update of Japanese Civil Code at April 2020, we revised text for reflecting this revision for Civil Code and its translation into English. However, since English translated version is not provided for a part of this code, we exclude these parts from the civil code text and questions related to these parts. As a results number of the articles used in the dataset is 768. Training data (the questions and relevant article pairs) was constructed by using previous COLIEE data (806 questions). In this data, questions related to revised articles are reexamined and ones for excluded articles are removed from the training data. For the test data, new questions selected from the 2020 bar exam are used (81 questions).

The number of questions classified by the number of relevant articles is listed in Table 5.

4.3 Approaches

The following 6 teams submitted their results (18 runs in total). All teams had experience in submitting results in the previous competition. Because the best performance system [14] of COLIEE 2020 uses BERT [2], most of the teams (HUKB, JNLP OvGU, and TR) uses BERT and ensemble results with ordinal IR system (HUKB and OvGU). One characteristic features proposed in this year's task is extension of training data for BERT-based IR system training. OvGU proposed a method to extend the contents of original article using text data related to the article (metadata, text from the website). JNLP proposed a method to select corresponding part of the article for the query using sliding window. HUKB proposed a method to add detailed information from the refereed articles. Other common techniques used in the system were well known IR engine mechanisms such as BM25, TF-IDF, Indri [15], and Word Movers' Distance (WMD) [5].

- **HUKB (three runs)** [18] uses BERT-based IR system and Indri for the IR module and compare the result of each system output to make final results. They constructs new article database with following two types; One is expanding the detailed information using refereed article and the other is splitting text for describing one judicial decision.
- **JNLP (three runs)** [8] uses BERT-based IR models that uses different BERT models for generating results and ensemble the outputs for the final results. They also construct training data of relevant article by selecting most relevant part of the article using sliding window.
- **LLNTU (three runs)** hasn' t submitted a paper describing the method used.
- **OvGU (three runs)** [16] uses sentence-BERT embedding [10] with TF-IDF by enriching the articles in the training data by using metadata, text from the web data related to the article and relevant queries from training data.
- **TR (three runs)** [13] uses Word Mover's Distance (WMD) (TR_HB) approach and BERT based on the spaCy large language model.
- **UA (three runs)** [4]1 uses BM25 (UA.BM25), TF-IDF(UA.tfidf) and language model (UA.LM) for IR module.

4.4 Results and Discussion

Table 6 shows the evaluation results of submitted runs. The official evaluation measures used in this task were macro average (average of evaluation measure values for each query over all queries) of F2 measure, precision , and recall. The terms "return", and "retrieved" represent the number of returned articles and the number of returned relevant articles, respectively.

$$precision = \frac{number\ of\ retrieved\ relevant\ articles}{number\ of\ returned\ articles} \quad (1)$$

$$recall = \frac{number\ of\ retrieved\ relevant\ articles}{number\ of\ relevant\ articles} \quad (2)$$

$$f2 = \frac{5 \times precision \times recall}{4 \times precision + recall} \quad (3)$$

We also calculate the mean average precision (MAP), recall at k (R_k: recall calculated by using the top k ranked documents as returned documents) by using the long ranking list (100 articles). Table 6 shows the results of the evaluation of submitted results.

This year, OvGU is the best run among all runs. JNLP achieves almost similar score and have higher MAP. This year, ordinal IR model BM25 achieves good performance for finding 1 relevant article for the question. From this results, we confirm effectiveness of using deep learning technology such as BERT is promising for this task.

Table 6: Evaluation results of submitted runs (Task 3)

sid	return	retrieved	F2	Precision	Recall	MAP	R_5	R_{10}	R_{30}
OvGU_run1	134	70	0.730	0.675	0.778	0.750	0.752	0.812	0.851
JNLP.CrossLMultiLThreshold	156	75	0.723	0.600	0.802	0.795	0.782	0.891	0.950
BM25.UA	81	61	0.709	0.753	0.704	0.756	0.713	0.733	0.812
JNLP.CrossLBertJP	132	72	0.709	0.624	0.772	0.778	0.822	0.842	0.901
R3.LLNTU	114	68	0.705	0.666	0.744	0.788	0.792	0.832	0.911
R2.LLNTU	104	67	0.704	0.677	0.731	0.789	0.782	0.842	0.911
R1.LLNTU	114	67	0.688	0.637	0.731	0.789	0.782	0.842	0.911
JNLP.CrossLBertJPC15030C15050	167	73	0.684	0.553	0.778	0.774	0.812	0.842	0.911
OvGU_run2	185	74	0.672	0.486	0.802	0.757	0.752	0.812	0.901
TFIDF.UA	81	55	0.657	0.679	0.654	0.731	0.723	0.743	0.812
LM.UA	81	46	0.546	0.568	0.543	0.642	0.644	0.683	0.812
TR_HB	162	54	0.523	0.333	0.617	0.662	0.713	0.743	0.842
HUKB-3	241	62	0.522	0.290	0.698	0.610	0.683	0.743	0.871
HUKB-1	251	57	0.473	0.240	0.654	0.613	0.663	0.752	0.871
TR_AV1	510	44	0.360	0.262	0.512	0.465	0.436	0.475	0.564
TR_AV2	564	48	0.337	0.149	0.556	0.435	0.396	0.446	0.495
HUKB-2	89	27	0.326	0.327	0.327	0.417	0.465	0.545	0.614
OvGU_run3	1379	66	0.302	0.157	0.701	0.556	0.574	0.614	0.703

Figure 1,2,3 shows average of evaluation measure for all submission runs. As we can see from Figure 1, there are many easy questions that almost all system can retrieve the relevant article. Easiest question is R01-12-U "An obligor may demand that a right of retention be extinguished by tendering reasonable security." whose relevant article is almost same as a question.

However, there are five queries for which none of the system can retrieve the relevant articles. All questions (R02-9-E, R02-15-I, 02-15-U, 02-15-E, and R02-23-E) are based on the use case of the article that requires semantic matching and handling anonymized symbol such as "A" and "B" for referring person. For example, question of R02-9-E is "B obtained A' s bicycle by fraud. In this case, A may demand the return of the bicycle against B by filing an action for recovery of possession." and related article is "Article 192 A person that commences the possession of movables peacefully and openly by a transactional act acquires the rights that are exercised with respect to the movables immediately if the person possesses it in good faith and without negligence." It is necessary to recognize following semantic relationship ("bicycle" as "movables" and "A" and "B" as person, and conflict between "by fraud" and "peacefully").

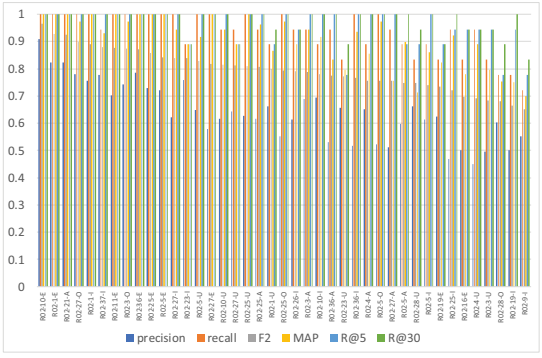

Figure 1: Averages of precision, recall, F2, MAP, R_5, and R_30 for easy questions with a single relevant article

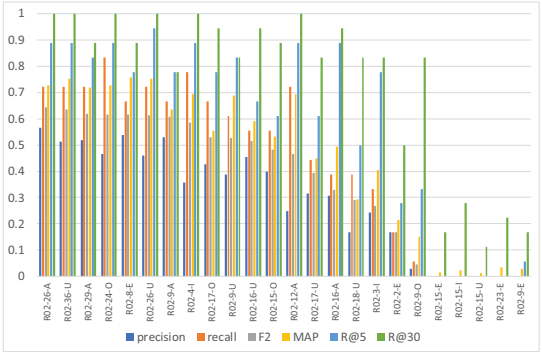

Figure 2: Averages of precision, recall, F2, MAP, R_5, and R_30 for non-easy questions with a single relevant article

5 TASK 4 AND 5 - STATUTE LAW TEXTUAL ENTAILMENT AND QUESTION ANSWERING

5.1 Task Definition

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

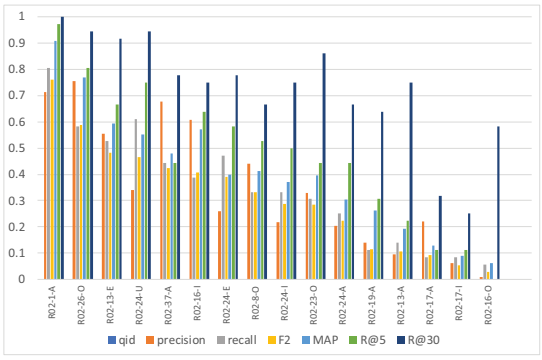

Figure 3: Averages of precision, recall, F2, MAP, R_5, R_10, and R_30 for non-easy questions with a single relevant article

style of task. In the Task 4 of COLIEE 2019 and 2020, we returned to the pure textual entailment task to attract more participants, allowing more focused analyses. In COLIEE 2021, we revived the question answering task as Task 5, retaining this textual entailment task as Task 4; Task 5 is a task to answer "yes" or "no" given a problem sentence(s) only. Participants can use any external data, however assuming that they do not use the test dataset.

5.2 Dataset

Our training dataset and test dataset are the same as Task 3. Questions related to Japanese civil law were selected from the Japanese bar exam. The organizers provided a data set used for previous campaigns as training data (806 questions) and new questions selected from the 2020 bar exam as test data (81 questions). Task 5 dataset is the same as Task 4. We performed Task 5 before Task 4 in order not to reveal the gold standard article labels which are included in the Task 4 dataset.

5.3 Approaches

We describe approaches for each team as follows, showing in a header format of **Team Name (number of submitted runs)**. All the teams submitted three runs for each of Task 4 and 5, except that the OvGU and HUKB teams participated Task 4 only.

- **HUKB (three runs)** [17] used an ensemble of BERT with data augmentation. They prepared an ensemble of 10 models. Their data augmentation extracts judical decision sentences, then makes positive/negative data from articles.
- **JNLP (three runs)** [9] uses *bert-base-japanese-whole-word-masking* with tf-idf based data augmentation. Their models are trained with different numbers of pretrain/fine-tune epochs (**JNLP.Enss5a** and **JNLP.Enss5b**), and an ensemble of these two models (**JNLP.EnssBest**) for Task 4, their proposed methods of Next Foreign Sentence Prediction (**JNLP.NFSP**) and Neighbor Multilingual Sentence Prediction (**JNLP.NMSP**) together with the original multilingual BERT (**JNLP.BERT_Multilingual**) for Task 5.
- **KIS (three runs)** [3] extended their previous works of a classic NLP approach to be explainable, based on predicate-argument structure analysis, original legal dictionary, negation detection, and ensemble of modules with different thresholds and combinations of these features.
- **OvGU (three runs)** [16] employed an ensemble of graph neural networks where each node represents either a query or an article, sentences embed by pre-trained *paraphrase-distilroberta-base-v1* (**OvGU_run1**), and LEGAL-BERT based on *legalbert- base-uncased* with different training phases (**OvGU_run2** and **OvGU_run3**).
- **TR (three runs)** [13] uses existing models of a T5-based ensemble **TR-Ensemble**, Multee **TR-MTE**, and Electra **TR_Electra** for Task4, distilled version of RoBERTa (**TRDistill-Roberta**), the largest model of GPT-3 **TRGPT3Davinci** and the smaller one **TRGPT3Ada** for Task 5.
- **UA (three runs)** [4]1 uses BERT (**UA_dl**), with semantic information (Kadokawa thesaurus concept number) (**UA_parser**).

5.4 Results and Discussion

Table 7 and Table 8 show evaluation results of Task 4 and 5, respectively. The test dataset characteristics seems not coherent throughout these years of the COLIEE series. For example, we observe more problems which require to handle anonymized symbol such as "A" and "B" for referring person (discussed in the Task 3 part as well) than previous years. Such a type of problems should be still very difficult for any NLP method to solve, except similar possible patterns could be sufficiently covered by some external training dataset. The best team in Task 4 would have solved "easier" problems well, while remaining "difficult" linguistic issues as our future works yet.

6 CONCLUSION

We summarized the systems and performances submitted to the COLIEE 2021 competition. For Task 1, TLIR was the best performing team with an F1 score of 0.1917, and applied a combination of LMIR and a BERT-based method. In Task 2, the winning team ensembled DeBERTa and monoT5 and achieved the F1 score of 0.6912. For Task 3, the top ranked team (OvGU) employed sentence-BERT embeddings and augmented the training data with metadata, web data related to the articles and relevant queries from the training data, achieving an F2 score of 0.73. HUKB was the Task 4 winner, with an Accuracy of 0.7037. They applied an ensemble of BERT models and data augmentation. In Task 5, JNLP was the best performing team and applied a variety of BERT-based models, achieving an Accuracy of 0.6049.

ACKNOWLEDGEMENTS

This competition would not be possible without the significant support of Colin Lachance from vLex and Compass Law, and the guidance of Jimoh Ovbiagele of Ross Intelligence and Young-Yik Rhim of Intellicon.

REFERENCES

[1] Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In *Proceedings of the COLIEE Workshop in ICAIL.*

Table 7: Evaluation results of submitted runs (Task 4). sid: submission id, L: Dataset Language (J: Japanese, E: English), #: number of correct answers (81 problems in total). JNLP.Enss5Ca and JNLP.Enss5Cb stand for JNLP.Enss5C15050 and JNLP.Enss5C15050SilverE2E10, respectively

Team	sid	L	Correct	Accuracy
N/A	BaseLine	N/A	Yes 43/All 81	0.5309
HUKB	HUKB-2	J	57	0.7037
HUKB	HUKB-1	J	55	0.6790
HUKB	HUKB-3	J	55	0.6790
UA	UA_parser	E	54	0.6667
JNLP	JNLP.Enss5Ca	J	51	0.6296
JNLP	JNLP.Enss5Cb	J	51	0.6296
JNLP	JNLP.EnssBest	J	51	0.6296
OVGU	OVGU_run3	E	48	0.5926
TR	TR-Ensemble	J	48	0.5926
TR	TR-MTE	J	48	0.5926
OVGU	OVGU_run2	E	45	0.5556
KIS	KIS1	J	44	0.5432
KIS	KIS3	J	44	0.5432
UA	UA_1st	E	44	0.5432
KIS	KIS2	E	43	0.5309
UA	UA_dl	E	43	0.5309
TR	TR_Electra	J	41	0.5062
OVGU	OVGU_run1	E	36	0.4444

Table 8: Evaluation results of submitted runs (Task 5). sid: submission id, L: Dataset Language (J: Japanese, E: English), #: number of correct answers (81 problems in total). JNLP.task5.BERT_Multilingual is abbreviated as JNLP.task5.BERT

Team	sid	L	Correct	Accuracy
N/A	BaseLine	N/A	No 43/All 81	0.5309
JNLP	JNLP.NFSP	J	49	0.6049
UA	UA_parser	E	46	0.5679
JNLP	JNLP.NMSP	J	45	0.5556
UA	UA_dl	E	45	0.5556
TR	TRDistillRoberta	J	44	0.5432
KIS	KIS_2	J	41	0.5062
KIS	KIS_3	J	41	0.5062
UA	UA_elmo	E	40	0.4938
JNLP	JNLP.task5.BERT	J	38	0.4691
KIS	KIS_1	J	35	0.4321
TR	TRGPT3Ada	J	35	0.4321
TR	TRGPT3Davinci	J	35	0.4321

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805

[3] Masaki Fujita, Naoki Kiyota, and Yoshinobu Kano. 2021. Predicate' s Argument Resolver and Entity Abstraction for Legal Question Answering: KIS teams at COLIEE 2021 shared task. In *Proceedings of the COLIEE Workshop in ICAIL.*

[4] Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2021. BM25 and Transformer-based Legal Information Extraction and Entailment. In *Proceedings of the COLIEE Workshop in ICAIL.*

[5] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings to Document Distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 957–966.

[6] Jieke Li, Xiaoyan Zhao, Junhao Liu, Jiabao Wen, and Min Yang. 2021. SIAT@COLIEE-2021: Combining Statistics Recall and Semantic Ranking for Legal Case Retrieval and Entailment. In *Proceedings of the COLIEE Workshop in ICAIL.*

[7] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving Legal Cases from a Large-scale Candidate Corpus. In *Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL).*

[8] Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. 2021. JNLP Team: Deep Learning Approaches for Legal Processing Tasks in COLIEE 2021. In *Proceedings of the COLIEE Workshop in ICAIL.*

[9] Ha-Thanh Nguyen, Vu Tran, Nguyen Le Minh, Minh-Phuong Nguyen, Thi-Hai-Yen Vuong, Minh Quan Bui, Minh-Chau Nguyen, Binh Dang, and Ken Satoh. 2021. ParaLaw Nets - Cross-lingual Sentence-level Pretraining for Legal Text Processing. In *Proceedings of the COLIEE Workshop in ICAIL.*

[10] Nils Reimers and Iryna Gurevych. 2019. 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)*. 3973–3983.

[11] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. To Tune or Not To Tune? Zero-shot Models for Legal Case Entailment. In *Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL).*

[12] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. In *Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL).*

[13] Frank Schilder, Dhivya Chinnappa, Kanika Madan, Jinane Harmouche, Andrew Vold, Hiroko Bretz, and John Hudzina. 2021. A Pentapus Grapples with Legal Reasoning. In *Proceedings of the COLIEE Workshop in ICAIL.*

[14] Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. 2020. BERT-based Ensemble Model for The Statute Law Retrieval and Legal Information Entailment. In *The Proceedings of the 14th International Workshop on Juris-Informatics (JURISIN2020).* The Japanese Society of Artificial Intelligence„ 223–234.

[15] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis.* 2–6.

[16] Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W. De Luca. 2021. Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization. In *Proceedings of the 18th International conference on Artificial Intelligence and Law (ICAIL).*

[17] Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. 2021. BERT-based Ensemble Methods with Data Augmentation for Legal Textual Entailment in COLIEE Statute Law Task. In *Proceedings of the COLIEE Workshop in ICAIL.*

[18] Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2021. BERT-based Ensemble Methods for Information Retrieval and Legal Textual Entailment in COLIEE Statute Law Task. In *Proceedings of the COLIEE Workshop in ICAIL.*

DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval

Sophia Althammer*
TU Vienna
Vienna, Austria
sophia.althammer@tuwien.ac.at

Arian Askari*
Leiden University
Leiden, Netherlands
a.askari@liacs.leidenuniv.nl

Suzan Verberne
Leiden University
Leiden, Netherlands
s.verberne@liacs.leidenuniv.nl

Allan Hanbury
TU Vienna
Vienna, Austria
allan.hanbury@tuwien.ac.at

ABSTRACT

In this paper, we present our approaches for the case law retrieval and the legal case entailment task in the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. As first stage retrieval methods combined with neural re-ranking methods using contextualized language models like BERT [5] achieved great performance improvements for information retrieval in the web and news domain, we evaluate these methods for the legal domain. A distinct characteristic of legal case retrieval is that the query case and case description in the corpus tend to be long documents and therefore exceed the input length of BERT. We address this challenge by combining lexical and dense retrieval methods on the paragraph-level of the cases for the first stage retrieval. Here we demonstrate that the retrieval on the paragraph-level outperforms the retrieval on the document-level. Furthermore the experiments suggest that dense retrieval methods outperform lexical retrieval. For re-ranking we address the problem of long documents by summarizing the cases and fine-tuning a BERT-based re-ranker with the summaries. Overall, our best results were obtained with a combination of BM25 and dense passage retrieval using domain-specific embeddings.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; **Rank aggregation**; **Language models**.

KEYWORDS

neural IR, dense retrieval, BERT, summarization

*Both authors contributed equally to this research.

COLIEE 2021, June 21, 2021, Online

ACM Reference Format:
Sophia Althammer, Arian Askari, Suzan Verberne, and Allan Hanbury. 2021. DoSSIER@COLIEE 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

In case law systems, precedent cases are a primary legal resource for the decision of a new given case. Therefore it is a part of a lawyer's or paralegal's daily work to find precedent cases which support or contradict a new case [24]. With the exponentially growing amount of electronically stored information in the legal domain [25], it costs legal professionals increasingly more effort to retrieve the cases which are relevant to their case. In order to find evidence lawyers require their search systems to find all cases which are relevant [1], but at the same time, legal researchers will examine up to 50 results in practice [19] and therefore require a precision-oriented solution. We tackle these requirements of legal case retrieval in Task 1 of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021 by first retrieving candidates from the whole corpus with the aim of a high recall and then re-ranking these candidates for precision at high ranks. In Task 2 of COLIEE 2021 it is the task to identify a paragraph of an existing case that entails the decision of a new case [14] and we approach this problem with re-ranking as well.

In news and web search, contextualized language models like BERT [5] brought substantial effectiveness gains to the first stage retrieval [6, 8–10, 16, 27] as well as to the re-ranking stage [2, 7, 12, 13]. We aim to transfer these advantages also to the task of case law retrieval, however this task has specific challenges compared to retrieval tasks in the web and news domains: documents are written in domain specific language [24], there is a specific notion of relevance not only on document- but also on paragraph-level [25] and the documents tend to be long [21, 25]. For example in the case law corpus from COLIEE 2021 the documents contain on average 1274.62 words,with minimum 1 word and maximum 76,818 words. This is not only the case for the candidate documents in the retrieval collection, but also for the query cases.

As the input length of BERT is limited, we propose two different approaches for handling the longer documents with BERT, one for first stage retrieval and one for re-ranking. In order to achieve a

high recall for the first stage retrieval, we reason that prior cases are not only relevant on a document-level to a query case, but that a prior case can be relevant to a query case only based on a single paragraph which is relevant to another paragraph of the query case [20, 21, 26]. Therefore we propose for the first stage retrieval to split up the query case and cases in the corpus into their paragraphs and retrieve for each paragraph of the query case relevant prior cases based on the relevance of their paragraphs. For the document-level and paragraph-level retrieval, we evaluate both lexical retrieval model and of a semantic retrieval model. For the lexical retrieval model we choose BM25 [17] and for the dense passage retrieval model we choose DPR [9]. We train the dense passage retrieval model on the legal entailing paragraph pairs as we suggest that the relevance of the paragraphs to each other is crucial for the relevance on the document-level. We denote the trained DPR model with lawDPR. When comparing the paragraph-level and document-level retrieval, we demonstrate that the paragraph-level retrieval achieves a higher recall for BM25 as well as for lawDPR and that lawDPR outperforms BM25 in terms of retrieval recall. We also show that the results of the lexical and semantic retrieval of BM25 and lawDPR complement each other and that the score combination of both retrieval models leads to superior results.

For re-ranking the retrieved results we experiment with the approach of summarizing the query cases and the cases in the corpus as Rossi and Kanoulas [18], Tran et al. [23] have shown that query document summarization is valuable for case law retrieval. For this, we fine-tune Longformer Encoder-Decoder (LED) as a state-of-the-art abstractive summarization model [3] and use this for summarizing the cases. For each query and document case, we summarize the text and interpret the re-ranking as a binary classification problem and fine-tune BERT on predicting whether the summarized query case is relevant to a summarized case in the corpus or not. For Task 1 we submit 3 runs: one based on the ranking with BM25, one with the combination of BM25 and lawDPR and one with the first stage retrieval of BM25 and lawDPR and with re-ranking with BERT.

For the legal entailment task (Task 2) we evaluate BM25 [17] and the DPR model [9] trained on the entailing paragraph pairs ('lawDPR'). Here we find that the combination of the BM25 and DPR scores also improves the overall performance for identifying the entailing paragraph. For Task 2 we submit 3 runs: one based on the ranking on BM25, one based on the ranking of lawDPR and one with the combination of BM25 and lawDPR.

We make the source code available at:
https://github.com/sophiaalthammer/dossier_coliee.

2 TASK DESCRIPTION

2.1 Task 1: The Legal Case Retrieval Task

The aim of legal case retrieval is to design systems to automatically identify the supporting cases of a given query case, which should be noticed for solving the query case [14]. The task consists of reading a new case Q and selecting supporting cases $S_1, S_2, \ldots, S_n$ ("noticed cases") from the whole case law collection for the decision of Q.

Table 1: Statistics of the training and test collection for Task1 and Task2

	Task 1		Task 2	
	Train	Test	Train	Test
# of queries	650	250	425	100
avg # of candidates	4415	4415	32.12	32.18
avg # relevant candidates	5.17	3.6	1.17	1.17
avg query length (words)	690.56	1817.12	38.26	37.41
avg candidate length (words)	1274.62	1274.62	102.67	117.91

2.2 Task 2: The Legal Case Entailment Task

In the legal case entailment task, the goal is to design a system that finds paragraphs in a relevant case that entail the decision of a given new case [14]. In the Task 2 of COLIEE there is a query paragraph given as well as the paragraphs of one legal case as candidates and the task is to find the paragraphs from the legal case which entail the decision of the query paragraph.

2.3 Training and test collection

The training and test collections for Task 1 and Task 2 contain cases from the Federal Court of Canada case laws. For Task 1, a corpus with 4415 legal cases is given from which relevant cases should be retrieved for each query case. The cases can contain as addition to the English version also a French version. For Task 2 there are for each query paragraph the paragraphs of one case as candidates given and one needs to identify the entailing paragraphs. The statistics of the collections are in Table 1 including the number of queries, the average number of relevant documents and the average length of the queries and candidates.

3 METHODS

3.1 Task 1

3.1.1 First stage retrieval. For the lexical retrieval in the first stage retrieval we use BM25 [17]. When querying the index with query q, the BM25 model assigns each document d in the index a ranking score $s_{BM25}(d, q)$, the higher the score, the higher the relevance of the query and the indexed document. For semantic retrieval, we use a dense passage retrieval model [9] based on two Siamese BERT Encoders, one encodes the query passage, the other one the candidate passage. The encoder encodes the query and candidate passage into a vector and the dot-product between those two vectors denotes the relevance score $s_{lawDPR}(p, q)$ between the query q and the candidate passage p. This dense retrieval model (lawDPR) is trained on the entailing query-paragraph pairs of Task 2, in order to align the vector representations of entailing paragraphs.

As the maximum input length for the dense passage retrieval model [9] is limited, we employ this model on the full text of the case by splitting up the whole document d into its paragraphs $p_1, .., p_{m_d}$ and indexing each paragraph separately. When querying the paragraph-level index, we also split up the query case q into its paragraphs $q_1, .., q_{n_q}$ and retrieve for each query paragraph q_i with $i \in [n_q]$ the ranked list of paragraphs $r_1, .., r_{n_q}$. The paragraph in the ranked lists r_i with $i \in [n_q]$ are reduced to the paragraphs documents and

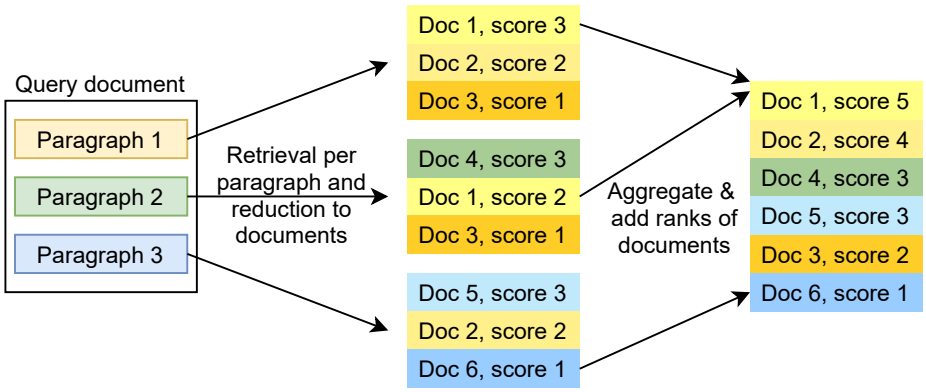

Figure 1: Aggregation of paragraph-level retrieval to an overall ranked list for the query document

then the ranked lists of documents are aggregated to one ranked list of documents for the whole query case. This aggregation will be further investigated in the next section. Equivalently we also use this approach for retrieval with BM25 on the paragraph-level and refer to this approach to **paragraph-level retrieval**.
In order to analyze the impact of the paragraph-level retrieval compared to document-level retrieval on the retrieval effectiveness, we also conduct **document-level retrieval** with BM25 and lawDPR. Here all documents in the corpus are indexed based on their whole text and the query case is also encoded with all its text. As the dense passage retrieval model has a limited input length and therefore can only retrieve passages up to 512 tokens, the document-level retrieval for lawDPR is the retrieval based on the first 512 tokens of the query case and the candidate case.

Aggregation. For the retrieval on the paragraph-level we retrieve for each paragraph of the query document $q_1, .., q_{n_q}$ a ranked list $r_1, .., r_{n_q}$ of top N results. These ranked lists r_i with $i \in [n_q]$ of paragraphs must be aggregated to one overall ranked list for the whole document, this process of aggregation is visualized in Figure 1. For the aggregation we reduce the paragraphs per ranked list r_i to the paragraphs' document. As the paragraphs are reduced to their documents it is possible that one document occurs multiple times in the list. Within this reduction from the paragraph p to its document d the scores of the retrieval model $s(p, q_i)$ of each paragraph p in the ranked list r_i are replaced with an integer scores of the document d $s(d, q_i)$ according to their position: The first document gets the score N, per position in the list the score decreases by 1 so that the last document gets the score 1. Then we aggregate the lists r_i per query paragraph q_i to one ranked list for the whole query document by adding the integer scores for each retrieved document and ranking by the new additive scores. The new additive score of a document d for a query q is then:

$$s(d, q) = \sum_{i=1}^{n_q} \sum_{d \in r_i} s(d, q_i)$$

We also experiment with other aggregation strategies using the similarity scores of the retrieval model or using an interleaving approach, but find that the aggregation with the additive integer scores $s(d, q)$ for a document d and query q lead to the overall best performance.

Combination. We find that 67% of the relevant cases of Top1000 which are retrieved from lawDPR on paragraph-level and 70% of the relevant cases of Top1000 which are retrieved from BM25 on paragraph-level are the same documents. Therefore 23% of the retrieved relevant cases from lawDPR and 30% of the retrieved relevant cases from BM25 are different, hence BM25 and lawDPR retrieve different relevant documents. For the first stage retrieval we combine for each document d in the index the relevance scores of BM25 $s_{BM25}(d, q)$ and the relevance score for the whole document d of lawDPR $s_{lawDPR}(d, q)$ from paragraph-level retrieval with BM25 and lawDPR. In order to get an overall score of a document d and a query q we add the BM25 and the lawDPR relevance scores with a scalar weighting $\alpha, \beta \in \mathbb{R}$

$$s_{BM25+lawDPR}(d, q) = \alpha\, s_{BM25}(d, q) + \beta\, s_{lawDPR}(d, q).$$

We denote the combination with BM25+lawDPR.

3.1.2 *Re-ranking.*

Summarizer: LED. The current state of the art in abstractive summarization is Transformer models [11, 15]. However, the input of pre-trained available models of these architectures is limited to 1024 tokens, and the majority of case law documents in our collection is longer than that. Beltagy et al. [3] proposed Longformer-Encoder-Decoder (LED), which is a Transformer variant that supports much longer inputs. For this competition, we evaluate the effectiveness of LED for case law retrieval.

BERT. For re-ranking we use BERT and fine-tune the pre-trained BERT model (BERT-Base, uncased) with a linear combination layer stacked atop the classifier [CLS] token on binary classification if a query summmary and a document summary are relevant to each other or not. We represent the query as sentence A and the document as sentence B in the BERT input:

"[*CLS*] *query document* [*SEP*] *candidate document* [*SEP*]"

Cut-off value. As the task is to retrieve the relevant cases to a given query q, we consider the top-k-ranked documents d in the

ranked list as relevant and denote k as cut-off value. We evaluate the best cut-off value k depending on the F1-score of the validation set.

3.2 Task 2

For identifying the entailing paragraphs p to a given query paragraph q we use lexical and semantic ranking approaches in order to rank the given candidate paragraphs p. In order to predict which paragraphs of the ranking are entailing the query paragraph, we consider the top-k-ranked paragraphs, where we denote k as cut-off value. We evaluate the best cut-off value k depending on the F1-score on the validation set.
In order to rank each paragraph given the query paragraph, we create per query paragraph an index containing the given candidate paragraphs. Then we query this index with the query paragraph and obtain the ranking for the candidate paragraphs. We do this procedure either using BM25 or lawDPR in order to compare both models for the legal entailment task.

BM25. For the lexical ranking we use BM25 [17].

lawDPR. For the semantic retrieval we use the same dense retrieval model [9] as in Task 1, which is trained on the entailing query-paragraph pairs of the training collection for Task 2. As Task 2 is on paragraph-level we can directly use the dense passage retrieval model without aggregating the results as in Task 1.

BM25+lawDPR. We also combine the ranking of BM25 and lawDPR with the same method as for Task 1.

4 EXPERIMENT DESIGN

4.1 Data pre-processing

For the experiments we divide the training collections of Task 1 and Task 2 into a training and validation set. The validation sets for Task 1 and for Task 2 contain the last 100 query cases of the training sets, respectively.
For the data pre-processing for Task 1, we remove the French versions of the cases when reading in the cases. We divide the cases into an introductory part, a summary, if they contain one, and their paragraphs. As we want to distinguish the cases by their text, we remove text parts of the cases which appear exactly the same in multiple cases, as they do not add any information which is particular for one case. Therefore we remove introductory parts and summaries, which appear exactly the same in more than 100 cases from the case text as we consider them as non-informative. The introductory parts have an average length of 73.66 words, the summaries have on average 227.05 words and the paragraphs have a average length of 92.80 words. We use the same data pre-processing for all submitted runs.
We pre-process the data of Task 2 by reading in the paragraphs, splitting the words at whitespaces and removing tabs. Here we also use the same data pre-processing for all submitted runs.

4.2 Task 1

4.2.1 *First stage retrieval.*

BM25. We use the BM25 implementation from ElasticSearch[1] with default parameter values $k = 1.2$ and $b = 0.75$.

lawDPR. We train the dense retrieval model with two BERT-based-uncased Siamese Encoders on the training set and validation set of Task 2 in the same fashion as in [9]. Here a pair of query and candidate paragraph which are relevant to each other are considered as positive sample and a pair of query and candidate paragraph which are not relevant to each other are considered as negative sample. Different to Karpukhin et al. [9] we sample the negative candidate paragraphs to a query paragraph randomly from the paragraphs which are not denoted as relevant to this query paragraph. We train the model for 40 epochs with a pairwise loss for the first 30 epochs and a list-based loss for the last 10 epochs, in order to include the pairwise and listwise relation of the samples in the training. We use batch size of 22, a maximum sequence length of 256 and a learning rate of $2 * 10^{-5}$.

We index the corpus on the paragraph- and on the document-level using BM25 and lawDPR and then query each index with the query cases. For the paragraph-level retrieval the query is split up into its paragraphs, for each query paragraph a ranked list of paragraphs is retrieved and aggregated to a ranked list of documents as described in section 3. For the first stage retrieval on the paragraph-level the introductory part and the summary are also treated as paragraph of the document. For the document-level retrieval the documents are indexed based on their whole text and for each query case the whole text is taken into account.

BM25+lawDPR. We experiment with combining the scores of BM25 and lawDPR as described in section 3. We take the Top1000 aggregated lists for the paragraph-level retrieval of BM25 and lawDPR and use the weights $[\alpha, \beta] \in [1, 1], [2, 1], [3, 1], [4, 1]$.

4.2.2 *Re-ranking.*

Summarizer: LED. For LED, we used the Huggingface transformers library[2] and set the local attention window size to 512 tokens. To limit memory use, we use gradient checkpointing and set the input size in training to 8192 tokens which covers more than 86% of COLIEE'18 Task 1 documents completely (the longer documents are truncated at 8912 tokens). We set the maximum length to generate a summary for an unseen document as 10% of the length of the original text. We use the COLIEE'18 Task 1 data, which contains human reference summaries, for evaluation of the summarizers[3].

In COLIEE'18, the summaries are provided for all queries and more than 80% of document cases. We extracted summaries like [22], and after removing duplicates, 6, 257 unique documents are left for which a summary is available. Finally, we fine-tuned the LED model on the unique documents that contains summary. We kept the other hyperparameters (optimizer, dropout, weight decay) identical to [3] and set the global attention on the first <s> token. We use the fine-tuned LED on COLIEE'18 to generate summary for query and

[1] https://github.com/elastic/elasticsearch
[2] https://huggingface.co/transformers/
[3] As opposed to COLIEE'19 and COLIEE'20, COLIEE'18 contains expert-written summaries.

Table 2: Task 1: Recall@N for the first stage retrieval of BM25 and lawDPR on the paragraph-level and the document-level retrieval on validation set

Model	Retrieval level	R@100	R@200	R@300	R@500
BM25	document-level	0.5519	0.6525	0.6938	0.7715
BM25	paragraph-level	0.5303	0.6765	0.7469	0.8164
lawDPR	document-level	0.1172	0.1381	0.1651	0.2466
lawDPR	paragraph-level	0.4776	0.6111	0.6757	0.7735
BM25+lawDPR	paragraph-level	**0.5639**	**0.6932**	**0.7577**	**0.8193**

Table 3: Task 1 validation set evaluation

Run	Precision	Recall	F1-Score
BM25 (cutoff at 7)	0.0948	**0.0471**	0.0629
BM25+lawDPR (cutoff at 7)	**0.0987**	**0.0471**	**0.0638**
BERT re-ranking (cutoff at 7)	0.0296	0.0157	0.0205

Table 4: Task 1 test set evaluation

Run	Precision	Recall	F1-Score
BM25 (cutoff at 7)	**0.0777**	**0.1959**	**0.1113**
BM25+lawDPR (cutoff at 7)	0.0737	0.1788	0.1044
BERT re-ranking (cutoff at 7)	0.0211	0.0546	0.0304

document cases in COLIEE'21, and provide the summaries as the representation of cases in BERT-based re-ranking step.

BERT-based re-ranking. We truncate the summarized documents such that the concatenated query document (truncated at 100 words), candidate document, and the separator tokens do not exceed 512 tokens. We re-rank the top-500 BM25+lawDPR results. We find $k = 7$ as the optimal cut-off of the ranking for the selection of documents. We train each model for 100 epochs, each with 32 batches of 16 training pairs, with the initial learning rate of $3 * 10^{-5}$, followed by a power 3 polynomial decay.

For Task 1 we submit 3 runs: one based on the paragraph-level retrieval and ranking of BM25, the second based on the combination of paragraph-level retrieval of BM25 and lawDPR and the third takes the retrieved candidates from the combination of BM25 and lawDPR and re-ranks them using the summarizer and BERT.

4.3 Task 2

BM25. We use the BM25 implementation from ElasticSearch with default parameters and create an index for each query paragraph and its candidates.

lawDPR. We take the trained lawDPR model from Task 1.

BM25+lawDPR. We combine the scores of BM25 and lawDPR as described in scetion 3 and use the weights $[\alpha, \beta] \in [1, 1], [2, 1], [3, 1], [4, 1]$. For Task 2 we submit 3 runs: one based on BM25, the second based on lawDPR and the third with the combination of BM25 and lawDPR.

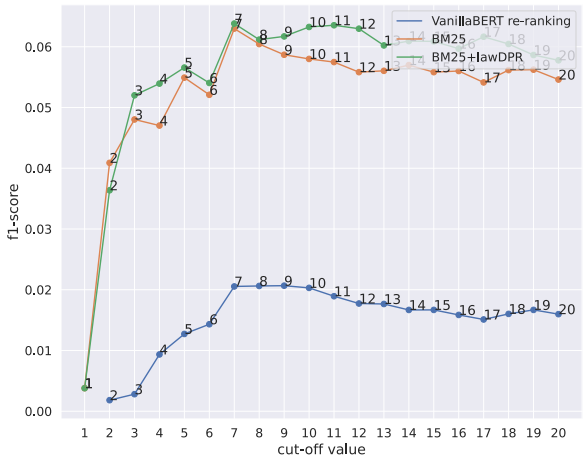

Figure 2: F1-score for Task 1 on the validation set for different cut-off values of our runs

5 RESULTS AND ANALYSIS

5.1 Task 1

5.1.1 First stage retrieval. We compare the first stage retrieval of BM25 and lawDPR on the paragraph- and the document-level and also evaluate the combination of the scores of BM25 and lawDPR. In the first stage retrieval it is our goal to achieve a high recall, therefore we evaluate the recall@100, recall@200, recall@300 and recall@500 using pytrec_eval[4] on the validation set. The results are shown in Table 2. As described in section 4 we combine the scores of BM25 and lawDPR and obtain a list ranked by the combination of the BM25 and lawDPR scores. When evaluating the retrieval on the validation set, we find that the combination with the weights [3, 1] leads to the best results, therefore we only present the results for this weighting and denote it as BM25+lawDPR.

Comparing the retrieval on paragraph- and document-level we see that the retrieval on paragraph-level outperforms the retrieval on whole document-level for BM25 as well as for lawDPR. This shows that our approach of tackling the long documents for a contextualized language model with limited input length is not only beneficial for the dense passage retrieval model, but also for the lexical retrieval with BM25.
Furthermore we see that BM25 outperforms the first stage retrieval on paragraph-level of lawDPR by $4 - 6\%$ in terms of recall@N. However when combining the scores of BM25 and lawDPR, we see that the overall recall is the best at all evaluated cut-off values.

5.1.2 Ranking. We evaluate the precision, recall and F1-scores for the runs we submitted at the cut-off value of 7 for each run as we find in out analysis in Figure 2 that the best cut-off value in terms of F1-score is 7 for all three runs. The evaluation results on the validation set are in Table 3 and on the test set in Table 4.
Here we see that the overall ranking performance is improved for

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

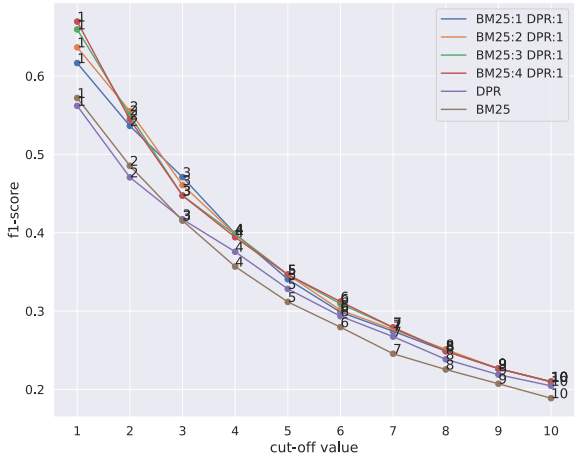

Figure 3: F1-score of task 2 on the validation set for different cut-off values of our runs, BM25:α DPR: β denotes the weighting factors of the BM25+lawDPR run

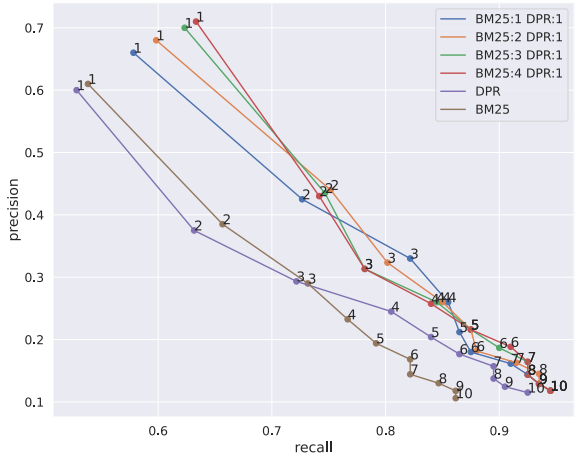

Figure 4: Precision and recall of Task 2 on the validation set for different cut-off values of our runs, BM25:α DPR: β denotes the weighting factors of the BM25+lawDPR run

Table 5: Task 2 validation set evaluation

Run	Precision	Recall	F1-Score
BM25 (cutoff at 1)	0.5583	0.6300	0.5719
lawDPR (cutoff at 1)	0.5283	0.6000	0.5618
BM25+lawDPR (cutoff at 1)	**0.6333**	**0.7100**	**0.6694**

Table 6: Task 2 test set evaluation

Run	Precision	Recall	F1-Score
BM25 (cutoff at 1)	0.6300	0.5953	0.6121
lawDPR (cutoff at 1)	0.5600	0.5203	0.5394
BM25+lawDPR (cutoff at 1)	**0.7100**	**0.6753**	**0.6922**

the validation set of BM25+lawDPR, this approach also outperforms the re-ranking based on the summaries. Contrary to that we find that on the test set BM25 achieves the best performance in terms of precision, recall and F1-score.

There is a difference in the evaluation scores between our evaluation and the evaluation of the task coordinator. We use for our evaluation the off-the-shelf pytrec_eval[5] library and were in contact with the task organizer, however we could not clarify the differences between the evaluations.

5.2 Task 2

For Task 2 we evaluated precision, recall and F1-score on the evaluation set for multiple cut-off values. The overall F1-score for multiple cut-off values is visualized in Figure 3, the performance for the precision and recall of various cut-off values is visualized in Figure 4. Figure 4 shows how the precision decreases with an increasing cut-off value and therefore an increasing recall. Here we also see that BM25 and lawDPR have a similar performance, with lawDPR having a better performance with a higher cut-off value than BM25. We also clearly see the gap between BM25 and lawDPR and the combination of the scores of BM25 and lawDPR. We see a similar picture in Figure 3, here the F1-score is continuously decreasing with a higher cut-off value, therefore we select a cut-off value of 1 for each run. We also see that the overall performance is the best with the weight [4, 1] therefore we use these weights for combining BM25 and lawDPR.

In Table 5 we can see the precision, recall and F1-score on the validation set for BM25, lawDPR and BM25+lawDPR. Here we see that BM25 has a better performance than lawDPR, but that both runs are outperformed by the combination of BM25 and lawDPR. This also suggests that combining the strengths of lexical and semantic retrieval models is beneficial for legal paragraph entailment and that BM25 and lawDPR complement each other.

We also evaluate our three final runs on the test set, the evaluation results can be found in Table 6. Here we find the same relations between the evaluation results as for the validation set evaluation.

6 CONCLUSION AND FUTURE WORK

Our participation at COLIEE 2021 in Task 1 and Task 2 gave the opportunity to explore information retrieval challenges in the legal case law retrieval and legal entailment. Our goal is to combine traditional lexical retrieval models with dense passage retrieval models as well as use contextualized re-ranking models for re-ranking the results. For legal case retrieval we identify the challenge of long documents for dense retrieval and neural re-ranking strategies. Therefore we present, compare and evaluate two approaches for handling the long documents:

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

- Splitting up the documents into their paragraphs and proposing a paragraph-level retrieval for first stage retrieval and
- Generating summaries for the neural re-ranking with BERT.

We show that the paragraph-level retrieval in the first stage outperforms the document-level retrieval and that the combination of lexical and semantic retrieval models leads to the best results. Also for the ranking we find that the combination of lexical and semantic models improves the overall effectiveness of the ranking. Our experiments with re-ranking based on the summaries of the cases seems not to improve the overall ranking effectiveness. Furthermore we also find that the combination of lexical and semantic retrieval methods improves the overall performance also for the passage entailment task. This leads to the conclusion that lexical and semantic retrieval methods have different strengths and complement each other. In future we plan to investigate dense retrieval models with domain specific contextualized language models like LegalBERT [4] and with different paragraph aggregation approaches.

ACKNOWLEDGMENTS

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

REFERENCES

[1] Khalid Al-Kofahi, Alex Tyrrell, Arun Vachher, and Peter Jackson. 2001. A Machine Learning Approach to Prior Case Retrieval. *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, 88–93.

[2] Sophia Althammer, Sebastian Hofstätter, and Allan Hanbury. 2021. Cross-domain Retrieval in the Legal and Patent Domains: a Reproducibility Study. In *Advances in Information Retrieval, 43rd European Conference on IR Research, ECIR 2021*.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[4] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2020. Complementing Lexical Retrieval with Semantic Residual Embedding. (4 2020). http://arxiv.org/abs/2004.13969

[7] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2021. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. arXiv:2010.02666 [cs.IR]

[8] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. arXiv:2104.06967 [cs.IR]

[9] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[10] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6086–6096. https://doi.org/10.18653/v1/P19-1612

[11] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[12] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104.

[13] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[14] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. A Summary of the COLIEE 2019 Competition. In *New Frontiers in Artificial Intelligence*, Maki Sakamoto, Naoaki Okazaki, Koji Mineshima, and Ken Satoh (Eds.). Springer International Publishing, Cham, 34–49.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[16] Nils Reimers and Iryna Gurevych. 2019. 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)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[17] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[18] Julien Rossi and Evangelos Kanoulas. 2019. Legal search in case law and statute law. *Frontiers in Artificial Intelligence and Applications* 322, 83–92. https://doi.org/10.3233/FAIA190309

[19] Tony Russell-Rose, Jon Chamberlain, and Leif Azzopardi. 2018. Information retrieval in the workplace: A comparison of professional search practices. *Information Processing and Management* 54 (11 2018), 1042–1057. Issue 6. https://doi.org/10.1016/j.ipm.2018.07.003

[20] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. THUIR@COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment. *CoRR* abs/2012.13102 (2020). arXiv:2012.13102 https://arxiv.org/abs/2012.13102

[21] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3501–3507. https://doi.org/10.24963/ijcai.2020/484 Main track.

[22] Vu Tran, Minh Le Nguyen, Satoshi Tojo, and Ken Satoh. 2020. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Artificial Intelligence and Law* 28, 4 (2020), 441–467.

[23] Vu Tran, Minh Nguyen, Satoshi Tojo, and Ken Satoh. 2020. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Artificial Intelligence and Law* 28 (12 2020). https://doi.org/10.1007/s10506-020-09262-4

[24] Howard Turtle. 1995. Text Retrieval in the Legal World. *Artificial Intelligence and Law* 3 (1995), 5–54.

[25] Marc Van Opijnen and Cristiana Santos. 2017. On the Concept of Relevance in Legal Information Retrieval. *Artif. Intell. Law* 25, 1 (March 2017), 65–87. https://doi.org/10.1007/s10506-017-9195-8

[26] H. Westermann, J. Šavelka, and K. Benyekhlef. 2020. Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment.. In *COLIEE 2020*.

[27] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*. https://openreview.net/forum?id=zeFrfgyZln

Predicate's Argument Resolver and Entity Abstraction for Legal Question Answering: KIS teams at COLIEE 2021 shared task

Masaki Fujita†
Faculty of Informatics
Shizuoka University
Hamamatsu, Shizuoka, Japan
mfujita@kanolabn.net

Naoki Kiyota
Faculty of Informatics
Shizuoka University
Hamamatsu, Shizuoka, Japan
nkiyota@kanolab.net

Yoshinobu Kano
Faculty of Informatics
Shizuoka University
Hamamatsu, Shizuoka, Japan
kano@inf.shizuoka.ac.jp

ABSTRACT

We developed a textual entailment system for the Japanese legal bar exam, aiming at providing explanations for the way our system solves a problem based on underlying linguistic structures. Based on our previous system, we suggested a couple of new features, such as an argument resolver that can find missing arguments of predicate-argument structures and sub-sentence divisions to accurately analyze the predicate-argument structures. We improved performance in extracting clause sets of subject, predicate, and object, which are used when comparing between problem texts and law article texts to lead the required Yes/No answers. We also employed new modules that cover broader types of problem texts while retaining their accuracies. As a result of the above improvements, we were able to increase the percentage of correct answers by 6.2% compared to our baseline in the training dataset. The test dataset, the COLIEE 2021 formal runs, does not show a good performance, for which a reason would be that this year's COLIEE 2021 test dataset includes person name to person role mapping issues in almost half of the problems, which are still almost impossible by any of the existing systems. Developing a more precise abstraction method for person and object names is our future work including the mapping issue above.

CCS CONCEPTS

• Natural Language Processing → Lexical semantics; Language resources.

KEYWORDS

COLIEE, Question Answering, Legal Bar Exam, Legal Information Extraction, Predict Argument Structure Analysis.

ACM Reference format:

Masaki Fujita, Naoki Kiyota and Yoshinobu Kano. 2021. Predicate's Argument Resolver and Entity Abstraction for Legal Question Answering: KIS teams at COLIEE 2021 shared task, In Proceedings of COLIEE 2021 workshop, ICAIL 2021, Rio de Janeiro, Brazil

1 Introduction

The COLIEE shared task series was held in association with the JURISIN (Juris-informatics) workshop and ICAIL (International Conference on Artificial Intelligence and Law [1]–[7]: overview papers. COLIEE 2021 is the eighth shared task, which consists of five tasks. Task 1 and 2 are case law tasks using Canadian law. Task 3, 4, and 5 are statute law tasks which use the Japanese legal bar examination.

In Task 3, participants are given a problem, then searche articles which are necessary to solve the problem. In Task 4, participants are given a problem and related article(s), then answer Yes or No whether the content entails the given article(s) or not regarding the given article(s). Task 5 a combination of Task 3 and Task 4; only the problem is given, then participants search for articles necessary to solve the problem if needed, finally answer Yes or No if the law article(s) entail the given problem or not. We challenged Task 4 and Task 5.

Question answering has a long research history [8]–[11], including classic approaches and machine learning approaches [12]. Recently, neural end-to-end learning methods are actively used in the question answering domain as well [13]–[15]. However, the COLIEE training dataset includes less than a thousand questions, while its potential variations are much huger as it could be any of the civil law applications. In addition, not just a descriptive knowledge in textbooks but also legal common sense is required to solve the legal bar exam, so it is difficult to directly use the past problems. Ultimately, a legal AI system is required to show evidences of its decisions in a human interpretable way. For these reasons, we do not employ the end-to-end machine learning method in our study.

In this paper, we describe an automatic problem solver for the legal bar exam that captures the structure of legal documents and handles their meaning. We focus on Task 4 and Task 5 throughout this paper. During our previous participations in the COLIEE series, we have created a program that predicts answers using syntactic analysis results [16]. We added new features to this program to improve its performance in this COLIEE 2021 challenge.

Specifically, we changed the way parsing results are used; we also replaced verbs so that clause sets of subject, predicate, and object, which are used in comparison between problem texts and law article texts, can be created more appropriately. Then we created a new type of clause sets for the comparison. We employed different combinations of these new features as different combinations of modules. We obtained 6.2% better result than our baseline.

We describe our system architecture in Section 2. Section 3 describes our experimental settings and results. We discuss error analysis in Section 4, finally conclude our paper in Section 5.

2 System Architecture

2.1 System Overview

We designed our system based on our previous system in COLIEE 2020. This subsection describes the overview; details are described in later subsections.

Our previous system parses the given problem sentence and the civil code using KNP [17], and then splits the sentence into multiple clauses based on the parsing results. Then, our system obtains a clause set of a subject, a predicate, and an object for each clause. Finally, our system compares the clause sets of the question text and the civil code to answer the Yes/No questions. We retain this architecture in COLIEE 2021, while we changed and added new features. Our major changes are as follows.

Firstly, we aimed to improve the accuracy of our clause set extraction. When the sentences are long, their parsing results were sometimes incorrect, caused the clause sets of a subject, a predicate, and an object not to be taken properly. Therefore, after parsing and dividing the sentence into multiple clauses, we attempted to get a more accurate clause set by parsing each clause again by shorter divided sub-sentences. This re-parsing is effective for nested structure sentences.

Secondly, we replace a Japanese functional verb *perform* (“行う”) with another functional verb *do* (“する”)", in order to accurately compare clause sets. Although the sentences "to perform A" and "to do A" are basically the same meaning, our previous system considers them as different verbs, thus our regards them as different. Such functional verbs very frequently occur in legal sentences with main verbs; when our analysis goes more detailed and accurate, then we need to capture roles of such functional verbs as well.

Thirdly, we improved our previous comparison modules that create Yes/No binary outputs. There were three main modules (precise match, loose match, and rough match modules) which were used to compare clause sets and derive answers in our previous system. The precise match and the loose match modules had better correct answer ratios, but few questions can be answered with these modules. In contrast, our rough match module could answer all questions but had lower accuracy. We aim to improve the overall performance by adding a new module that can solve more problems than our previous precise match and loose match modules based on our previous loose match module while retaining its accuracy.

Fourthly, we implemented a new method to fill subjects and objects in clause sets. Previously, we obtained these words directly from the parser’s results, but there were many missing subjects and objects. We implemented a method to create a clause set by ourselves.

We prepared an article search feature, which is required in the COLIEE 2018 Task 4 because this task was a question answering task without any target article given. We used the same article search feature for COLIEE 2021 Task 5, as Task 5 requires an article search. Because COLIEE 2021 Task 4 is a textual entailment task where the target gold standard articles are given, we do not use this related article search feature in Task 4.

2.2 Sentence Preprocessing

In this subsection, we describe our sentence preprocessing, which is applied before our predicate argument structure analysis. This preprocessing is almost the same as in our previous system, but we improved the parenthesis processing part as follows.

2.2.1 Itemization.

An itemization in Japanese legal document explains its result first, and then lists examples to show the result. Therefore, we need the entire article information rather than individual sentence (or item) in an itemization.

Itemizations in Japanese legal articles tend to omit subjects and predicates, which makes the linguistic post-processing more difficult. As a countermeasure, we concatenate the items with its first sentence; such an itemization starts with a first sentence common to the following items; the first sentence in an legal article’s itemization always contains a phrase *the following things* (“次に掲げる”), so we delete this specific phrase and then concatenate the texts of the following items until the target item. We repeat this process for the number of items, generating a sentence for each item. Concatenated sentences are sometimes syntactically invalid, causing errors in predicate argument structure analysis though.

2.2.2 Parenthesis.

There are sometimes punctuation marks inside parentheses. For example, in the following case “*A person who installs a window or porch (including a verandah. Hereinafter in this and the following paragraph) must put up a privacy screen.* (縁側（ベランダを含む、次項において同じ、）を設ける者は、目隠しを付けなければならない。)”, a sentence is inserted within the parenthesis, but the whole sentence makes sense both with the outer and inserted inner sentence. Texts in the parentheses sometimes indicate exceptional conditions, while sometimes may simply be a supplementary explanation, which are difficult to distinguish automatically.

In our previous system, we replaced all occurrences of punctuation marks which followed by a parenthesis close mark, with Japanese comma mark. In our COLIEE 2021 system, we regard texts in parenthesis as one of condition clauses of the entire sentence, by deleting the parenthesis in a sentence and replacing punctuation marks if any, to be a single sentence.

When a given problem consists of two or more sentences, we regard a proposition clause of the latter sentence as a proposition clause of the entire problem, and all other clauses as condition clauses of the entire problem. We define a proposition clause and a condition clause in the next section.

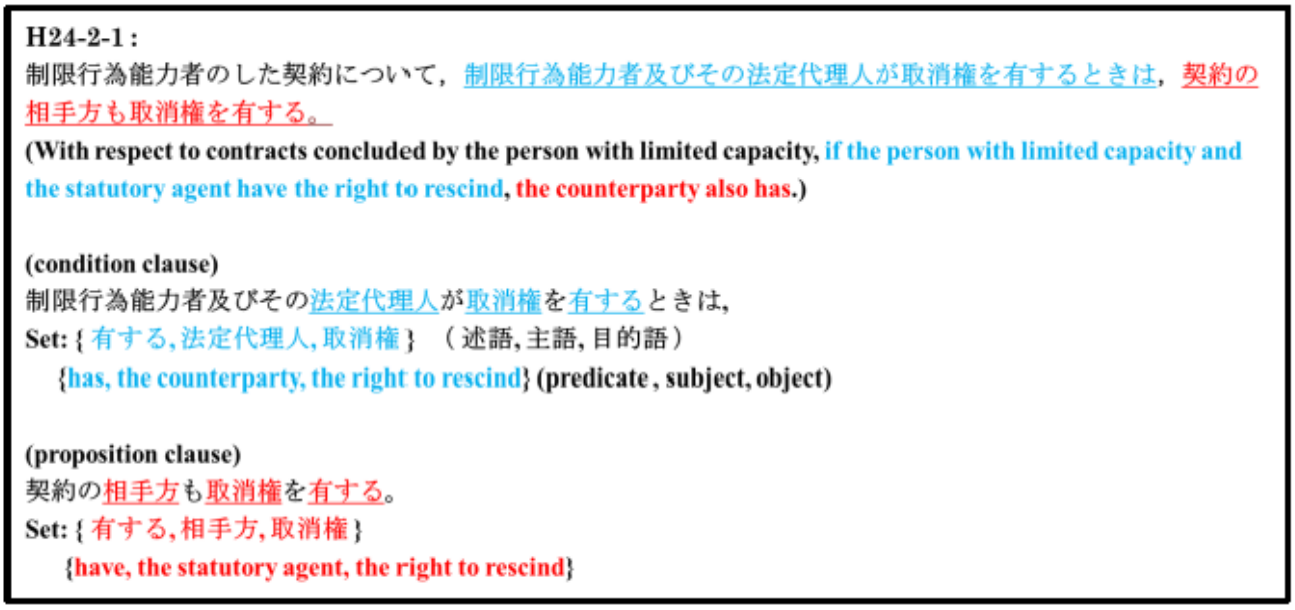

Figure 1: An example of predicate argument structure analysis

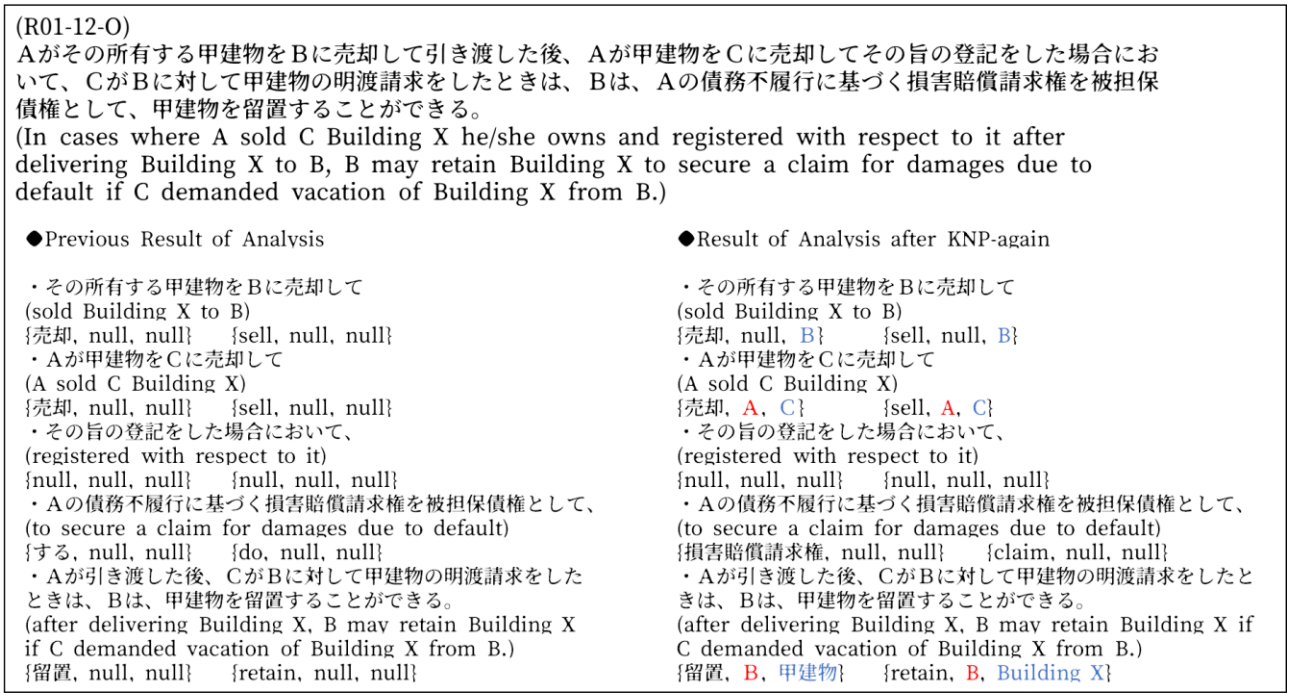

Figure 2: Comparison of parsing results between our previous system and the new sub-sentence processing

2.3 Condition Clause and Proposition Clause

We perform clause analysis as same as our previous system. Figure 1 shows an example of the predicate argument structure analysis. We defined our own original clause unit ("節") in order to recognize condition clauses and main clauses precisely, which are included in a single sentence; a clause should include a single predicate as a core element of that clause. We apply a dependency parser that makes chunks ("文節") of a couple of morphemes. Starting form a chunk that includes a predicate, we aggregate neighboring chunks when a neighboring chunk does not include any predicate, until a clause unit is formed.

A predicate is not always suitable to be a core predicate of a clause. For example, "holding" in "condition holding a court" could be regarded as a predicate. However, this is not suitable as a core single predicate in a clause because we need to compare larger predicate-argument structures rather than such a noun

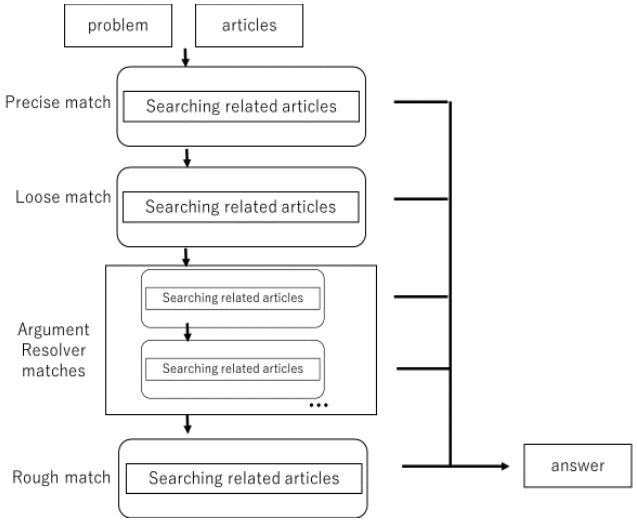

Figure 3: Overview of our module integration (Task 4)

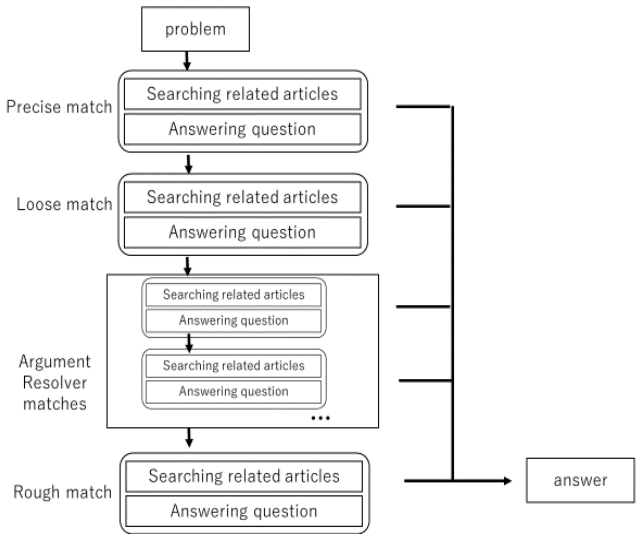

Figure 4: Overview of our module integration (Task 5)

phrase. We define two types of special clauses: a proposition clause and a condition clause. A proposition clause includes an end of the sentence. In the Japanese language, a clause which includes an end of sentence often represents a proposition. We regard a clause as a condition clause when that clause includes specific patterns, e.g." when…"," in case of …", etc.

For each sentence, we compare proposition clauses of the problem sentences and the civil law articles using these sets. The same applies for condition clauses. We use the base form of predicates for their comparison. For example, *admit* ("認める") and do not *admit* ("認めない") have the same meaning, sharing the same base form *admit* ("認める").

Normally, a sentence consists of a proposition clause and an arbitrary number of condition clauses. Because an article consists of one or more sentences, we compare sentences one by one when comparing the article sentences with the problem sentence.

2.4 Sub-sentence for Syntactic Analysis

Regarding longer sentences (e.g. shown in Figure 2) there could be many missing arguments (nulls) for predicates, subjects, and objects. This problem is caused because the sentence structure becomes more complex in longer sentences, making it difficult to analyze the detailed case relations.

2.4.1 KNP.

In our COLIEE 2021 system, after obtaining the clauses as described in the previous section, we divide the sentence text into sub-sentences so that it contains a single predicate as the core. We then apply the syntactic analyzer KNP again to this sub-sentence, performing clause analysis again to obtain clauses and their arguments (predicates, subjects, objects) again. We also aim to deal with the nested structure of sentences by this method. Figure 2 shows an example that the number of null argument clauses in the clause set is reduced by this two-pass parses, compared to the previous one-pass parsing.

There are still null arguments in our clause analysis even after this sub-sentence divisions. One of the reasons is failures of the syntactic analysis. Another reason is that arguments are scattered over multiple sentences. The other reason is that some detailed attributes are not included in predicate words, which requires to extract attribute information from other depending words. We aim to resolve the latter two issues as described later.

2.4.2 Functional Verb.

A functional verb is defined as "a verb that performs a grammatical function while leaving the substantive meaning to the noun [18]." For example, a Japanese verb *study* ("勉強する)" can be expressed in another form do a *study* ("勉強をする"). In this case, its substantive meaning can be obtained from the preceding noun *study* ("勉強"), but not from the verb *do* ("する").

Likewise, a verb *perform* ("行う") is a functional verb without a concrete meaning. Because our previous model supports the functional verb *do* ("する") to obtain the substantive meaning, we replace the verb *perform* ("行う") with the verb *do* ("する") in order to accurately compare clause sets.

2.5 Problem Solver Module

2.5.1 Precise Match Module.

Our precise match module is the same with our previous system. Our precise match module extracts articles with propositional clauses in which all of the clause's arguments (predicate, subject and object) match with arguments of a clause of a given problem statement. We use nouns' particles for each predicate to find the subject and object. If predicate's subject and object are not found, its corresponding sentence is skipped. If the sentence is in a negative form, we reverse its Yes/No answer. The accuracy of this module is around 70%, which is higher than our other modules; but on the other hand, around 80% of the problems are not applicable due to the strict match conditions.

2.5.2 Loose Match Module.

Loose match module is a looser version of the precise match. In addition to the loose match module of our previous system, we added new features to this module to cover broader types of texts.

When comparing proposition clauses and condition clauses, this module regards a pair of clauses as matched if either a pair of objects or a pair of subjects is matched, in addition to a pair of predicates. For example, if a problem's clause is {*sign* ("結ぶ"), *agent* ("代理人"), *contract* ("契約")} and if an article's clause is {*sign* ("結ぶ"), *agent* ("代理人"), *sales contract* ("売買契約")}, then our precise match module judges that they do not have a same meaning because they do not have a same object. On the other hand, our loose match module judges that they have a same meaning as this module ignores objects in this case.

2.5.3 Argument Resolver (AR).

In our COLIEE 2021's new module, we have implemented a mechanism to reduce the number of nulls (missing arguments) in the clause set and replace them with inferred contextual words, to cover more texts that can compare. We explain our mechanism as follows.

In our previous system, we directly used syntactic analysis results only, which include such predicate argument information. This method could result in better precision, but it could miss subjects and objects that were not captured by the parser. Therefore, when the above method fails to find a subject, we regard a preceding noun within that clause with a Japanese subjective case marker, *wa* ("は") and *ga* ("が") as its subject. When no object was found, we regard a noun followed by objective case markers *ni* ("に") and *wo* ("を") within the clause as its object.

There are problems that contain person names as "A" and "B", etc. Because such a style of names does not appear in the civil law articles, we cannot compare the problem text with the law articles directly. We replace such alphabetic names with an abstract word *person* ("人"). Since there are various forms of words indicating persons such as *creditor* ("債権者") and *guarantor* ("保証人"), it is difficult to make an accurate comparison without capturing precise semantics of such personal words. Because the aim of this loose match module is to cover broader types of textual expressions, we replaced these words by the abstract word *person* ("人"). Similarly, special Japanese words "甲", "乙", and "丙" (translated into *X*, *Y*, *Z* in the English version) are used to refer to physical objects in general, which are also difficult to compare. In this case, we replace these words with an abstract word *thing* ("物"). These replacements make the system not to be able to distinguish those named entities. However, it is still not possible to predict such correspondences between symbols and actual entity names, thus this would not decrease the entire system performance.

When splitting into clauses, original arguments sometimes appear in other clauses; in such cases, its clause set cannot be filled within the clause information. We filled these null arguments using its neighboring clauses if their arguments are filled.

We call these mechanisms as AR (Argument Resolver) hereafter. AR reduces the number of null arguments to allow comparisons, while it could be inaccurate compared the original clause set only. Therefore, we try different combinations of the previous method (baseline) and AR, which could result in higher accuracy with broader types of texts.

2.5.4 New Loose Match Module with AR.

The new loose match module in COLIEE 2021 aims to solve problems that could not be solved by precise match and the previous loose match module.

In order to cover more types of problem texts, we created a loosened version of the argument matching condition in the previous loose match; if more than one of a subject, a predicate, or an object matches, we perform comparison, while the previous loose match assumed the predicate should always be matched. When there are multiple candidate civil law clause sets, a clause set with the highest number of matches. We regard a pair of strings in subset relations, such as "contract" and "purchase contract", to be comparable pairs.

Since clause sets at the end of sentences are often relevant to compare between, we have been comparing only such clause sets at the end of sentences in the problem and civil law in our previous system. However, because end-of-sentence clauses are not always relevant to compare, we compare other clauses than

the end-of-sentence clauses in COLIEE 2021 depending on the setting of each module.

We created a new loose match module with the above implementations, hereafter referred to as AR (Argument Resolver) module. Two layers of AR modules, AR and AR(+law), are prepared between the existing Loose match and Rough match modules. Firstly, we apply the Precise match module if applicable; then we try applying the AR module if applicable, then AR(+law) module if applicable, finally the Rough match module.

The AR module compares clause sets at the end of the sentence the civil law and clause sets at the end of the sentence in the problem. If this comparison is not applicable due to the missing argument condition, the AR(+law) module compares the entire set of civil law clauses, i.e. compares the set of clauses at the end of, and not the end of, the civil law articles, with the clauses at the end-of-sentence of the problems.

As mentioned in the previous section, there are two types of clause sets: baseline (same as our previous system) and AR. Because the baseline is considered to be more accurate when applicable i.e. better in precision, we created another module, AAR (Abstract Argument Resolver) module, that uses both types of clause sets. When the number of matches in a clause set is the same, the baseline module is selected. During comparisons, the baseline of the problem is compared with the baseline of the civil law, and the AR of the problem is compared with the AR of the civil law. Similar to the AR modules, two layers of AAR modules are prepared between the Loose match and Rough match modules.

Problem text's clause sets other than the end of the sentence could be important in some cases. We created another module that refers to clause sets other than the end of the sentence of the problem texts. This module also uses two different clause sets for comparison as AR and AAR modules. Four layers of AAR modules were prepared between the existing Loose match and Rough match modules. Firstly, the AAR module compares the clause sets of the civil law and the end-of-sentence of the problems. If AAR is not applicable, an AAR(+law) module

compares the entire set of civil law clauses with the set of clauses at the end-of-sentence of the problems. Next, an AAR(+problem) module compares the entire clause set in the problem with the clause set at the end-of-sentence of the civil law; +problem indicates that all clause sets of the problem texts are compared regardless of end-of-sentence or not. Finally, an AAR(+law +problem) module compares all the clause sets in the problem with all the clause sets of the civil law.

2.5.5 Rough Match Module.

The Rough match module is a module for matching with the loosest conditions. This module compares only predicates, detecting negative forms of propositional clauses. This module covers all problems, but this module is too loose in its comparison. The contribution of this Rough match module decreased in our COLIEE 2021 system, by introducing the new Loose match module described above, which performs more precise analysis than the Rough match module.

2.6 Article Search

We need to search the civil law articles for relevant articles as such relevant articles are not given in Task 5. We used our article search system implemented earlier. This article search is

performed for each module; all civil law articles, which satisfy each module's comparison condition, are extracted. If more than one civil law articles are extracted, each civil law article is compared with the problem text, and then outputs a Yes/No

answer. Finally, this article search system compares the number of Yes/No answers, outputs Yes or No depending on which answer has more outputs.

3 Experiments and Results

We describe our experimental settings and results of Task 4 and Task 5 in this section. In addition to our previous system as a baseline, we employed different modules of Sub-sentence (abbreviated as Sub-S in the tables), AR, AR(+law), AAR, AAR(+law), AAR(+problem) and AAR(+law +problem) in COLIEE 2021.

3.1 Experiments using Training Dataset

Table 1 and Table 2 show the percentages of correct answers using the training data year by year. These results show that

correct answer ratios were improved both in Task 4 and 5. In Task 4, 25 more problems were correctly answered compared to the baseline.

Table 3 and Table 4 show detailed analysis of the different module combinations using the entire training data. The numbers in parentheses represent percentages of correct answers per module.

Comparing with our baseline, where more than half of the problems were answered by the Rough match module with its loosest condition, we have succeeded in reducing this number by our new loose match modules that answer problems from a more linguistic perspective.

We found that the number of problems in the Precise match module increased due to our improvements in predicate argument structure analysis, such as Sub-S. This is thought to be due to the fact that the system was able to handle the nested structure of sentences.

Table 1: Evaluation results using COLIEE training data (Task 4)

Year	#	baseline	Sub-S	AR AR(+law)	Sub-S AR AR(+law)	Sub-S AAR AAR(+law) AAR(+problem) AAR(+law +problem) (KIS1)	Sub-S AAR AAR(+law) (KIS2)	AAR AAR(+law) (KIS3)
H18	36	0.556	0.556	0.583	0.583	0.583	0.583	0.583
H19	37	0.514	0.514	0.541	0.568	0.568	0.568	0.541
H20	41	0.634	0.634	0.585	0.634	0.659	0.659	0.61
H21	54	**0.667**	0.667	**0.704**	**0.704**	**0.722**	**0.722**	**0.704**
H22	47	0.553	0.553	0.574	0.574	0.596	0.574	0.553
H23	41	0.659	**0.683**	0.634	0.585	0.61	0.61	0.659
H24	79	0.57	0.57	0.582	0.595	0.595	0.595	0.582
H25	60	0.583	0.567	0.617	0.633	0.667	0.65	0.633
H26	74	0.595	0.595	0.608	0.622	0.635	0.622	0.608
H27	49	**0.51**	**0.51**	**0.51**	**0.531**	**0.531**	**0.531**	0.531
H28	49	0.633	0.633	0.673	0.673	0.673	0.673	0.673
H29	58	0.534	0.534	0.517	0.569	0.569	0.552	**0.517**
H30	70	0.614	0.614	0.6	0.6	0.614	0.6	0.6
R01	111	0.622	0.622	0.649	0.649	0.649	0.649	0.64
	806	0.592	0.592	0.603	0.613	**0.623**	0.617	0.605

Table 2: Evaluation results using COLIEE training data (Task 5)

Year	#	baseline	DE	AR AR(+law)	Sub-S AR	Sub-S (KIS_1)	Sub-S AAR AAR(+law) (KIS_2)	Sub-S AR AR(+law) (KIS_3)
H18	36	0.528	0.5	0.5	0.528	0.528	0.5	0.528
H19	37	0.568	**0.649**	**0.649**	0.622	0.541	0.649	0.622
H20	41	**0.659**	0.634	0.634	0.634	0.659	0.634	0.634
H21	54	0.63	0.63	0.63	**0.648**	**0.667**	**0.685**	**0.685**
H22	47	0.532	0.511	0.511	0.511	0.532	0.511	0.511
H23	41	0.634	0.634	0.634	0.61	0.634	0.659	0.634
H24	79	0.532	0.519	0.519	0.57	0.557	0.582	0.582
H25	60	0.567	0.6	0.6	0.567	0.55	0.583	0.567
H26	74	0.595	0.635	0.635	0.608	0.581	0.608	0.608
H27	49	**0.51**	**0.49**	**0.49**	**0.49**	**0.51**	**0.49**	**0.49**
H28	49	0.571	0.51	0.51	0.551	0.592	0.531	0.551
H29	58	0.534	0.586	0.586	0.569	0.534	0.569	0.569
H30	70	0.586	0.571	0.571	0.543	0.571	0.543	0.543
R01	111	0.595	0.577	0.577	0.604	0.604	0.595	0.604
	806	0.574	0.574	0.574	0.577	0.577	**0.582**	**0.582**

Table 3: Number of correct answers and accuracy for each module using training data (Task 4)

Module		clause set	baseline	AR AR(+law)	Sub-S AR	Sub-S (KIS_1)	Sub-S AAR AAR(+law) (KIS_2)	Sub-S AR AR(+law) (KIS_3)
precise(baseline)		baseline	133/184 (0.723)	133/184 (0.723)	128/179 (0.715)	128/179 (0.715)	128/179 (0.715)	128/179 (0.715)
Loose	w/o AR (baseline)	baseline	109/195 (0.559)	109/195 (0.559)	117/206 (0.568)	118/206 (0.573)	118/206 (0.573)	118/206 (0.573)
	AR	AR	-	140/266 (0.526)	138/264 (0.523)	-	-	140/265 (0.528)
	AR (+law)	AR	-	80/159 (0.503)	-	-	-	81/152 (0.533)
	AAR	baseline + AR	-	-	-	-	140/265 (0.528)	-
	AAR (+law)	baseline + AR	-	-	-	-	81/152 (0.533)	-
Rough		baseline	221/428 (0.516)	1/3 (0.333)	82/158 (0.519)	219/422 (0.519)	2/5 (0.4)	2/5 (0.4)

Table 4: Number of correct answers and accuracy for each module using training data (Task 5)

Module		Clause set	baseline	Sub-S	AR AR(+law)	Sub-S AR AR(+law)	Sub-S AAR AAR(+law) AAR(+problem) AAR(+law +problem) (KIS1)	Sub-S AAR AAR(+law) (KIS2)	AAR AAR(+law) (KIS3)
Precise (baseline)		baseline	102/125 (0.816)	105/129 (0.814)	102/125 (0.816)	105/129 (0.814)	105/129 (0.814)	105/129 (0.814)	101/124 (0.815)
Loose	w/o AR (baseline)	baseline	65/106 (0.613)	67/111 (0.604)	65/106 (0.613)	67/111 (0.604)	67/111 (0.604)	67/111 (0.604)	66/107 (0.617)
	AR	AR	-	-	61/110 (0.555)	61/107 (0.57)	-	-	-
	AR (+law)	AR	-	-	159/277 (0.574)	167/281 (0.594)	-	-	-
	AAR	baseline + AR	-	-	-	-	61/107 (0.57)	61/107 (0.57)	61/109 (0.56)
	AAR (+law)	baseline + AR	-	-	-	-	173/287 (0.603)	173/287 (0.603)	164/283 (0.58)
	AAR (+problem)	baseline + AR	-	-	-	-	4/12 (0.333)	-	-
	AAR (+law +problem)	baseline + AR	-	-	-	-	29/58 (0.5)	-	-
Rough (baseline)		baseline	310/576 (0.538)	305/566 (0.539)	99/188 (0.527)	94/178 (0.528)	63/102 (0.618)	91/172 (0.529)	96/183 (0.525)

3.2 Results of COLIEE 2021 formal runs

Regarding the COLIEE 2021 formal run submissions, we submitted three runs for each of Task 4 and Task 5, based on the accuracies in the training dataset. Table 5 and Table 6 show the results of COLIEE 2021 formal runs for Task 4 and Task 5, respectively, where KISx are runs of our team. Correspondences of KIS1, KIS2, and KIS3 are shown in the column headers of Table 3 and Table 4.

While we achieved accuracies over 60% in the training dataset, accuracies in the test dataset (formal runs) fell far short of those accuracies. Because we do not use the training data as our system does not have any parameter to tune by "training", we can regard the training data results almost as same as the test data results. Therefore, this difference should be because of the different tendency between the training dataset and this year's test dataset. We discuss error analysis in the next section.

Table 5: Results of COLIEE 2021 formal run (Task 4)

Team	sid	Correct	Accuracy
	BaseLine	No 43/All 81	0.5309
HUKB	HUKB-2	57	0.7037
HUKB	HUKB-1	55	0.6790
HUKB	HUKB-3	55	0.6790
UA	UA_parser	54	0.6667
JNLP	JNLP.Enss5C15050	51	0.6296
JNLP	JNLP.Enss5C15050SilverE2E10	51	0.6296
JNLP	JNLP.EnssBest	51	0.6296
OVGU	OVGU_run3	48	0.5926
TR	TR-Ensemble	48	0.5926
TR	TR-MTE	48	0.5926
OVGU	OVGU_run2	45	0.5556
KIS	KIS1	44	0.5432
KIS	KIS3	44	0.5432
UA	UA_1st	44	0.5432
KIS	KIS2	43	0.5309
UA	UA_dl	43	0.5309
TR	TR_Electra	41	0.5062
OVGU	OVGU_run1	36	0.4444

Table 6: Results of COLIEE 2021 formal run (Task 5)

Team	sid	Correct	Accuracy
	BaseLine	No 43/All 81	0.5309
JNLP	JNLP.NFSP	49	0.6049
UA	UA_parser	46	0.5679
JNLP	JNLP.NMSP	45	0.5556
UA	UA_dl	45	0.5556
TR	TRDistillRoberta	44	0.5432
KIS	KIS_2	41	0.5062
KIS	KIS_3	41	0.5062
UA	UA_elmo	40	0.4938
JNLP	JNLP.task5.BERT_Multilingual	38	0.4691
KIS	KIS_1	35	0.4321
TR	TRGPT3Ada	35	0.4321
TR	TRGPT3Davinci	35	0.4321

4 Discussion

We confirmed that almost all of our new module systems performed better than our baseline. These results suggest that we were able to deal with nested structures of sentences, obtaining correct clause sets even in long sentences as well, by sub-sentence divisions. We were also able to answer more problems based on linguistic information, reduced the ratio using the Rough match module.

We show examples where our new modules performed their analysis correctly in Figure 5. Because the Rough match module is applied and a negation exists, the baseline answered No. In our new model using abstract expressions, the new Loose match module was able to match with the article text, returned a correct answer Yes.

Figure 6 shows another example where the baseline answered a correct answered No, while our new system answered Yes. In this example, the Rough match module of the baseline model was correct by chance; our new module was incorrect because the clauses abstracted by the Argument Resolver (AR) were matched due to its abstraction was too strong for this case. Dynamic application of the Argument Resolver depending on individual problems would be a future work.

We analyzed the test dataset to show the reason why the test dataset results (formal runs) were much worse than the training dataset results. The major reason should be the person names and their relationships. In the test dataset of this year's COLIEE 2021 challenge, there are 35 out of 81 problems include the alphabetical person names like "A" and "B", compared to 34 out of 111 in COLIEE 2020, 13 out of 70 in COLIEE 2019, 22 out of 58 in COLIEE 2018, when we count them manually. In order to correctly capture these person relationships, it is important to understand the roles of each person (creditor, owner, etc.), their relationships (sales, guardianship, etc.), and the conditions of operation (fraud, duress, etc.), then the system need to match the roles in the law articles with the alphabetical names. However, we believe that none of the existing systems in the world could perform such matches in sufficient performance yet. Therefore, if a system could achieve good accuracies in this type of problems, we need to analyze whether they capture such matches or just by chance. If there is a clear statement such as "Creditor A is …", the role can be assigned as it is. If there is no clear statement, it can be determined from the predicate. For instance, if a sentence is "A sold X to B.", then A can be judged as the person who sold the object, i.e., the seller. In this way, the roles of the characters can be determined from the expressions in the sentences. Then we can expect further improvement in accuracy by focusing on character relationships.

5 Conclusion and Future works

We developed a textual entailment system for the Japanese legal bar exam, which could explain the way system solves based on underlying linguistic structures. We applied our system to the COLIEE 2021 dataset.

In our suggested model, we improved the method of extracting clause sets of subject, predicate, and object used for later comparisons, which was a left issue in our previous system. We suggested our new method using the results of the syntactic parser and sentence preprocessing. To fill the precision-recall gap between previous modules, we created a new intermediate module that covers broader types of problems without lowering the percentage of correct answers, allowing an improvement the overall performance. In this new module, we obtained sets of clauses that contain information suitable for comparison, loosened the conditions for comparison between the problems and the civil law articles. As a result of the above improvements, we were able to increase the percentage of correct answers by 6.2% compared to our baseline in the COLIEE training dataset.

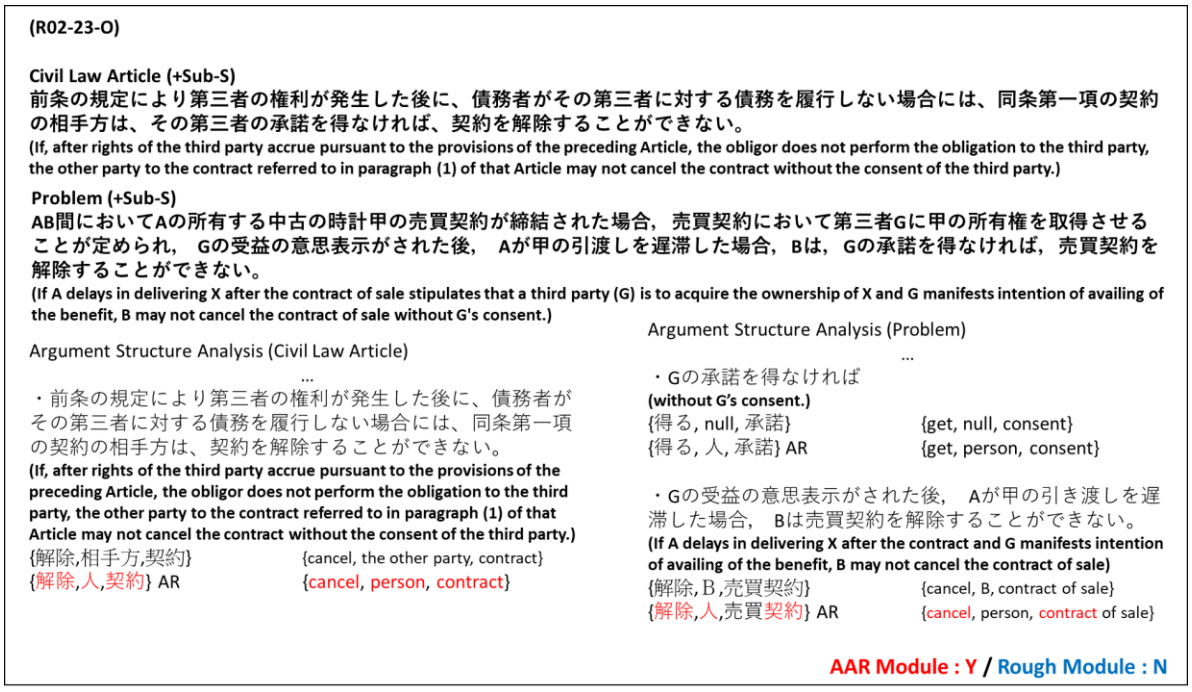

Figure 5: An example which our AR module correctly analyzed

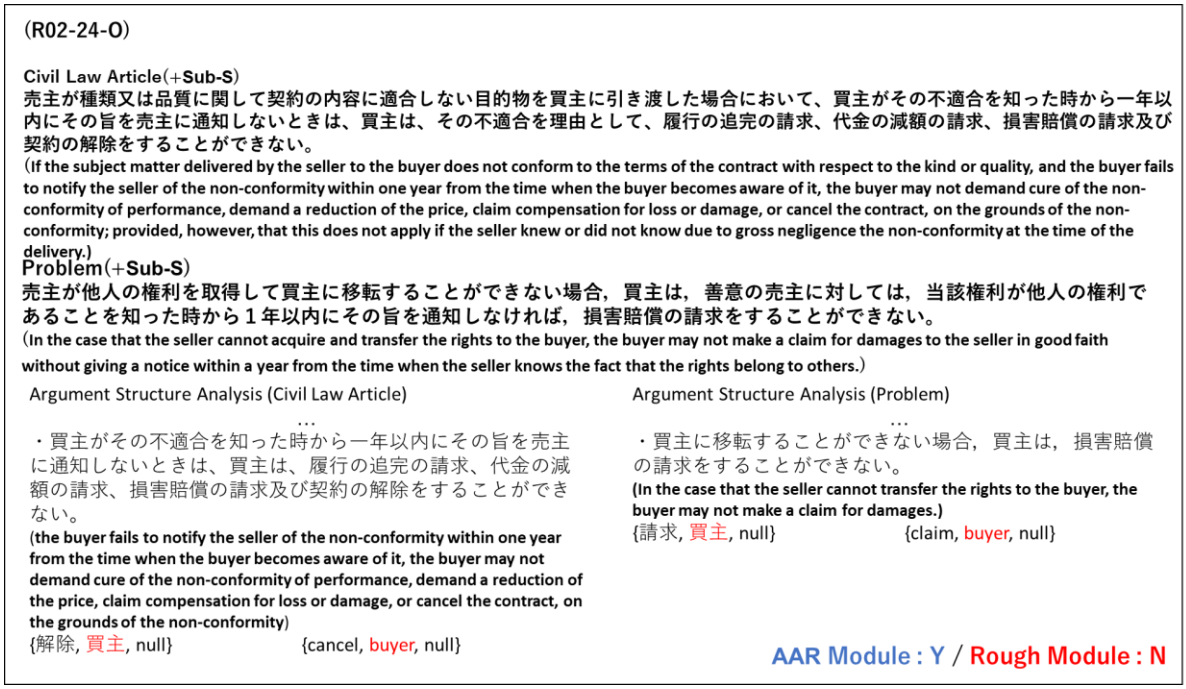

Figure 6: An example which our AR module incorrectly analyzed because of the abstract expression

Unfortunately, our formal run submissions showed lower performance compared to the training dataset, while our system does not rely on the training dataset as we do not "train" our system at all. One of the major reasons should be a specific type of the problems in the test dataset: person names and their relations, which shares more than 40% in the COLIEE 2021 test dataset. In the future, we would like to improve such person relationships.

ACKNOWLEDGMENTS

This research was partially supported by Kakenhi, MEXT Japan.

REFERENCES

[1] "Competition on Legal Information Extraction/Entailment (COLIEE-14), Workshop on Juris-informatics (JURISIN) 2014," 2014. http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html.

[2] M.-Y. Kim, R. Goebel, and K. Satoh, "COLIEE-2015 : Evaluation of Legal Question Answering," in Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN 2015), 2015, pp. 1–11, [Online]. Available: https://www.researchgate.net/publication/319311540.

[3] M.-Y. Kim, R. Goebel, Y. Kano, and K. Satoh, "COLIEE-2016: Evaluation of the Competition on Legal Information Extraction and Entailment," in Proceedings of the Tenth International Workshop on Juris-informatics (JURISIN 2016), 2016, pp. 1–13, [Online]. Available: https://www.researchgate.net/publication/319311378.

[4] Y. Kano, M.-Y. Kim, R. Goebel, and K. Satoh, "Overview of COLIEE 2017," in Proceedings of the Competition on Leal Information Retrieval and Entailment Work-shop (COLIEE 2017) in association with the 16th International Conference on Artificial Intelligence and Law, 2017, vol. 47, no. no.Icail, pp. 1–8.

[5] M. Yoshioka, Y. Kano, N. Kiyota, and K. Satoh, "Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018," in Proceedings of the Twelfth International Workshop on Juris-informatics (JURISIN 2018), 2018, pp. 1–12.

[6] R. Goebel, Y. Kano, M.-Y. Kim, J. Rabelo, K. Satoh, and M. Yoshioka, "COLIEE 2019 Overview," in Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law, Jun. 2019, pp. 1–9.

[7] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, "COLIEE 2020: Methods for Legal Document Retrieval and Entailment," in Proceedings of the Fourteenth International Workshop on Juris-informatic (JURISIN 2020), 2020, pp. 1–15.

[8] D. Lin and P. Pantel, "Discovery of Inference Rules for Question-answering," Nat. Lang. Eng., vol. 7, no. 4, pp. 343–360, 2001, doi: 10.1017/S1351324901002765.

[9] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 235–242, doi: 10.1145/860435.860479.

[10] D. Ravichandran and E. Hovy, "Learning Surface Text Patterns for a Question Answering System," in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002, pp. 41–47, doi: 10.3115/1073083.1073092.

[11] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences," in Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 129–136, doi: 10.3115/1119355.1119372.

[12] D. Ferrucci, "Introduction to 'This is Watson,'" IBM J. Res. Dev., vol. 56, no. 3.4, pp. 1:1-1:15, 2012, doi: 10.1147/JRD.2012.2184356.

[13] N. Sabharwal and A. Agrawal, "BERT Model Applications: Question Answering System," in Hands-on Question Answering Systems with BERT, 2021.

[14] H.-T. Nguyen et al., "JNLP Team: Deep Learning for Legal Processing in COLIEE 2020," in Proceedings of the Fourteenth International Workshop on Juris-informatic (JURISIN 2020), Nov. 2020, pp. 195–208, [Online]. Available: http://arxiv.org/abs/2011.08071.

[15] J. Rabelo, M.-Y. Kim, and R. Goebel, "Application of Text Entailment Techniques in COLIEE 2020," in Proceedings of the Fourteenth International Workshop on Juris-informatic (JURISIN 2020), 2020, pp. 209–222.

[16] R. Hoshino, N. Kiyota, and Y. Kano, "Question Answering System for Legal Bar Examination using Predicate Argument Structures focusing on Exceptions," in Proceedings of the Competition on Legal Information Retrieval and Entailment Workshop (COLIEE 2019) in association with the 17th International Conference on Artificial Intelligence and Law (ICAIL 2019), 2019, pp. 38–44.

[17] "Japanese Dependency and Case Structure Analyzer KNP." http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP.

[18] Y. SATO, "機能動詞の類型に関する再考 : BCCWJ における動詞性名詞「影響」の使用実態を例に(A Reconsideration of 'Factive Verbs' in Japanese : Based upon the Corpus Analysis of a Verbal Noun 'Eikyo' using BCCWJ)," 東京外国語大学日本研究教育年報 = Japanese Stud. Res. Educ. anuual Rep., vol. 21, pp. 37–54, 2016, Accessed: Apr. 20, 2021. [Online]. Available: http://ci.nii.ac.jp/naid/120006356907/ja/. (in Japanese)

BM25 and Transformer-based Legal Information Extraction and Entailment

Mi-Young Kim
miyoung2@ualberta.ca
Dept. of Science, Augustana Faculty, University of Alberta
Camrose, Alberta, Canada

Juliano Rabelo
Randy Goebel
rabelo@ualberta.ca
rgoebel@ualberta.ca
Alberta Machine Intelligence Institute, University of Alberta
Edmonton, Alberta, Canada

ABSTRACT

We describe the techniques applied by the University of Alberta (UA) team in the Competition on Legal Information Extraction and Entailment (COLIEE 2021). We participated in retrieval and entailment tasks for case law and statute law, applying a transformers-based approach for the case law entailment task, an information retrieval technique based on BM25 for legal Information Retrieval, and a natural language inference mechanism using semantic knowledge for statute law text. This competition included 25 teams from 14 countries; our case law entailment approach was ranked no. 4 in Task 2, the BM25 technique for legal information retrieval was ranked no. 3 in Task 3, and the natural language inference technique incorporating semantic information was ranked no. 4 in Task 4. The combination of the latter two techniques on Task 5 was ranked no. 2.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**; **Similarity measures**; **Clustering and classification**; *Document topic models*; *Information extraction*; Specialized information retrieval.

KEYWORDS

legal textual entailment, text classification, imbalanced datasets

ACM Reference Format:
Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2021. BM25 and Transformer-based Legal Information Extraction and Entailment. In *COLIEE'21,June, 2021, Sao Paulo, Brazil*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Tools to help legal professionals handling the increasing volume of legal documents are becoming more and more necessary. The volume of information produced in the legal sector by its many different actors (such as law firms, law courts, independent attorneys, legislators, and many other sources) is overwhelming. To help build a legal research community, the Competition on Legal Information Extraction and Entailment (COLIEE) was created, to help develop a research community by focusing on four specific problems in the legal domain: case law retrieval, case law entailment, statute law retrieval and statute law entailment. Here we provide details of our approaches for the legal information retrieval and legal text entailment tasks.

Initial techniques for open-domain textual entailment focused on shallow text features, and evolved to the usage of word embeddings, logical models and general machine learning. The current state of the art, especially for problems which have access to enough labelled data, rely on deep learning based approaches (more notably those based on transformer methods), which have shown very good results in a wide range of textual processing benchmarks, including benchmarks specific to entailment tasks.

Our method for the case law entailment task is based on our methods in the past editions [27, 28], with an increased focus on transformer methods and a heuristic post-processing technique based on *a priori* probabilities. This year we decided to drop similarity calculations, as our previous results have shown they did not significantly contribute to improved performance. For the statute law tasks, we applied BM25 for the retrieval task and a combination of a transformer-based methods and semantic information for the entailment tasks. In future, we intend to explore other techniques to capture semantic similarity as well as data augmentation approaches.

The rest of this paper is organized as follows: in Section 2 we briefly review information retrieval (IR) and textual entailment. Section 3 describes our approaches and presents our results on both case law and statute law entailment tasks in COLIEE 2020. Section 4 concludes the paper and comments on future work.

2 RELATED WORK

Textual entailment, which is also called Natural Language Inference (NLI), is a logic task in which the goal is to determine whether one sentence can be inferred from another. In the more general sentential case, the task consists of categorizing an ordered pair of sentences into one of three categories: "positive entailment" occurs when one can use the first sentence to prove that a second sentence is true. Conversely, "negative entailment" occurs when the first sentence can be used to disprove the second sentence. Finally, if the two sentences have no correlation, they are considered to have a "neutral entailment." In COLIEE, teams are challenged with the task of determining whether two case law textual fragments have a "positive entailment" relationship or not (i.e., either they

COLIEE '21, June, 2021, Sao Paulo, Brazil

ACM ISBN 978-1-4503-XXXX-X/18/06.

have "negative entailment" or "neutral entailment"). The statute law entailment task (Task 4) in COLIEE is similarly designed: the participants are required to decide if a query is entailed from the relevant civil law statues.

In the following subsections, we will discuss related research on textual entailment in general and techniques developed specifically for case law entailment.

2.1 Open-domain Textual Entailment

Textual entailment can be an independent task per se or as a component in larger applications. For example, question-answering systems may use textual entailment to identify an answer from previously stored databases [2]. Textual entailment may also be used to enhance document summarization (e.g., used to measure sentence connectivity or as additional features to summary generation [20]). Because of growing interest in textual entailment, there has been an increase in publicly available benchmarks to evaluate such systems (e.g., [3, 32]).

Early approaches for open-domain textual entailment relied heavily on exploiting surface syntax or lexical relationships, then followed by a broader range of tools, such as word embeddings, logical models, graphical models, rule systems and machine learning were developed [1]. A modern research trend for open-domain textual entailment is the application of general deep learning models, such as ELMo [24], BERT [10] and ULMFit [14].

These methods build on the approach introduced by Dai and Le [9], which showed how to improve document classification performance by using unsupervised pre-training of an LSTM [13], followed by supervised fine-tuning for domain specific downstream tasks. The pre-training is typically done on very large datasets, which do not need to be labeled and are intended to capture general language use knowledge (usually, the pre-training is formulated as a language modeling task). Subsequently, supervised learning can be used as a fine-tuning step, thus requiring a labeled but significantly smaller dataset, which aims to adjust the weights of the final layers of the model suitable for the specific task. These models have achieved impressive results in a wide range of publicly available benchmarks of different common natural language tasks, such as RACE (reading comprehension) [18] , COPA (common sense reasoning) [30] and RTE (textual entailment) [8], to name a few.

2.2 Case Law Textual Entailment

The specific task of assessing textual entailment for case law documents is quite new. The first COLIEE edition which included this task was in 2018 [17]. Chen et al. [7] proposed the application of association rules for the problem. They applied a machine learning-based model using Word2Vec embeddings [21] and Doc2Vec [19] as features. This approach faces two main problems: the lack of sufficient training data to make the models converge and generalize, and the computational cost of training, which increases exponentially on the size of the dataset. To overcome that issue, they proposed two association rule models: (1) the basic association rule model, which considers only the similarity between the source document and the target document, and (2) the co-occurrence association rule model, which uses a relevance dictionary in addition to the basic model. Another approach [26] worth mentioning approached the task as a binary classification problem, and built feature vectors comprised of the measures of similarity between the candidate paragraph and (1) the entailed fragment of the base case, (2) the base case summary and (3) the base case paragraphs (actually a histogram of the similarities between each candidate paragraph and all paragraphs from the base case). Those feature vectors are used as input to a Random Forest [4] classifier. To overcome the problem of severe data imbalance in the dataset, the dominant class was under-sampled and the rarer class was over-sampled by SMOTE sample synthesis [6]. Rabelo et al. [27] present a method for case law entailment combining similarity based features which rely on multi-word tokens instead of single words, and exploits the BERT framework [10], fine-tuned to the task of case law entailment on the provided training dataset.

2.3 Statute Law Textual Entailment

Natural language inference (NLI), the task of identifying whether a hypothesis can be inferred from, contradicted by, or not related to a premise, has become one of the standard benchmark tasks for natural language understanding. NLI datasets are typically built by asking annotators to compose sentences based on premises extracted from corpora, so that the composed sentences stand in entailment/contradiction/neutral relationship to the premise [15]. In COLIEE 2021, we have two relationships that need to be verified: entailment and non-entailment. Yang et al. [34] showed that human-created knowledge can further complement the use of pre-training models, to achieve better NLI prediction. Based on the results of Yang et al. [34], we exploit the external knowledge of the Kadokawa thesaurus [22] in Tasks 4 and 5.

For information retrieval, Shan et al. [31] claimed that empirical studies showed global representative features like BM25 capture term importance with global context information. A word with a high BM25 score reveals its uniqueness in the corpus, and this method has been widely adopted in traditional learning to rank tasks.

3 COLIEE 2021 - APPROACHES AND RESULTS

Legal question answering can be considered as consisting of a number of intermediate steps. For instance, consider a question such as "The landowner shall have the owner of the adjacent land repair or remove the obstacle if the owner of the adjacent land is damaging his or her land due to the destruction or blockage of the drainage ditch installed in the adjacent land?" In this example, a system must first identify and retrieve relevant documents, typically legal statutes. It must then compare the semantic connections between question and the relevant legal statutes, and determine whether an entailment relation holds.

COLIEE includes both retrieval and entailment tasks in two broad areas: case law and statute law. The case law retrieval task consists in finding out which cases from a pool should be "noticed" with respect to each base case in a given list. The entailment task for case law consists in determining whether an entailment relationship exists between one or more paragraphs in a referenced case and a given fragment from a base case.

For case law, the competition focuses on two aspects of legal information processing: case law retrieval (Task 1), and case law

entailment (Task 2). For statute law, the competition provides three aspects of legal information processing related to answering yes/no questions from legal bar exams: legal document retrieval (Task 3), natural language inference (NLI) for Yes/No question answering of legal queries (Task 4), and combination of document retrieval and natural language inference (Task 5).

In the next subsections, we present more details on the methods we applied in COLIEE 2021.

3.1 Case Law Entailment - Task 2

The main component of our case law entailment method applies BERT [10] by fine-tuning on the provided training dataset. BERT is a framework designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This leads to pre-trained representations which can be fine-tuned with only one additional output layer on downstream tasks, such as question answering, language inference and textual entailment, but without requiring task-specific modifications. BERT has achieved impressive results on other well-known benchmarks such as GLUE [32], MultiNLI [33] and MRPC [11].

We used a BERT model pre-trained on a large (general purpose) dataset (the goal being make it acquire general language "knowledge"[1]) which can be fine-tuned on smaller, specific datasets (the goal being to make it learn how to combine the previously acquired knowledge in a specific scenario). This makes BERT a good fit for this task, since we do not have a large dataset available for training the model. Our BERT model is based on the HuggingFace uncased-BERT distribution (bert-base-uncased), then fine-tuned on the COLIEE training dataset for 3 epochs, receiving as inputs pairs of entailment fragment and candidate paragraph, and confirming whether or not there is an entailment relationship.

We encode each candidate paragraph and its corresponding entailed fragment. If the tokenization step produces more than the 512 token limit, we apply another transformer-based model to generate a summary of the input text, and then process the pair again. Since the input text often includes text in French, we apply a simple language detection model[2] based on naive Bayesian filter to remove those fragments. As we mentioned before, in past editions we tried to enlarge the training dataset through data augmentation techniques but that did not produce the expected results. Nevertheless, we intend to further explore the data augmentation idea in future editions, probably with different techniques.

The fine-tuned model is then applied to the test dataset (with the same summarization model, when needed). The model predicts scores for the entailment and non-entailment classes, which are later considered when post processing the results. The objective of the post processing step is to add some context to the classification: the classifier itself only sees pairs of input candidate paragraphs and entailed fragments, so it could easily output a high score for many of those candidates in the same case or do not output any one with a high enough score for a different case. Whether those situations are potentially feasible, the priors show that usually there are very few actual entailing paragraphs in a case (by far, most of the cases only have one entailing paragraph). So in the post processing step we establish limits for the maximum number of outputs allowed per case. At the same time, we know at least one paragraph is the correct answer. We also make use of that fact to establish at least one paragraph should be returned, but in this case we do observe a minimum score in an attempt to reduce number the of false positives.

[1]Calling the kind of representations learned by BERT (or any other transformer based model) "knowledge" is a stretch and even some sort of anthropomorphization but seems to be appropriate in the context of machine "learning."

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

Table 1: Task 2 official results

Team	File	F1
NM	Run_task2_DebertaT5.txt	0.6912
NM	Run_task2_monoT5.txt	0.6610
NM	Run_task2_Deberta.txt	0.6339
UA	UA_reg_pp.txt	0.6274
JNLP	JNLP.task2.BM25Sup._Den..txt	0.6116
JNLP	JNLP.task2.BM25Sup._Den._F..txt	0.6091
UA	UA_def_pp.txt	0.5875
JNLP	JNLP.task2.NFSP_BM25.txt	0.5868
siat	siatCLS_result-task2.txt	0.5860
DSSIR	run_test_bm25.txt	0.5806
siat	siatFGM_result-task2.txt	0.5670
UA	UA_loose_pp.txt	0.5603
TR	task2_TR.txt	0.5438
DSSIR	run_test_bm25_dpr.txt	0.5161
DSSIR	run_test_dpr.txt	0.5161
MAN01	[MAN01] task2 run1.txt	0.5069
MAN01	[MAN01] task2 run0.txt	0.2500

Given pre-training influences on how transformer-based models "understands" language, we decided to experiment with LegalBERT [5], a BERT model fine-tuned on legal corpora. That model was fine-tuned using the same procedure described for the generic BERT model (please see above), but the final results produced were not satisfactory. We intend to go back and further explore this option in future editions of COLIEE. The pre-trained LegalBERT model used in our experiments is available at HuggingFace (model id 'nlpaueb/legal-bert-base-uncased')

The official COLIEE 2021 results for this task are shown in Table 1. Our submissions were based on a fine-tuned BERT model with summarization enabled for long paragraphs and entailed fragments. The difference between the submissions are in the post processing parameters: UA_reg_pp.txt applies a post processing which will keep at most one answer per case given its confidence score is at least -1. UA_def_pp.txt is similar but requires the minimum confidence score to be at least 0. UA_loose_pp.txt also established 0 as the minimum score but allows for at most 2 predictions to be made for each base case.

3.2 Statute Law Information Retrieval - Task 3

The key component of the probabilistic information retrieval (IR) model is to estimate the probability of relevance of the documents for a query. This is where most probabilistic models differ from one another. A number of weighting formulae have been developed and BM25 [29] has, so far, been the most effective. The major differences between BM25 and the other commonly used TFIDF models are the slight variants of inverse document frequency (IDF) formulation

and the use of the query term frequency. The length normalization factor uses the average document length and a parameter has been introduced to control the relative length effect. A probabilistic language modeling technique [25], [12] is another effective ranking model that is widely used. Typically, language modeling approaches compute the probability of generating a query from a document, assuming that the query terms are chosen independently. Unlike TFIDF models, language modeling approaches do not explicitly use document length factor and the IDF component. It seems that the length of the document is an integral part of this formula and that automatically takes care of the length normalization issue [23]. However, smoothing is crucial and it has very similar effect as the parameter that controls the length normalization components in pivoted normalization or BM25 model. Three major smoothing techniques (Dirichlet, Jelinek-Mercer and Two-stage) are commonly used in this model [35], and we use Dirichlet smoothing in our language model-based IR [25] for COLIEE 2021.

BM25 is computed as following:

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

where $f(q_i, D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in words, and *avgdl* is the average document length in the text collection from which documents are drawn. K_1 and b are free parameters. We used 1.5 for K_1 and 0.75 for b. $IDF(q_i)$ is the IDF (inverse document frequency) weight of the query term q_i. It is usually computed as:

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

where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i [3].

The legal IR task that we use to test our system has several sets of queries paired with the Japan civil law articles as documents (724 articles in total). Here follows one example of the query and a corresponding relevant article:

Question: Land owners can cut off the branches of bamboo trees in the neighboring land when they cross the border.

Related Article: Article 233 (1) When a branch of a bamboo tree in the adjacent land crosses the boundary line, the owner of the bamboo tree may cut the branch. (2) When a branch of a bamboo tree in the adjacent land crosses the boundary line, the owner of the bamboo tree may cut the branch.

Before the final test set was released, we received 14 sets of queries for a "dry run" in COLIEE 2021. The 14 sets of data include 806 queries, and 1040 relevant articles (average 1.29 articles per query). The metrics for measuring our IR model performance is F2:

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

[3]https://en.wikipedia.org/wiki/Okapi_BM25

Table 2: IR (Task3) results on test run data in COLIEE 2021.

Team	F2	P	R	MAP	R_5	R_10	R_30
OvGU_run1	0.73	0.67	0.77	0.74	0.75	0.81	0.85
JNLP.CLMLT	0.72	0.60	0.80	0.79	0.78	0.89	0.95
BM25.UA	0.70	0.75	0.70	0.75	0.71	0.73	0.81
JNLP.CLBJP	0.70	0.62	0.77	0.77	0.82	0.84	0.90
R3.LLNTU	0.70	0.66	0.74	0.78	0.79	0.83	0.91
R2.LLNTU	0.70	0.67	0.73	0.78	0.78	0.84	0.91
R1.LLNTU	0.68	0.63	0.73	0.78	0.78	0.84	0.91
JNLP.CLBJ	0.68	0.55	0.77	0.77	0.81	0.84	0.91
OvGU_run2	0.67	0.48	0.80	0.75	0.75	0.81	0.90
TFIDF.UA	0.65	0.67	0.65	0.73	0.72	0.74	0.81
LM.UA	0.54	0.56	0.54	0.64	0.64	0.68	0.81
TR_HB	0.52	0.33	0.61	0.66	0.71	0.74	0.84
HUKB-3	0.52	0.29	0.69	0.61	0.68	0.74	0.87
HUKB-1	0.47	0.23	0.65	0.61	0.66	0.75	0.87
TR_AV1	0.35	0.26	0.51	0.46	0.43	0.47	0.56
TR_AV2	0.33	0.14	0.55	0.43	0.39	0.44	0.49
HUKB-2	0.32	0.32	0.32	0.41	0.46	0.54	0.61
OvGU_run3	0.30	0.15	0.70	0.55	0.57	0.61	0.70

Table 2 shows the results of experiments with our three IR models on the final test set in COLIEE 2021: BM25 (BM25.UA), TF-IDF (TFIDF.UA), and language-model based IR (LM.UA). BM25 showed the best performance amongst the three models. The test data size is 81 queries for Task 3. The performance of our system was ranked 3rd among the submitted systems in the Competition on Legal Information Extraction/Entailment (COLIEE) 2021.

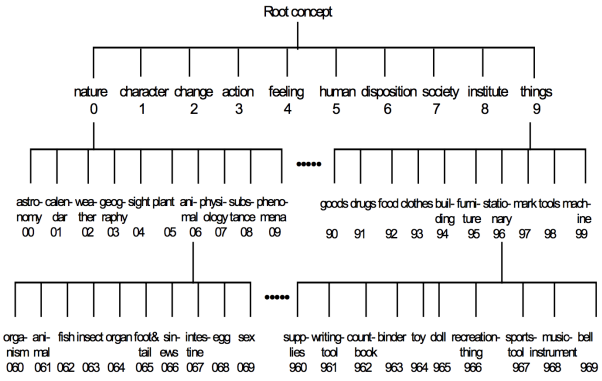

Figure 1: Kadokawa Thesaurus Hierarchy [16]

3.3 Answering Yes/No Questions - Tasks 4 and 5

The problem of answering a legal yes/no question can be viewed as a binary classification problem. Assume a set of questions Q, where each question $q_i \in Q$ is associated with a list of corresponding article sentences $a_{i1}, a_{i2}, \ldots, a_{im}$, where $y_i = 1$ if the answer is 'yes' and $y_i = 0$ otherwise. We choose the most relevant sentence a_{ij} using the algorithm of Kim et al. [28], and we simply treat each data point as a triple (q_i, a_{ij}, y_i). Therefore, our task is to learn a classifier over these triples so that it can predict the answers of

Table 3: NLI (Task 4) results on test data.

Team	sid	Correct	Accuracy
	BaseLine	Yes 43/All 81	0.5309
HUKB	HUKB-2	57	0.7037
HUKB	HUKB-1	55	0.6790
HUKB	HUKB-3	55	0.6790
UA	UA_parser	54	0.6667
JNLP	JNLP.EC	51	0.6296
JNLP	JNLP.ECS	51	0.6296
JNLP	JNLP.EB	51	0.6296
OVGU	OVGU_run3	48	0.5926
TR	TR-Ensemble	48	0.5926
TR	TR-MTE	48	0.5926
OVGU	OVGU_run2	45	0.5556
KIS	KIS1	44	0.5432
KIS	KIS3	44	0.5432
UA	UA_1st	44	0.5432
KIS	KIS2	43	0.5309
UA	UA_dl	43	0.5309
TR	TR_Electra	41	0.5062
OVGU	OVGU_run1	36	0.4444

any additional question-article pairs. BERT [10] has shown good performance on the natural language inference tasks. However, Jiang and Marnaffe [15] insisted that despite high F1 scores, BERT models have systematic error patterns, suggesting that they still do not capture the full complexity of human pragmatic reasoning. To help the pragmatic reasoning, our system incorporates the semantic information into the BERT language model for natural language inference. We add corresponding semantic category numbers of the Kadokawa thesaurus to content words as an additional feature, as shown in Figure 1. Through the numbers of the thesaurus, the semantic closeness between two words can be figured out.

Table 3 shows the Task 4 results on test data in COLIEE 2021. In the table, UA_dl is the result of BERT without incorporating semantic information, and UA_parser is the result of BERT with semantic information (Kadokawa thesaurus concept number). UA_parser was ranked no. 4 in Task 4 of COLIEE 2021.

The difference between Task 4 and Task 5 is whether the gold standard answer for the relevant statutes is used or not. In Task 4, participants use the gold standard relevant statutes provided by the organizers, while in Task 5, participants use the retrieved statutes using their own results of Task 3. Table 4 shows the results of the submitted systems in COLIEE 2021 for Task 5. When we submitted the Task 5 results, we did not know that our BM25 technique showed the best performance amongst our three submitted systems in Task 3. So, we chose the output of a traditional TF-IDF technique for IR, and combined the IR output with our NLI systems for Task 5 submission. Our system combining TF-IDF in IR (Task 3) + NLI (Task 4) was ranked no. 2. As future work, we will combine our NLI approach with our BM25 technique and see if it can improve our current Task 5 performance.

Table 4: Task 5 (IR+NLI) results on test data in COLIEE 2021.

Team	sid	Correct	Accuracy
	BaseLine	No 43/All 81	0.5309
JNLP	JNLP.NFSP	49	0.6049
UA	UA_parser	46	0.5679
JNLP	JNLP.NMSP	45	0.5556
UA	UA_dl	45	0.5556
TR	TRDistillRoberta	44	0.5432
KIS	KIS_2	41	0.5062
KIS	KIS_3	41	0.5062
UA	UA_elmo	40	0.4938
JNLP	JNLP.task5.B_M	38	0.4691
KIS	KIS_1	35	0.4321
TR	TRGPT3Ada	35	0.4321
TR	TRGPT3Davinci	35	0.4321

4 CONCLUSIONS

We explained our models for legal entailment and question answering in COLIEE 2021. For the case law entailment task, our transformers-based system ranked 4th place among all submissions (2nd among all teams). Our future work will include exploring combinations of complementary techniques as well as alternatives for appropriate data augmentation for task 2. We have experimented with that in the past without much success (please see [28] for more details), but we believe there it can yield better results if we can find alternative data sources. For the statute law tasks, our BM25 system was ranked 3rd in Task 3, and our NLI system combining BERT and semantic information was ranked 4th in Task 4 (we were the 2nd best team in that task) and 2nd in Task 5. As future research, we will investigate to obtain semantic representation from paragraph and perform natural language inference between paragraphs.

ACKNOWLEDGEMENTS

This research was supported by Alberta Machine Intelligence Institute (AMII), and would not be possible without the significant support of Colin Lachance from vLex and Compass Law, and the guidance of Jimoh Ovbiagele of Ross Intelligence and Young-Yik Rhim of Intellicon.

REFERENCES

[1] Ion Androutsopoulos and Prodromos Malakasiotis. 2009. A Survey of Paraphrasing and Textual Entailment Methods. *CoRR* abs/0912.3747 (2009). arXiv:0912.3747 http://arxiv.org/abs/0912.3747

[2] Asma Ben Abacha and Dina Demner-Fushman. 2019. A Question-Entailment Approach to Question Answering. *CoRR* abs/1901.08079 (2019). arXiv:1901.08079 http://arxiv.org/abs/1901.08079

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL.

[4] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32.

[5] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

[6] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Int. Res.* 16, 1 (June 2002), 321–357.

[7] Ying Chen, Yilu Zhou, Zhen Lu, Hao Sun, and Wenjun Yang. 2018. Legal information retrieval by association rules. In *Twelfth International Workshop on Juris-informatics*.

[8] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *ML Challenges Workshop*. Springer, 177–190.

[9] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. *CoRR* (2015). arXiv:1511.01432

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805

[11] William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proc. of the 3rd International Workshop on Paraphrasing*.

[12] Robertson S. Zaragoza H. Hiemstra, D. 2004. A language modeling approach to information retrieval. In *Parsimonious language models for information retrieval*. 178–185.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14] Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned Language Models for Text Classification. *CoRR* abs/1801.06146 (2018). arXiv:1801.06146 http://arxiv.org/abs/1801.06146

[15] N. Jiang and M.C. de Marneffe. 2019. Evaluating BERT for natural language inference: A case study on the CommitmentBank. 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)*. 6088–6093.

[16] Sin-Jae Kang and Jong-Hyeok Lee. 2001. Semi-automatic practical ontology construction by using a thesaurus. In *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*. 413–419.

[17] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. COLIEE-2018: Evaluation of the Competition on Legal Information Extraction and Entailment. In *12rh International Workshop on Juris-informatics*.

[18] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. *CoRR* abs/1704.04683 (2017). arXiv:1704.04683

[19] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* abs/1405.4053 (2014). arXiv:1405.4053

[20] Elena Lloret, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. 2008. A Text Summarization Approach under the Influence of Textual Entailment. In *NLPCS - 5th International Workshop on Natural Language Processing and Cognitive Science*. 22–31.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR* (2013).

[22] S. Ohno and M. Hamanishi. 1981. *MNew Synonyms Dictionary, Kadogawa Shoten*. Tokyo.

[23] Jiaul H Paik. 2013. A novel TF-IDF weighting scheme for effective ranking. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 343–352.

[24] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[25] Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 275–281.

[26] Juliano Rabelo, Mi-Young Kim, Housam Babiker, Randy Goebel, and Nawshad Farruque. 2018. Legal Information Extraction and Entailment for Statute Law and Case Law. In *Twelfth International Workshop on Juris-informatics (JURISIN)*.

[27] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2019. Combining Similarity and Transformer Methods for Case Law Entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law* (Montreal, QC, Canada) *(ICAIL '19)*. Association for Computing Machinery, New York, NY, USA, 290–296.

[28] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2020. Application of Text Entailment Techniques in COLIEE 2020. In *JURISIN*.

[29] Zaragoza H. Robertson, S. 2009. The probabilistic relevance framework: BM25 and beyond. In *Found. Trends Inf. Retr.* 333–389.

[30] Melissa Roemmele, Cosmin Bejan, and Andrew Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium Series*.

[31] Xuan Shan, Chuanjie Liu, Yiqian Xia, Qi Chen, Yusi Zhang, Kaize Ding, Yaobo Liang, Angen Luo, and Yuxiang Luo. 2020. GLOW : Global Weighted Self-Attention Network for Web Search. arXiv:2007.05186 https://arxiv.org/abs/2007.05186

[32] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *CoRR* abs/1804.07461 (2018). arXiv:1804.07461 http://arxiv.org/abs/1804.07461

[33] Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *CoRR* abs/1704.05426 (2017). arXiv:1704.05426

[34] Xiaoyu Yang, Xiaodan Zhu, Huasha Zhao, Qiong Zhang, and Yufei Feng. 2019. Enhancing unsupervised pretraining with external knowledge for natural language inference. In *Proc. of the Canadian Conference on Artificial Intelligence*. Springer, 413–419.

[35] Lafferty J. Zhai, C. 2004. A study of smoothing methods for language models applied to information retrieval. In *ACM Trans. Inf. Syst.* 179–214.

SIAT@COLIEE-2021: Combining Statistics Recall and Semantic Ranking for Legal Case Retrieval and Entailment

Jieke Li[1,2*], Xiaoyan Zhao[1,2*], Junhao Liu[1,2*], Jiabao Wen[1,2], Min Yang[1†]
[1]Shenzhen Key Laboratory for High Performance Data Mining,
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{jk.li1,xy.zhao,jh.liu,jb.wen,min.yang}@siat.ac.cn

ABSTRACT

In the 2021 Competition on Legal Information Extraction/Entailment (COLIEE-2021), we participate in two case law tasks. Task 1 is a legal case retrieval task aiming to automatically extract supporting cases from the corpus for a given new case. We propose a pipeline method based on statistics features and semantic understanding models, which enhances the retrieval method with both recall and semantic ranking. Task 2 is an entailment task whose goal is to identify specific paragraphs in the relevant case that entail a decision of a new case. We introduce a BERT-Legal model to enrich the language understanding model with legal knowledge. What's more, we further explore the data augmentation and adversarial training in the legal domain to enhance the performance. Our team SIAT respectively ranks 4th among all teams in Task 1 and Task 2. Experimental results show that our methods can help overcome the lack of semantic understanding of legal data and obtain competitive performance.

ACM Reference Format:
Jieke Li[1,2] [1], Xiaoyan Zhao[1,2] [1], Junhao Liu[1,2] [1], Jiabao Wen[1,2], Min Yang[1] [2]. 2021. SIAT@COLIEE-2021: Combining Statistics Recall and Semantic Ranking for Legal Case Retrieval and Entailment. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

The law affects our daily lives and therefore constitutes a vital information resource. The ability to retrieve and review supporting materials in a large collection of documents is an essential part of legal research, especially for lawyers to prepare the legal reasoning. With the continuous generation of legal information and the improvement of user demands, traditional legal retrieval model uses keyword-based retrieval systems to access legal information with low efficiency and effectiveness, which seems to bring limited satisfaction to users. The superior performance of artificial intelligence technology has stimulated research aimed at improving the quality of tools used to process legal texts.

The Competition on Legal Information Extraction / Entailment (COLIEE) is an annual competition held since 2014, which aims to find automated techniques for legal texts retrieval and entailment. In COLIEE-2021, there are five tasks covering two legal systems in the world, statute law and case law. The five tasks organized by COLIEE can be divided into two categories of retrieval and entailment task. We participate in two tasks of the competition, the legal case retrieval task and the legal case entailment task (Task 1 & 2).

In COLIEE-2019, plenty of teams explored to apply the combination of feature engineering and deep learning to the competition tasks. JNLP team [11] was the winning team in COLIEE-2019 for Task 1, which combined traditional lexical matching with deep learning. UA team [7] achieved the best performance in COLIEE-2019 for Task 2 by using the BERT model and lexical similarity. In COLIEE-2020, THUIR team [10] introduced both exact matching and semantic understanding by using word-entity duet framework and BERT-PLI model in the law tasks. JNLP team [6] utilized BERT model by fine-tuning on additional law dataset in Task 2 and achieved promising results.

In this paper, we propose several novel solutions on Task 1 and Task 2 in COLIEE-2021. The design of our models fully considers the pre-training model and text semantic understanding for the specific legal field, which extremely useful for solving legal-related decision support. Concretely, we introduce an effective pipeline method to joint the effective recall and semantic ranking in Task 1, the legal case retrieval. In Task 2, we propose a BERT-Legal model enriched with legal knowledge, to capture the entailment relationship between the query paragraph and the supporting paragraphs. Moreover, we explore data augmentation and adversarial training for further improvement. Experimental results show that such domain-adaptive training achieves good generalization performance for current legal case retrieval and entailment tasks. And our team SIAT ranks 4th among all teams in Task 1 and Task 2, respectively.

2 TASK DESCRIPTION

2.1 Task 1: Legal Case Retrieval

Task 1 is the legal case retrieval task, which consists of reading a new case (query case Q) and recalling supporting cases ($D = \{d_1, d_2, \ldots, d_n\}$) for the decision of Q from the entire case law corpus. The "supporting case" (or "noticed case") is a legal term that denotes the precedent in a given collection, which can support the

*Equal contribution
†Corresponding author

COLIEE 2021, June 21, 2021, Online

new query case. In our paper, we formulate this task as a relevant case retrieval problem applied to the domain of the legal text.

2.2 Task 2: Legal Case Entailment

Task 2 is the legal case entailment task, which is used to identify all the specific paragraphs p_i in a relevant case $R = \{p_1, p_2, \ldots, p_n\}$ that entail the decision Q of a new case. The difference from the retrieval task (eg. Task 1) is that many related paragraphs retrieved in R do not necessarily entail the query fragments. Formally, this task is to find all pairs that satisfy $entail(Q, p_i)$, where p_i is the i-th paragraph in the corresponding relevant case R.

2.3 Dataset

Both of Task 1 and Task 2 datasets are drawn from an existing collection of predominantly Federal Court of Canada case law. The dataset statistics details of Task 1 and Task 2 are described in Table 1. In Task 1, there are 250 query cases for training and testing respectively and both test and training query retrieve cases from the same case collection which contains 4415 cases. In Task 2, the training and testing sets contain 426 and 100 base cases respectively. The ground truth labels of the training set are provided in both tasks. All the participants are required to submit the retrieval and entailment results on the testing set for online evaluation purposes.

Table 1: The statistics of datasets in COLIEE-2021 Task 1 and Task 2.

Task 1	**Train**	**Test**
# Query case	250	250
# Candidate cases / Query	4415	4415
# Noticed cases / Query	5.09	3.6
Task 2	**Train**	**Test**
# Query case	426	100
# Candidate paragraphs / Query	35.72	35.24
# Entailing paragraphs / Query	1.17	1.17

2.4 Evaluation Measure

Both tasks use micro-average precision, recall, and F1-measure as evaluation metrics, which are formulated as follows:

$$\begin{aligned} Precision &= \frac{N_{TP}}{N_{TP} + N_{FP}}, \\ Recall &= \frac{N_{TP}}{N_{TP} + N_{FN}}, \\ F1 &= \frac{2 \cdot P \cdot R}{P + R}, \end{aligned} \tag{1}$$

where N_{TP} denotes the number of correct positive prediction for all queries, $N_{TP} + N_{FP}$ is the total positive prediction number for all queries, and $N_{TP} + N_{FN}$ is the ground truth positive cases number.

3 METHODS

3.1 Task 1: Legal Case Retrieval

3.1.1 Efficient Token Matching Recall. Traditional retrieval models mostly utilize the token-level exact matching and statistics information based on the bag-of-words model. Term frequency–inverse document frequency (TF-IDF) has been verified efficiency in information retrieval tasks such as Ad-hoc [1]. For each word w_i in query case Q, the term frequency (TF) score with candidate case d_j is calculated as follows:

$$TF(w_i, d_j) = \frac{count(w_i \in d_j)}{\left\|d_j\right\|}, \tag{2}$$

where $count(w_i \in d_j)$ indicates the occurrence number of word w_i in case d_j, $\left\|d_j\right\|$ denotes the document length. Term frequency tends to emphasize cases which includes the common word more frequently, while underestimate the importance of more meaningful topic word in query and case. To overcome this problem, inverse document frequency (IDF) is proposed to measure the informative degree of the provided word w_i which is formulated by:

$$IDF(w_i, D) = \log \frac{\|D\|}{count(w_i \in D)}, \tag{3}$$

$\|D\|$ is the candidate size of supporting cases, $count(w_i \in D)$ calculates the the occurrence number of word w_i in candidate set D. The relevant score between query Q and case d_j is defined as:

$$sim(Q, d_j) = \sum_{w_i \in Q} TF(w_i, d_j) * IDF(w_i, D). \tag{4}$$

As we mentioned above, we consider Task 1 as a ranking problem and recall top-50 cases ranked by Eq. (4) as supporting cases. Before calculating the similarity score, we preprocess all documents by removing stop words and converting words into lower case via NLTK tool [4]. We apply BM25 [8], an advanced TF-IDF implementation, as our retrieval engine to efficiently recall relevant cases.

3.1.2 Contextual Based Semantic Ranking. Traditional information retrieval methods can efficiently recall relevant documents while lack of contextual understanding ability to further discriminate the semantic relevance between query and document. Large-scale pre-trained language models (e.g. ELMo [8], BERT [2], etc.) have been demonstrated effectively on leveraging text context to language understanding which approaches human-like performance on most language understanding tasks such as machine reading comprehension, natural language inference. However, it still remains a big challenge on utilizing pre-trained language models for the legal text ranking task, for its text-domain drift and extreme long text.

We mitigate these problems from the following two aspects. Firstly, we propose a post-training strategy to overcome the text domain drift. In details, we further pretrain the language model on legal text corpus which enhances the language representation generalization ability on the target legal domain data. We abbreviate the post-training language model as BERT-Legal in the remaining content, and the post-training details will be described in section 3.2.1. Secondly, to make the language model feasible to process extreme long content, we model the semantic relevance in a hierarchical way consisting of paragraph-level and document-level. The overall model details are illustrated in Figure 1.

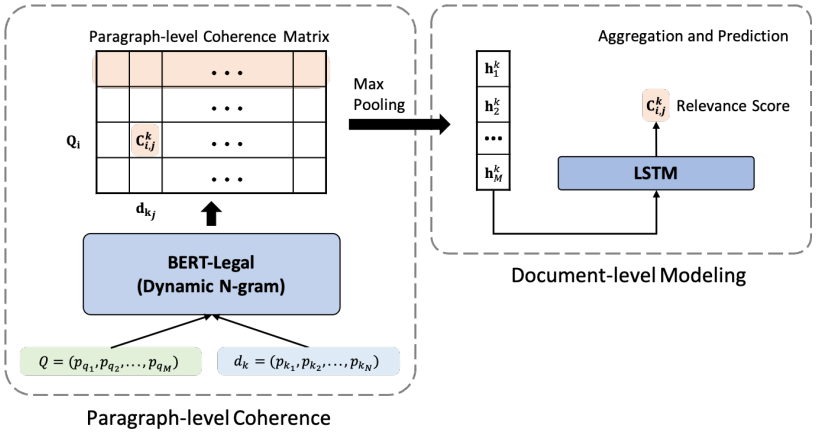

Figure 1: The detailed architecture of BERT-Legal based semantic ranking model.

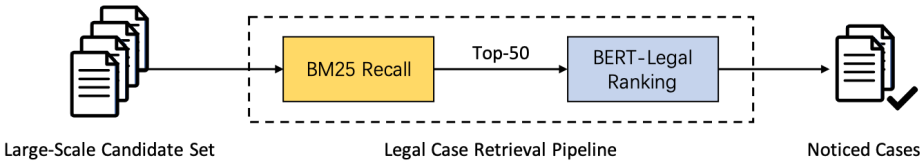

Figure 2: The detailed illustration of legal case retrieval pipeline.

For each query case and candidate case, we fragment a long document into paragraph-level units, which can be denoted as $Q = (p_{q_1}, p_{q_2}, \ldots, p_{q_M})$ and $d_k = (p_{k_1}, p_{k_2}, \ldots, p_{k_N})$ for query and candidate case respectively. Each paragraph sequence is encoded by a pre-trained language model (i.e. BERT-Legal) into a semantic representation and obtaining the hidden vector at [CLS] token position. As a result, the paragraph-level features of query and candidate case can be denoted as $\mathbf{Q} \in \mathbb{R}^{M \times d_h}$, $\mathbf{d}_k \in \mathbb{R}^{N \times d_h}$, where d_h is the hidden dimension of BERT-Legal. The paragraph-level coherence matrix of $M \times N$ pairs of sequences can be calculated as:

$$\mathbf{C}^k = \mathbf{Q} \otimes \mathbf{d}_k^T \in \mathbb{R}^{M \times N \times d_h}, \ \forall k \in \{1, \ldots, n\}, \tag{5}$$

where $\otimes$ indicates the broadcast element-wise multiplication.

To aggregate the paragraph-level features into document-level understanding, we use an LSTM to encode the coherence matrix $\mathbf{C}^k$:

$$\begin{aligned} \mathbf{h}_i^k &= Max(\mathbf{C}_{i,1}^k, \ldots, \mathbf{C}_{i,N}^k) \in \mathbb{R}^{d_h}, \\ \tilde{\mathbf{h}}^k &= LSTM(\mathbf{h}_1^k, \ldots, \mathbf{h}_M^k) \in \mathbb{R}^{d_l}, \end{aligned} \tag{6}$$

where $\tilde{\mathbf{h}}^k$ is the document relevance features, d_l is the hidden size of LSTM. Finally, a linear classifier is adopted to finish relevance prediction:

$$\hat{y}_k = softmax(\mathbf{W}\tilde{\mathbf{h}}^k + \mathbf{b}), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{2 \times d_l}$, $\mathbf{b} \in \mathbb{R}^{d_l}$. At the training stage, we optimize the ranking model with cross-entropy loss:

$$\mathcal{L}_{rank} = -y_k \log \hat{y}_k - (1 - y_k) \log(1 - \hat{y}_k), \tag{8}$$

where $y_k = 1$ indicates the query and document are relevant.

Negative Sampling. We propose a negative sampling strategy to construct the semantic ranking training set. Specifically, the irrelevant documents in the top-50 recall cases are labeled as hard negative samples while the ground truth relevant cases are used as positive samples. The semantic ranking model is taught to learning to select the relevant cases from those hard negative candidates.

3.1.3 Retrieval Pipeline. To balance the advantage of efficient recall and semantic ranking, we combine the recall and ranking as a pipeline method shown in Figure 2. In details, given a new query Q we first use BM25 to recall top-50 relevant documents. The BERT-Legal based ranking model then ranks the small recall set into a more rational list. This pipeline method makes large-scale case retrieval feasible and reduces the inference cost.

3.2 Task 2: Legal Case Entailment

3.2.1 Pretraining BERT with N-gram Representations. We format our inputs as "[CLS] Q [SEP] p_i [SEP]", where Q is the desicion segment of a new case and p_i is the segment of a relevent case R. It is noted that a segment usually consists of more than one sentence. [CLS] is the special classification label, which is always the first label of each sequence.

Similar to BERT, we use the same vocabulary and tokenize the inputs using SentencePiece [3] as in XLNet. We assume that BERT has achieved impressive performance in several NLP tasks for its strong knowledge in the general field, but there has been limited to investigate its adaptation in the legal field. Therefore, we propose a pre-training task on BERT (BERT-base-uncased), dynamic N-gram

masking, to get a special BERT model with legal knowledge (BERT-Legal). We utilize n-gram masking [2] to generate masked inputs for "masked language model" (MLM) targets, where the input texts are all legal articles in the Task 1 dataset. The length of each n-gram mask is randomly selected from the length candidate set n=1,2,3. The probability calculation method of length n $p(n)$ is given by

$$p(n) = \frac{1/n}{\sum_{k=1}^{N} 1/k}. \tag{9}$$

The maximum length of n-gram is set to n=3, which means that the target of MLM can contain up to a complete 3-gram words, like "United Nations Building". The process of generating masked training samples is dynamic, that is, the masked word in the same sample is randomly selected again in the next epoch. Compared with the MLM target of BERT to predict the masked single words, the prediction target of the N-gram method is the fragments of words, which contain more semantic information than the previous method.

The total number of tokens in the input paragraph pairs exceed the length limit (512) and will be truncated in alignment. The input text is the concatenation of constructed paragraph pair (Q, p_i). The final hidden state corresponding to [CLS] is considered to be the representation aggregated the whole sequence in the classification task. Thus, the final prediction result is obtained by feeding the final hidden state vector [CLS] obtained by BERT-Legal into the output layer for binary classification.

We assume that the BERT-Legal learned both general linguistic features and legal-specific features by fine-tuning BERT with dynamic N-gram masking on legal texts, which is beneficial to improve the performance of the following tasks.

3.2.2 BERT-Legal with Domain Data Augment. Considering that the limited labeled data in Task 2 may affect the accuracy of prediction and the generalization ability of our model, we use the labeled data in Task 1 to augment corpus and obtain more training data in Task 2.

The specific strategy of data augment is: (1) For the positive "document pair" in Task 1, we segment each document and construct candidate paragraph pairs from the paragraphs in the positive "document pair". Then, we calculate the BM25 score in each paragraph pairs and take top-5 as the additional positive samples of Task 2. (2) And vice versa for augmenting the negative paragraph pair, we take the top-5 paragraph pairs of BM25 score as the augmented negative samples, where the paragraph pairs are extracted from negative "document pair". (3) Finally, 50,000 additional samples (equally divided between positive and negative samples) are added to the training process of BERT-Legal model in Task 2, denoted as BERT-Legal (+Data Augment).

3.2.3 Adversarial Training with Fast Gradient Method. Adversarial training is aimed to train a model to correctly classify unmodified examples and adversarial examples. Adversarial examples designed to substantially increase the model loss are examples different from unmodified examples, which are created by making subtle perturbations on inputs. Previous work mainly applied adversarial training to image classification tasks in computer vision (CV) and text classification tasks in nature language processing (NLP). It not only improves the robustness of adversarial examples, but also improves

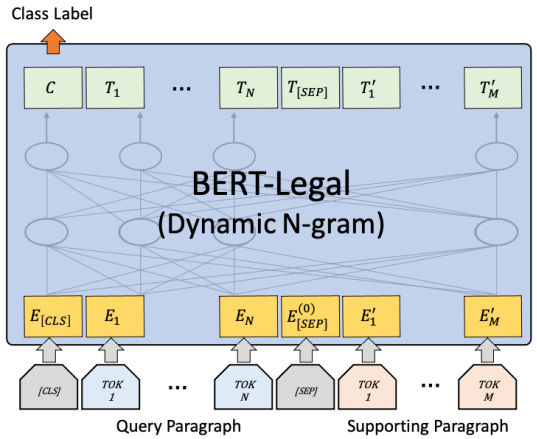

Figure 3: An detailed illustration of the fine-tuning BERT-Legal model.

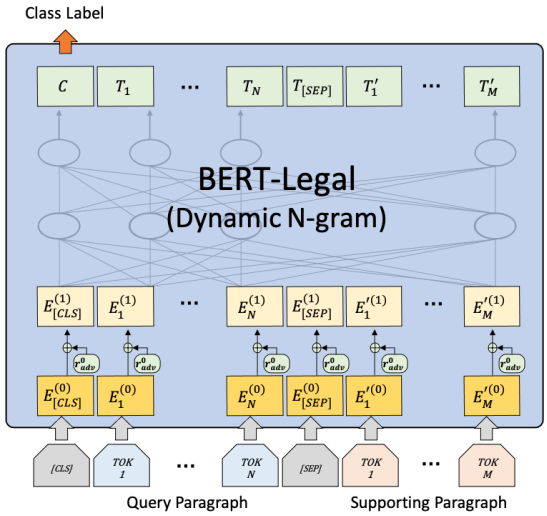

Figure 4: The detailed illustration of the fine-tuning BERT-Legal (+FGM) model.

the generalization performance of original examples. Encouraged by these results, we extend these techniques to legal text process in legal case entailment task and train BERT-Legal with Fast Gradient Method (FGM) [5], denoted as BERT-Legal (+FGM).

As shown in Figure 4, we train BERT-Legal with FGM to be robust to the embedding disturbances with adversarial training. Here, we describe these perturbations in detail. FGM strategy proposed by [5] emphasizes that advanced performance can be achieved by optimizing the additional hyperparameter ϵ, where ϵ is norm constraint to limit the size of adversarial perturbations. The parameters of BERT-Legal is denoted as θ and the input is denoted as x, which is the concatenation of constructed paragraph pair (Q, p_i). When adversarial training is applied to BERT-Legal, the following items

Table 2: The experimental results of our methods on the Task 1 development subset.

Model Variant	Precision	Recall	F1-score	Acc	Auc
BERT-Legal (MPN 15)	0.650	0.506	0.570	**0.917**	0.737
BERT-Legal (MPN 20)	**0.675**	**0.538**	**0.599**	0.906	**0.750**

Table 3: The experimental results of our methods on the Task 1 test subset.

Team ID	Model Variant	F1-score	Rank
SIAT	**BERT-Legal (MPN 15)**	**0.030**	**4**
SIAT	BERT-Legal (MPN 20)	0.029	-
TLIR	**Run 1**	**0.192**	**1**
NM	NM-Run-Task1-BM25	0.094	2
DSSIR	Run-Test-BM25	0.041	3

are added to the cost function:

$$
\begin{aligned}
&-\log p(y|x + r_{adv}; \theta), \\
&where\ r_{adv} = \arg\min_{r, \|r\| < \epsilon} \log p(y|x + r; \hat{\theta}),
\end{aligned} \tag{10}
$$

where $\hat{\theta}$ represents a constant set in parameters of the current classifier, y is the corresponding target label and r represents a perturbation of the input. Using a constant copy $\hat{\theta}$ instead of θ indicates that the back propagation algorithm will not be used to propagate gradients by the construction process of adversarial examples. For each step of training, the worst perturbations r_{adv} against to current model $p(y|x; \hat{\theta})$ in Eq.10 are identified and used to train the model to be robust by minimizing Eq.10. However, we usually cannot calculate this value accurately in practical applications. This is mainly because the precise minimization of r is tricky for many neural networks. FGM [5] further proposed to linearize $p(y|x; \hat{\theta})$ around x in order to approximate this value. With linear approximation and L_2 norm constraint described in the case of Eq.10, the adversarial disturbance is given by

$$
\begin{aligned}
&r_{adv} = -\epsilon \cdot \frac{g}{\|g\|_2}, \\
&where\ g = \nabla_x \log p(y|x; \hat{\theta}).
\end{aligned} \tag{11}
$$

This perturbation make the back propagation is easily to be computed in neural models. By designing a cost function to maximize the adversarial perturbation, we fine-tune the BERT-Legal model with FGM to improve the performance on legal entailment. It can be seen as a novel regularization method of BERT-Legal, which is able to improve the robustness and handle approximately worst case perturbations.

4 EXPERIMENTS AND RESULTS

In our experiments, we split the original training data into two separate partitions. In Task 1, we randomly select 20% of queries with their candidate cases as our development data while the remaining queries along with their candidates are used for training models. For the COLIEE Task 2 dataset, we divide them into two subsets, train and development. The development subset accounts for about 10% of the total, which is used is used to find the best setting of hyperparameters. To conduct fair and reliable comparisons, all experiments are using the same random-split setting and division.

Table 4: Hyperparameter configurations of the experiments in Task 1 and Task 2.

Hyperparameter	Value
Batch Size	8
Training Epoch	5
Max Sequence Length	512
Learning Rate	1e-5
Max Paragraph Number	15, 20

4.1 Experimental Settings

In Task 1, we construct the training dataset with negative sampling strategy, where we make the positive and negative sample number ratio as 1:1. Finally, we obtain a training dataset with 7500 samples. To reduce the training consumption, we limit the max paragraph number in a document to 15 or 20. In Task 2, we also adopt negative sampling and construct a training dataset containing 15,216 candidate paragraphs/query pairs. All of the proposed models are fully differentiable, hence, we can train them in an end-to-end manner. The whole model uses a LAMB optimizer [12] and train on Tesla P100. The hyperparameter configurations are listed in Table 4.

4.2 Baselines

We conduct experiments on the following baseline methods:

- BERT. This model is only trained using BERT. The final hidden state vector of [CLS] is fed to the output layer and get the prediction result.
- BERT-Legal. This model is trained using the dynamic N-gram mask pre-training method, which domain pre-training on thousands of legal document corpus of Task 1 before fine-tuning the model with dataset of Task 2.
- BERT-Legal (+FGM). This model trains the BERT-Legal model with Fast Gradient Method (FGM) adversarial training.

Table 5: The experimental results of our methods on the Task 2 development subset.

Model Variant	Precision	Recall	F1-score	Acc	Auc
BERT	0.662	0.413	0.508	0.975	0.703
BERT-Legal	0.698	0.606	**0.649**	0.980	0.799
BERT-Legal (+FGM)	**0.764**	0.556	0.643	**0.981**	0.775
BERT-Legal (+Data Augment)	0.610	**0.616**	0.613	0.976	**0.802**

Table 6: The final result on Task 2 test set.

Team ID	Model Variant	F1-score	Rank
SIAT	**BERT-Legal**	**0.586**	**4**
SIAT	BERT-Legal (+FGM)	0.567	-
NM	**Run-task2-DebertaT5**	**0.691**	**1**
UA	UA-Reg-Pp	0.627	2
JNLP	BM25-Supporting-Denoising	0.612	3

- BERT-Legal (+Data Augment). This model is trained with domain data augment based on BERT-Legal model. The augment training corpus is the "paragraph pairs" extracted from the "document pairs" in Task 1.

4.3 Experimental Results in Task 1

Table 2 reports the evaluation results on our development dataset in Task 1, the legal case retrieval task. BERT-Legal (MPN 15) indicates adopting the post-training BERT on the legal corpus as a semantic ranking component in the legal case retrieval pipeline and set the max paragraph number (MPN) of a document to 15. Similarly, we test the model performance with BERT-Legal (MPN 20). As the shown results, one conclusion is increasing the processing paragraph number can improve the retrieval performance on the F1-score. We submit two runs on the Task 1 test evaluation. The testing results are listed in Table 3. Due to the large-scale candidate set in the test set, all models can only achieve a relatively poor performance. We note that the difference between our two runs in F1 is less than 0.01. Both of them achieve rather close performance in this task. Compare to our retrieval pipeline mechanism with other BM25 based methods, our semantic ranking model obtains worse results which may be due to the reason that the recall method can not return good candidate cases and hence inject many noisy samples causing the downstream ranking module to make wrong predictions.

4.4 Experimental Results in Task 2

As shown in Table 5, we can see the results on our development dataset in Task 2, the legal case entailment. The BERT model only achieves 0.508 on F1. While the BERT-Legal model surpasses BERT model by 0.141 on F1-socre, which indicates the significant improvement of our BERT-Legal model. Meanwhile, the BERT-Legal (+Data Augment) shows low performance on F1 compared with BERT-Legal, which reflects the limitations of our idea of augmenting data on Task 2 from Task 1. This may be because the data distribution of Task 1 and Task 2 are too different, causing that the addition of this part of the extra data undermines the learning process of the model.

In addition, the experimental results show another interesting phenomenon. According to the results shown in our development dataset, the BERT-Legal (+FGM) combining the BERT-Legal model with the FGM adversarial training appears obviously fluctuations in the metric of Precision, and there is no obvious drop in performance compared with BERT-Legal. However, in the Table 5, we see a significant reduction in performance on the BERT-Legal (+FGM) model according the official testing dataset. This shows that there is a difference on the data distribution between the development dataset obtained by our random division and the official testing dataset, which also shows a place we can further improve.

5 CONCLUSION AND FUTURE WORK

In this paper, we describe our proposed methods on two legal tasks of COLIEE-2021, including the legal case retrieval task and legal case entailment task. We attempt both statistics features and contextual semantic understanding methods to solve these two tasks. In the legal case retrieval task (Task 1), the BERT-Legal based semantic ranking model can achieve a good performance in our development dataset. However, this method obtains poor performance on the online testing data. The reason may be the statistics featured based recall method can not return high-quality candidate cases causing a big challenge for the following downstream ranking module to provide a precise prediction. In the final test results, most of the semantic based methods get a low F1-score even worse than the statistics based methods (i.e. BM25). In the legal case entailment task (Task 2), we propose a post-training strategy with dynamic n-gram masking where a legal domain pre-trained BERT model (i.e. BERT-Legal) is obtained. We attempt to enhance the representation ability of BERT-Legal with FGM and data augmentation tricks. The testing results demonstrate the efficiency of our dynamic n-gram post-training strategy. In summary, we rank 4th in both tasks, while still contains some performance gap with other competitive teams.

Some attempts have conducted to tackle the challenges behind the legal case retrieval and entailment tasks, there still have some limitations existing in our work and many unexplored methods. We would like to list them here for supporting future research or application. As we mentioned above, the traditional statistics features based recall method (i.e. BM25) is not good at retrieving large-scale legal cases to provide the downstream ranking module with reliable prior knowledge. To solve this problem, a possible direction is to fuse reasonable semantic features into the recall module at the early stage, such as dense-sparse phrase index [9] for real-time open-Domain question answering. Although we train a legal domain BERT model to obtain a better language representation ability, the max sequence length still restricts the language understanding performance of long legal documents. Utilizing a long context transformer to pre-trained the language model may be a possible direction to mitigate this problem.

REFERENCES

[1] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 126–134.

[2] Barbara J Grosz, Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. (1995).

[3] Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226* (2018).

[4] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).

[5] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* (2016).

[6] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. arXiv:2011.08071 [cs.CL]

[7] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2019. Combining Similarity and Transformer Methods for Case Law Entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL 2019, Montreal, QC, Canada, June 17-21, 2019*. ACM, 290–296. https://doi.org/10.1145/3322640.3326741

[8] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[9] Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur P Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. *arXiv preprint arXiv:1906.05807* (2019).

[10] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. THUIR@COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment. arXiv:2012.13102 [cs.IR]

[11] Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 275–282.

[12] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962* (2019).

Retrieving Legal Cases from a Large-scale Candidate Corpus

Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu*, Min Zhang, Shaoping Ma
yiqunliu@tsinghua.edu.cn
Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China

ABSTRACT

This paper presents approaches employed in COLIEE 2021 Task 1, a legal case retrieval task that aims to retrieve all noticed cases given a large-scale candidate case corpus. Of all two methods, the first method is a traditional language model for information retrieval (IR) and the second is a neural-based method named refined BERT-PLI. In addition, we design a filter to remove unreasonable candidates from the result list. The official Task 1 results show that our Run 1 has the best performance of all 15 runs and is significantly better than the second-place method. Besides, all of our three runs have a top-5 performance.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

legal case retrieval, language model, neural-based method

ACM Reference Format:
Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu*, Min Zhang, Shaoping Ma. 2021. Retrieving Legal Cases from a Large-scale Candidate Corpus. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Legal case retrieval is of vital significance to the legal domain. Under different law systems, a relevant case can be directly or indirectly involved in the final decision as a crucial reference for judges. Lawyers or judges used to manually search for previous cases as supporting materials of the case in trial. However, as the number of precedents continues to grow, it is time-consuming to manually collect relevant cases. Hopefully, with the development of information retrieval (IR), adopting IR methods to automatically retrieve legal information in need, especially in the legal case retrieval domain, has currently received increasing attention. An efficient method proposed for the relevant case retrieval task can alleviate the heavy material preparation work. Therefore, competitions and evaluations [1, 2] are held to promote such methods used in AI & Law.

As one of the popular competitions, the Competition on Legal Information Extraction/Entailment (COLIEE) [4] is an evaluation competition held annually to develop IR and document entailment methods in the legal domain since 2014. In the latest COLIEE 2021, there are in total five tasks, among which Task 1 is a legal case retrieval task, Task 2 is a legal case entailment task, Task 3 is a statute law retrieval task, Task 4 is a legal textual entailment task, and Task 5 is a legal question answering (QA) task. Both Task 1 and Task 2 are based on a database of predominantly Federal Court of Canada case laws provided by Compass Law, while Task 3 and Task 4 are based on Japanese legal bar exams. This year, our team, Tsinghua Legal Information Retrieval (TLIR) participates in task1. Notably, compared with previous COLIEE competitions that each query of Task 1 only contains 200 candidate cases, all queries of Task 1 in COLIEE 2021 share a 4415-case candidate pool, which significantly increases the task difficulty. In total, COLIEE 2021 Task 1 received 15 submissions from seven teams.

In this paper, we mainly introduce our approaches corresponding to three runs of Task 1. For the approach corresponding to Run 1, we first adopt a series of data mining methods to clean the raw dataset and collect a word-level corpus. Then the corpus is used to train a language model for IR. Finally, all top-scoring outputs are passed through a filter to get our final results. Run 2 and Run 3 are all results of a refined BERT-PLI [11]. For the approach corresponding to Run 2 and Run 3, we first use the first approach to sample top-30 relevant cases from the candidate pool. Then, unlike the BERT-PLI in COLIEE 2020, we only adopt part of paragraphs of a query case for training leveraging both the final performance and the expansion of the Task 1 dataset this year. As a result, the placements of our three runs in Task 1 are: **1st place** (Run 1), 3rd place (Run 3), and 5th place (Run 2). The evaluation F1 score of our Run 1 is more than twice the F1 score of the 2nd place run.

2 TASK OVERVIEW

2.1 Task 1 Description

Task 1 is a legal case retrieval task related to a database of predominantly Federal Court of Canada case laws provided by Compass Law. Given a query case Q, the target is to extract all supporting cases $S = \{S_1, S_2, ...S_n\}$ from the entire case law corpus. The supporting case, which is also called 'noticed cases', denotes precedents that can support the judgment for the query case Q. There are in total 650 query cases with noticed case labels as the training set and 250 query cases without noticed case labels as the test set.

2.2 Data Corpus

The dataset of Task 1 is drawn from an existing collection of predominantly Federal Court of Canada case law. Statistics of the dataset are shown in Table 1. All query cases share a large-scale candidate

*Corresponding author

COLIEE 2021, June 21, 2021, Online

Table 1: Dataset statistics of COLIEE Task 1.

Statistic	Training	Testing
# queries	650	250
# candidate cases	4415	4415
# noticed cases per query	5.09 (0.12%)	3.60 (0.08%)

pool with 4415 case documents in total, and even the queries themselves are sampled from such a pool. By comparison, each query in previous COLIEE Task 1 has a similar number of noticed cases to retrieve from a much smaller independent candidate pool (e.g., 200 candidates per query in COLIEE 2020). Therefore, retrieving noticed cases this year is more challenging.

2.3 Evaluation Metrics

The evaluation metrics of Task 1 are precision, recall and F1 score. Definition of these measures are as follows:

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

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

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

where $\#TP$ is the number of correctly retrieved cases for all queries, $\#FP$ is the number of falsely retrieved cases for all queries, and $\#FN$ is the number of missing noticed cases for all queries.

3 METHODS

3.1 Run 1: Language Models for IR

Traditional retrieval models such as BM25 [8], TF-IDF [9], and Language Model for Information Retrieval (LMIR) [6] rank candidates by statistical probabilistic framework based on the bag-of-words representations. Shao et al. [11] demonstrate that traditional retrieval methods has competitive results in legal case retrieval. Therefore, in COLIEE 2021, we choose traditional retrieval models as our first run for Task 1.

The original case document mainly has two types of structures: headings and paragraphs. Headings include title, court, date, summary, and other information. Paragraphs with a number tag at the beginning are the main content of the case. Considering that most information in headings is irrelevant to the retrieval task, we only use paragraphs for both of our approaches in this competition.

The first step is data pre-processing. Since the data is drawn from Canada law, some of the case documents contain French text. After analysis, most of these French parts are the translation of the English statement in the same document. Therefore, removing such French text from their belonging documents does not influence the overall information obtained in documents. In this paper, we adopt Langdetect [12] to remove all French paragraphs. Then, we convert all English letters to lowercase. Finally, we apply Nltk [5] to split words, remove stopwords and punctuation, and stem. After pre-processing, we get all cases in the form of tokens. All tokens are gathered together as the training corpus C for traditional retrieval models.

In this competition, we adopt LMIR as the Run 1 model, where computing the relevance between candidates and queries is considered a query generation process. In other words, given a candidate case c, the probability of generating the correct query q $P(q \mid c)$ is denoted as:

$$P(q \mid c) \propto \prod_{t \in q} P(t \mid c) \tag{4}$$

where t is the token in a query generated from data pre-processing step and $P(t \mid c)$ represents the probability of generating token t given c. For LMI, there are multiple methods to estimate $P(t \mid c)$. In this paper, we choose the linear interpolation language model and $P(t \mid c)$ is defined as:

$$P(t \mid c) = \lambda P_{ml}(t \mid M_c) + (1 - \lambda) P_C(t \mid M_C) \tag{5}$$

where $P_{ml}(t \mid M_c)$ denotes token probability in case document c, $P_C(t \mid M_C)$ denotes token probability in the whole training corpus C, and λ is a smoothing parameter ranging from 0 to 1.

When computing the relevance score between queries and candidate cases, instead of taking all tokens of a query as the input q of LMIR, we only identify tokens from paragraphs that are more likely to cite noticed cases. Specifically, according to the data format, sentences with a placeholder such as 'FRAGMANT_SUPPRESSED' or 'REFERENCE_SUPPRESSED' etc. are citations or references from other noticed cases. These sentences are directly relevant to a noticed case. Furthermore, considering the context of the reference sentence may also have a connection with noticed cases, we take all tokens from paragraphs to which a placeholder belongs as our input q of LMIR.

Although our LMIR model is already able to output a ranking list given query input q, there are still regulations that can filter some unreasonable candidates. In this competition, we design a two-step candidate filter. First, a query case can only cite cases judged before the query case itself. Therefore, we extract dates from cases containing time information. Besides the trial date, some case documents also record other dates such as its previous judgment date or date of the case. To avoid mistakenly filtering noticed cases, we define our first regulation as:

$$Rank_1 = \{c \mid c \in Rank_0 \wedge max(dates(c)) \leq min(dates(q))\} \tag{6}$$

where $dates(d)$ is the set of dates appeared in document d and $Rank_0$ is the original output of LMIR.

In addition, we also find 222 document pairs describing the same case but have different document ids. For example, document '008447.txt' and document '089987.txt'. As one case never cites itself as a noticed case, we remove such documents from $Rank_1$ to get our final rank list $Rank_2$. The final submission consists of top-K cases from $Rank_2$ for each query. K is a hyperparameter.

3.2 Run 2 & Run 3: Refined BERT-PLI

Shao et al. [11] propose BERT-PLI to tackle challenges in legal case retrieval scene that case documents have extended length and complex structure. This method divides a document into paragraph

Table 2: Recall rate of the top-30, top-50, and top-100 scored candidate cases using different traditional retrieval models. Avg. Rank is the average rankings of all noticed cases.

Method	30	50	100	Avg. Rank
TF-IDF	0.333	0.420	0.560	332.0
BM25	0.373	0.463	0.583	384.8
LMIR	**0.437**	**0.537**	**0.660**	**243.1**

level and computes interactions between paragraphs using BERT [3]. Compared with other neural models, BERT-PLI can take long-text representation as an input without cutting off long documents in the middle. Previous COLIEE Task 1 results [10] illustrate that BERT-PLI has competitive performance. Thus, Run 2 and Run 3 are mainly based on BERT-PLI but have some revisions according to the feature of COLIEE 2021 dataset.

The overall model structure is shown in Figure 1. In general, the model consists of three stages. In Stage 1, we first sample top-N candidates from the whole candidate pool by traditional retrieval models. In order to choose a traditional retrieval model with a better recall, we first conduct a pre-experiment between TF-IDF [9], BM25 [8], and LMIR[6]. We compute the overall recall rate and average rankings of all noticed cases in the training set. As shown in Table 2, LMIR has both the highest recall and lowest average rankings. Therefore, we adopt LMIR to sample candidates. In practice, we sample all top-30 candidates and other noticed cases ranking more than 30 to be the training set for BERT-PLI.

In Stage 2, we fine-tune the BERT [3] with a case-entailment dataset in COLIEE 2019 [7] Task2 which aims to identify paragraphs entailing the decision paragraph in a document. The fine-tuning process is handled on a next sentence prediction task. Composed of a decision paragraph and a candidate paragraph separated, the input sentence pair is separated by a [SEP] token and appended by [CLS] token. Vectors from the final hidden layer are fed into a classification layer to get the final prediction.

In Stage 3, the original idea in Shao et al. [11] computes interactions between all query paragraphs and all top-30 candidate case paragraphs. This strategy may be practical for a test set within 200 candidates, but this year the size of the candidate pool is over 20 times larger than before. As shown in Table 2, the top-30 recall is only less than 50 percent which is far below our expectation. Therefore, in this competition, we take top-200 candidates into consideration. Following the idea in Run 1, we only compute the interactions between query paragraphs with a citation token and all top-30 candidate paragraphs (top-200 candidate paragraphs for test set). We apply BERT to infer semantic connections between these paragraph pairs and generate a paragraph-level interaction map. Then, the map is fed into a max-pooling layer to generate the most representative value for each query paragraph. Finally, an RNN combined with an attention layer is utilized to encode a paragraph-level sequential feature p_{qk} into a document-level feature d_{qk}, where q and k represent the query and candidate respectively. Finally, d_{qk} is passed through a fully connected layer to output a two-dimensional prediction vector l.

Table 3: Results of LMIR ($\lambda = 0.95$) on the training set with different hyperparameter K.

K	Precision	Recall	F1
5	**0.1809**	0.1776	0.1792
6	0.1718	0.2024	**0.1858**
7	0.1596	0.2193	0.1847
8	0.1508	0.2368	0.1842
9	0.1429	0.2525	0.1825
10	0.1349	**0.2649**	0.1788

Similar to Run 1, we also apply a filter to our model. The first two steps are exactly the same as Run 1 filter. An additional filter step is adopted due to the fact that our model tends to overestimate the relevance between queries and all candidates. Specifically, suppose l is the prediction vector of a single candidate and L is a list that contains all l predicted as noticed. Then, the whole list L is fed into a filter function f to get the final prediction y. The filter function f is defined as:

$$f(L) = \begin{cases} \arg\max_{l} \ Softmax(l)[1] & |L| = 0 \\ L & 0 < |L| \leq 10 \\ \{l \mid l \in L \ \wedge \ Softmax(l)[1] > T\} & |L| > 10 \end{cases} \quad (7)$$

where $Softmax(l)$ is a two-dimensional vector generated by a Softmax layer and $Softmax(l)[1]$ denotes the value on the second dimension which is the probability of a positive (noticed) prediction. T is the threshold to control the number of overall cases predicted as noticed.

4 EXPERIMENTS

For Run 1, we set the hyperparameter λ to be 0.95 based on a comparison between the performances with different λ values. $\lambda = 0.95$ achieving the best performance illustrates that when the query has a short text length (only paragraphs with a token in Run 1) and the candidate pool has a relatively large size, $P_{ml}(t \mid M_c)$ plays a more important role in determining the relevance between queries and candidates. In other words, term frequencies in the query are decisive to retrieve noticed cases in Task 1 this year. Another key hyperparameter that needs to be determined before further experiments is K, the number of retrieved cases per query. According to the evaluation results on the training set shown in 3, we set K to be 6. For Run 2 and Run 3, we set the maximum input query paragraph number to be 30 and the maximum input candidate paragraph number to be 40. The rest of some important hyperparameters are as follows: learning rate = $5e-4$ for both runs, weight decay = $1e-6$ for Run 2 and 0 for Run 3, threshold T = 0.83 for Run 2 and 0.56 for Run 3. On the validation set, Run 2 has a more balanced performance (precision = 0.3171, recall = 0.3212, F1 = 0.3191), while Run 3 mainly focuses on precision (precision = 0.3785, recall = 0.1223, F1 = 0.1848). We train a model like Run 3 because the training set (top-30 candidate + other noticed case) has a higher ratio of noticed cases to all cases than the test set (0.08%). Therefore, controlling recall and improving precision on the validation set can have a better F1 score on the test set.

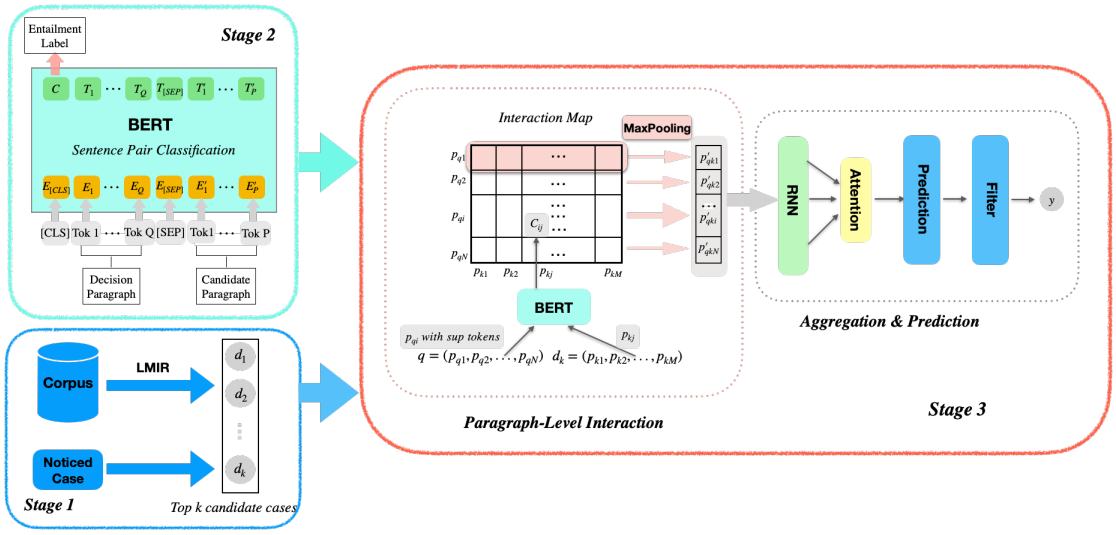

Figure 1: The overall structure of refined BERT-PLI.

Table 4: Final top-5 results of Task 1 on the test set.

Team	Precision	Recall	F1 (official)	Rank
TLIR (Run1)	0.1533	0.2556	**0.1917**	1
NM	-	-	0.0937	2
TLIR (Run3)	0.0350	0.0656	0.0456	3
DSSIR	-	-	0.0411	4
TLIR (Run2)	0.0259	0.0456	0.0330	5

The final top-5 results of COLIEE Task 1 are illustrated in Table 4. The organizers only publish F1 scores, and we further evaluated our three runs by precision and recall after the test set labels are released. Of all 15 runs, our Run 1 has the best F1 score and significantly outperforms other runs. Besides, Run 3 has the third placement and Run 2 has the fifth placement. From the results above, we can conclude that while in previous COLIEE Task1, neural methods have a slightly better performance than traditional retrieval models [7], this year traditional retrieval models (rank 1, 2) outperforms neural methods. Therefore, traditional retrieval models are robust and still competitive in the legal search domain, especially when the candidate pool size is relatively large (e.g. 4415).

5 CONCLUSION AND DISCUSSION

In this paper, we presented two retrieval methods for the legal case retrieval task in COLIEE 2021. For the first approach, we utilize LMIR and design a filter to remove unreasonable candidates from the result list. For the second approach, we refine a competitive neural method BERT-PLI and also design a filter to control positive predictions. Competition results show that Run 1 has the best performance of all runs and is significantly better than the second-place method. In addition, all of our three runs have a top-5 performance.

On the other hand, as the size of the candidate case pool per query is changed from 200 to 4415 this year, Task 1 in COLIEE 2021 becomes more challenging than previous legal case retrieval tasks in COLIEE. Consequently, the overall performances of Task 1 this year decrease to a large extent. In addition to the pool size, there are other reasons for this decline: First, as mentioned in Section 3.1, there exist some documents describing the same documents. If such document pairs are query documents, their noticed cases can even be totally different. For example, documents '067501.txt' and '030050.txt' are about the same case, but the noticed cases of '067501.txt' are '038025.txt' and '072553.txt' while the noticed case of '030050.txt' is '028189.txt'.

The second possible explanation is that with the growth of candidate case number, the ratio of noticed cases to top-k scored candidate cases decreases if we still use semantic-based or term-level methods to retrieve legal cases. In other words, existing methods do not effectively support large-scale legal case retrieval. Therefore, future works need to explore method which can utilize more than semantic or term-level information in legal documents.

REFERENCES

[1] Jason R Baron, David D Lewis, and Douglas W Oard. 2006. TREC 2006 Legal Track Overview.. In *TREC*. Citeseer.
[2] Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Overview of the FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance.. In *FIRE (Working Notes)*. 1–12.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[4] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In *JSAI International Symposium on Artificial Intelligence*. Springer, 177–192.
[5] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).
[6] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 275–281.
[7] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2019. A Summary of the COLIEE 2019 Competition. In *JSAI International Symposium on Artificial Intelligence*. Springer, 34–49.

[8] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.
[9] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.
[10] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. THUIR@ COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment. *arXiv preprint arXiv:2012.13102* (2020).
[11] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20.* 3501–3507.
[12] Nakatani Shuyo. 2010. Language detection library for java. *Retrieved Jul* 7 (2010), 2016.

Yes, BM25 is a Strong Baseline for Legal Case Retrieval

Guilherme Moraes Rosa
NeuralMind
University of Campinas (Unicamp)

Ruan Chaves Rodrigues
NeuralMind
Federal University of Goiás (UFG)

Roberto de Alencar Lotufo
NeuralMind
University of Campinas (Unicamp)

Rodrigo Nogueira
NeuralMind
University of Campinas (Unicamp)
University of Waterloo

ABSTRACT

We describe our single submission to task 1 of COLIEE 2021. Our vanilla BM25 got second place, well above the median of submissions. Code is available at https://github.com/neuralmind-ai/coliee.

ACM Reference Format:
Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 3 pages.

1 INTRODUCTION

The Competition on Legal Information Extraction/Entailment (COLIEE) [8, 9, 14, 15] is an annual competition to evaluate automatic systems on case and statute law tasks.

In this paper, we describe our submission to the legal case retrieval task of COLIEE 2021. The goal of this task is to explore and evaluate the performance of legal document retrieval technologies. It consists of retrieving from a corpus the cases that support or are relevant to the decision of a new case. These relevant cases are referred to as "noticed cases".

2 RELATED WORK

Some successful NLP approaches to the legal domain use a combination of data-driven methods and hand-crafted rules [20]. For example, in task 1 of COLIEE 2019, Gain et al. [6] used a combination of techniques, such as Doc2Vec and BM25. Leburu-Dingalo et al. [10] used a learning to rank approach with features generated from models such as BM25 and TF-IDF. For task 1 of COLIEE 2020, Mandal et al. [12] applied filtered-bag-of-ngrams and BM25.

Gomes and Ladeira [7] compared TF-IDF, BM25 and Word2Vec models for jurisprudence retrieval. The results indicated that the Word2Vec Skip-Gram model trained on a specialized legal corpus and BM25 yield similar performance. Althammer et al. [1] investigate BERT [5] for document retrieval in the patent domain and found that BERT model does not yet achieve performance improvements for patent document retrieval compared to the BM25 baseline.

COLIEE 2021, June 21, 2021, Online

Pradeep et al. [13] showed that BM25 is above the median of competition submissions in TREC 2020 Health Misinformation and Precision Medicine Tracks.

3 THE TASK

The dataset for task 1 is composed of predominantly Federal Court of Canada case laws, and it is provided as a pool of cases containing 4415 documents. The input is an unseen legal case, and the output is the relevant cases extracted from the pool that support the decision of the input case. The training set includes 650 query cases and 3311 relevant cases with an average of 5.094 labels per example. In the test set, only the query cases are given, 250 documents in total. We also show the statistics of this dataset in Table 1.

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

$$\mathrm{F1} = (2 \times P \times R)/(P + R), \tag{1}$$

where P is the number of correctly retrieved cases for all queries divided by the number of retrieved cases for all queries, and R is the number of correctly retrieved cases for all queries divided by the number of relevant cases for all queries.

	Train	Test
Number of base cases	650	250
Number of candidate cases	4415	4415
Number of relevant cases	3311	900
Avg. relevant cases per base case	5.1	3.6

Table 1: COLIEE 2021 task 1 data statistics.

4 OUR METHOD: BM25

BM25 [4, 17] is an algorithm developed in the 1990s based on a probabilistic interpretation of how terms contribute to the relevance of a document and uses easily computed statistical properties such as functions of term frequencies, document frequencies and document lengths. The algorithm is a weighting scheme in the vector space model characterized as unsupervised, although it contains the free parameters k_1 and b that can be tuned to improve results.

BM25 score between a query q and a document d is derived from a sum of contributions from each query term that appears in the document and it is defined as:

$$\text{BM25}(q,d) = \sum_{t \in q \cap d} \log \frac{N - \text{df}(t) + 0.5}{\text{df}(t) + 0.5} \cdot \frac{\text{tf}(t,d) \cdot (k_1 + 1)}{\text{tf}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{l_d}{L}\right)} \quad (2)$$

The first part of the equation (the log term) is the inverse document frequency (idf): N is the total number of documents in the corpus, and $df(t)$ refers to the document frequency or the number of documents that term t appears. In the second part, $tf(t,d)$ represents the number of times term t appears in document d or its term frequency. The denominator performs length normalization since collections usually have documents with different lengths. ld refers to the length of document d while L is the average document length across all documents in the collection. As said before, k_1 and b are free parameters.

Until today, BM25 still provides competitive performance in comparison with modern approaches on text ranking tasks.

We use BM25 from Pyserini, which is a Python library designed to help research in the field of information retrieval. It includes sparse and dense representations [11]. Pyserini was created to provide easy-to-use information retrieval systems that could be combined in a multi-stage ranking architecture in an efficient and reproducible manner. The library is self-contained as a standard Python package and comes with queries, pre-built indexes, relevance judgments, and evaluation scripts for many used IR test collections such as MS MARCO [2], TREC [13, 16, 19] and more. In this work, we use retrieval with sparse representations and it is provided via integration with Anserini [18], which is built on Lucene [3].

To apply BM25 to task 1, we first index all base and candidate cases present in the dataset. Before indexing, we segment each document into segments of texts using a context window of 10 sentences with overlapping strides of 5 sentences. We refer to these segments as candidate case segments.

In task 1, queries are base cases, which are also long documents. In our experiments, we found that using shorter queries improves efficiency and effectiveness. Thus, we apply to the base cases the same segmentation procedure described during the indexing step, creating, as we refer to, base case segments. We then use BM25 to retrieve candidate case segments for each base case segment. We denote $s(b_i, c_j)$ as the BM25 score of the i-th segment of the base case b and the j-th segment of the candidate case c.

The relevance score $s(b,c)$ for a (base case, candidate case) pair is the maximum score among all their base case segment and candidate case segment pairs:

$$s(b,c) = \max_{i,j} s(b_i, c_j) \quad (3)$$

We then rank the candidates of each base case according to these relevance scores and use the method described in Section 4.1 to select the candidate cases that will comprise our final answer.

Due to the large number of segments produced from base cases, retrieving the base cases of the test set takes more than 24 hours on a 4-core machine. Thus, we also evaluate our system using only the first N segments. Table 2 summarizes our three best hyperparameters. The models are named using the format BM25-(N, window size, stride). We achieve the best result using all base case segments, a window size of 10 sentences, and a stride of 5 sentences. However, due to the high computational cost of scoring all segments, our submitted system uses only the first 25 windows of each base case, i.e., $N = 25$.

Method	F1	Precision	Recall
BM25-(10, 10, 5)	0.1040	0.0785	0.1560
BM25-(25, 10, 10)	0.1203	0.0997	0.1516
BM25-(All, 10, 5)	0.1386	0.1027	0.2134

Table 2: Task 1 results on the 2021 dev set.

4.1 Answer Selection

Given a base case b, BM25 estimates a relevance score $s(b,c)$ for each candidate case c retrieved from the corpus using the method explained above. To select the final set of candidate cases, we apply three rules:

- Select candidates whose relevance scores are above a threshold α;
- Select the top β candidate cases with respect to their relevance scores;
- Select candidate cases whose scores are at least γ of the highest relevance score.

We use an exhaustive grid search to find the best values for α, β, γ on the first 100 examples of the 2021 training dataset. We swept $\alpha = [0, 0.1, ..., 0.9]$, $\beta = [1, 5..., 200]$, and $\gamma = [0, 0.1, ..., 0.9, 0.95, 0.99, 0.995, ..., 0.9999]$.

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

5 RESULTS

Results	F1	Precision	Recall
Median of submissions	0.0279	-	-
3rd best submission of 2021	0.0456	-	-
Best submission of 2021	0.1917	-	-
BM25 (ours)	0.0937	0.0729	0.1311

Table 3: Task 1 results on the 2021 test set.

Results are shown in Table 3. Our vanilla BM25 is a good baseline for the task as it achieves second place in the competition and its F1 score is well above the median of submissions. This result is not a surprise since it agrees with results from other competitions, such as the Health Misinformation and Precision Medicine tracks of TREC 2020 [13]. The advantage of our approach is the simplicity of our method, requiring only the document's segmentation and the grid search. One of the disadvantages is the time spent during the retrieval of segmented documents.

6 CONCLUSION

We showed that our simple BM25 approach is a strong baseline for the legal case retrieval task.

REFERENCES

[1] Sophia Althammer, Sebastian Hofstätter, and Allan Hanbury. 2020. Cross-domain Retrieval in the Legal and Patent Domains: a Reproducibility Study. *arXiv preprint arXiv:2012.11405* (2020).

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3* (2018).

[3] Andrzej Białecki, Robert Muir, and Grant Ingersoll. 2012. Apache Lucene 4. *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval* (2012).

[4] F. Crestani, M. Lalmas, C. J. van Rijsbergen, and I. Campbell. 1999. Is this document relevant?... probably: A survey of probabilistic models in information retrieval. *ACM Computing Surveys, 30(4):528–552* (1999).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[6] B. Gain, D. Bandyopadhyay, T. Saikh, and A Ekbal. 2019. IITP@COLIEE 2019: legal information retrieval using BM25 and BERT. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE 2019* (2019).

[7] Thiago Gomes and Marcelo Ladeira. 2020. A new conceptual framework for enhancing legal information retrieval at the Brazilian Superior Court of Justice. *MEDES '20: Proceedings of the 12th International Conference on Management of Digital EcoSystems* (2020).

[8] Yoshinobu Kano, M. Kim, R. Goebel, and K. Satoh. 2017. Overview of COLIEE 2017. In *COLIEE 2017 (EPiC Series in Computing, vol. 47)*. 1–8.

[9] Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. COLIEE-2018: Evaluation of the competition on legal information extraction and entailment. In *JSAI International Symposium on Artificial Intelligence*. 177–192.

[10] T. Leburu-Dingalo, E. Thuma, N. Motlogelwa, and M Mudongo. 2020. Ub Botswana at COLIEE 2020 case law retrieval. *COLIEE (2020)* (2020).

[11] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations. *arXiv preprint arXiv:2102.10073* (2021).

[12] A. Mandal, S. Ghosh, K. Ghosh, and S. Mandal. 2020. Significance of textual representation in legal case retrieval and entailment. *COLIEE (2020)* (2020).

[13] Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. [n.d.]. H2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. *Corpus* 5, d3 ([n. d.]), d2.

[14] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2019. A Summary of the COLIEE 2019 Competition. In *JSAI International Symposium on Artificial Intelligence*. 34–49.

[15] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. (2020).

[16] Kirk Roberts, Dina Demner-Fushman, E. Voorhees, W. Hersh, Steven Bedrick, Alexander J. Lazar, and S. Pant. 2019. Overview of the TREC 2019 Precision Medicine Track. *The ... text REtrieval conference : TREC. Text REtrieval Conference* 26 (2019).

[17] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. *Proceedings of the 3rd Text REtrieval Conference (TREC-3), pages 109–126, Gaithersburg, Maryland* (1994).

[18] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. *SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pages 1253–1256* (2017).

[19] Edwin Zhang, Nikhil Gupta, Rodrigo Nogueira, Kyunghyun Cho, and Jimmy Lin. 2020. Rapidly Deploying a Neural Search Engine for the COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

[20] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. *arXiv:2004.12158* (2020).

JNLP Team: Deep Learning Approaches for Legal Processing Tasks in COLIEE 2021

Ha-Thanh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Phuong Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Thi-Hai-Yen Vuong
University of Engineering and Technology, VNU
Hanoi, Vietnam

Quan Minh Bui
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Chau Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Binh Tran Dang
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Vu Tran
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

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

Ken Satoh
National Institute of Informatics
Tokyo, Japan

ABSTRACT

COLIEE is an annual competition in automatic computerized legal text processing. Automatic legal document processing is an ambitious goal, and the structure and semantics of the law are often far more complex than everyday language. In this article, we survey and report our methods and experimental results in using deep learning in legal document processing. The results show the difficulties as well as potentials in this family of approaches.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Applied computing** → *Law*.

KEYWORDS

Deep Learning, Legal Text Processing, JNLP Team

1 INTRODUCTION

COLIEE is an annual competition in automatic legal text processing. The competition uses two main types of data: case law and statute law. The tasks for automated models include: retrieval, entailment, and question answering. With deep learning models, JNLP team achieves competitive results in COLIEE-2021.

Task 1 is a case law retrieval problem. With a given case law, the model needs to extract the cases that support it. This is an important problem in practice. It is actually used in the attorney's litigation as well as the court's decision-making. Task 2 also uses caselaw data, though, the models need to find the paragraphs in the existing cases that entail the decision of a given case. Task 3, 4, 5 uses statute law data with challenges of retrieval, entailment, and question answering, respectively.

For the traditional deep learning approach, the amount of data provided by the organizer is difficult for constructing effective models. For that reason, we use pretrained models from problems that have much more data and then finetune them for the current task. In Tasks 1, 2, 3, and 4 we have used lexical score and semantic score to filter a correct candidate. In our experiments, the ratio is not 50:50 for lexical score and semantic score, we find out that the increase in the rate of lexical score lead to efficiency in ranking candidates.

Through problem analysis, we propose solutions using deep learning. We also conducted detailed experiments to explore and evaluate our approaches. Our proposals of deep learning methods for these tasks can be a useful reference for researchers and engineers in automated legal document processing.

2 RELATED WORKS

2.1 Case Law

In COLIEE 2018, most teams chose the lexical-based approaches. UBIRLED ranked the candidate cases based on tf-idf. They got approximately 25% of the candidate cases with the highest scores. UA and several teams also chose the same approach with UBIRLED. They compared lexical features between a given base case and corresponding candidate cases. JNLP team combined lexical matching and deep learning, which achieved state-of-the-art performance on Task 1 with the F1 score of 0.6545.

In COLIEE 2019, several teams applied machine learning including deep learning to both tasks. JNLP team achieved the best result of Task 1 in COLIEE 2019 [13] using an approach similar to theirs in COLIEE 2018. In Task 2, their deep learning approach achieved lower performance compared to their lexical approach. Team UA's combination of lexical similarity and BERT model achieved the best performance for Task 2 in COLIEE 2019 [8].

In COLIEE 2020, the Transformer model and its modified versions were widely used. TLIR and JNLP [7] teams used them to classify candidate cases with two labels (support/non-support) in Task 1. Team cyber encoded candidate cases and the base case in tf-idf space and used SVM to classify. They demonstrated the ability of the approach with the first rank in Task 1. In Task 2, JNLP [7] continually applied the same approach in Task 1 in the weakly-labeled dataset. It made them surpass team cyber and won Task 2.

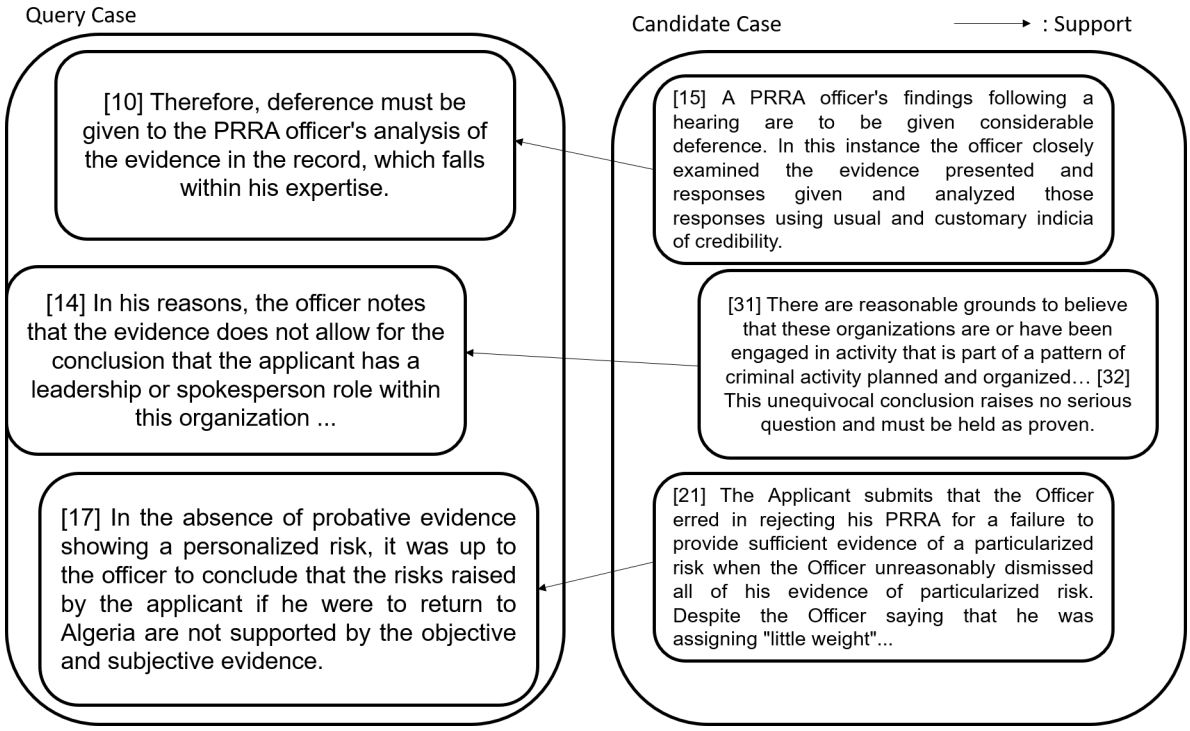

Figure 1: Supporting Definition.

2.2 Statute Law

The retrieval task (Task 3) is often considered as a ranking problem with similarity features. In COLIEE 2019, most of the teams used lexical methods for calculating the relevant scores. JNLP [12], DBSE [11] and IITP [1] chose tf-idf and BM25 to build their models. JNLP used tf-idf of noun phrase and verb phrase as keywords which show the meaning of statements in the cosine-similarity equation. DBSE applied BM25 and Word2vec to encode statements and articles. The document embedding was used to calculate and rank the similarity score. Team KIS [9] represented the article and query as a vector by generating a document embedding. Keywords were selected by tf-idf and assigned with high weights in the embedding process. In COLIEE 2020, with the popularity of Transformer based methods, the participants change their approaches. The task winner, LLNTU, only used BERT model to classify articles as relevant or not.

Regarding the entailment task (Task 4), approaches using deep learning have attracted more attention. In COLIEE 2019, KIS [10] used predicate-argument structure to evaluate similarity. IITP [1] and TR [4] applied BERT for this task. JNLP [5] classified each query to follow binary classification based on big data. In COLIEE 2020, BERT and multiple modified versions of BERT were used. JNLP [7] chose a pretrained BERT model on a large legal corpus to predict the correctness of statements.

3 METHOD

3.1 Task 1 and Task 2. Case Law Processing

In 2021, COLIEE has changed a lot in the data structure of Task 1, and this modification made this task more challenging. In the last year, for each given query, there are only about 200 candidate cases to search for the relevant case law. But in this year, for a given query, we have to search over 4000 candidate cases for relevant cases. Because of the competition's increment of difficulty, the performance of participants' models drops badly from nearly 70% to 20% or worse. The particular reason for this circumstance is the huge searching space. For tackling this issue, we firstly used a lexical model to calculate the similarity between a given query and the whole case law corpus. Limiting the searching space from 4000 candidate cases to 100 candidate cases by picking the top 100 cases that have the highest BM25 scores for a given query.

The previous approaches focus on encoding the whole case law and calculating the similarity of vector representation for each query-candidate pair of cases. In our method, we handle the similarity between candidates on the paragraph level. "Supporting" and "Relevant" are subtractive definitions, and we assume that for each paragraph in the query case, there are one or more paragraphs that carry useful information for the given query's paragraph. As we can see in the Figure 1, paragraph [10] has *"PRRA officer's"* and in the candidate case number [15] also has *"PRRA officer's"*, there seems to be a lexical relationship here. This is evidence that the lexical model could be effective. Along the line of the query paragraph number [14] has *"the evidence"* corresponding to *reasonable grounds*

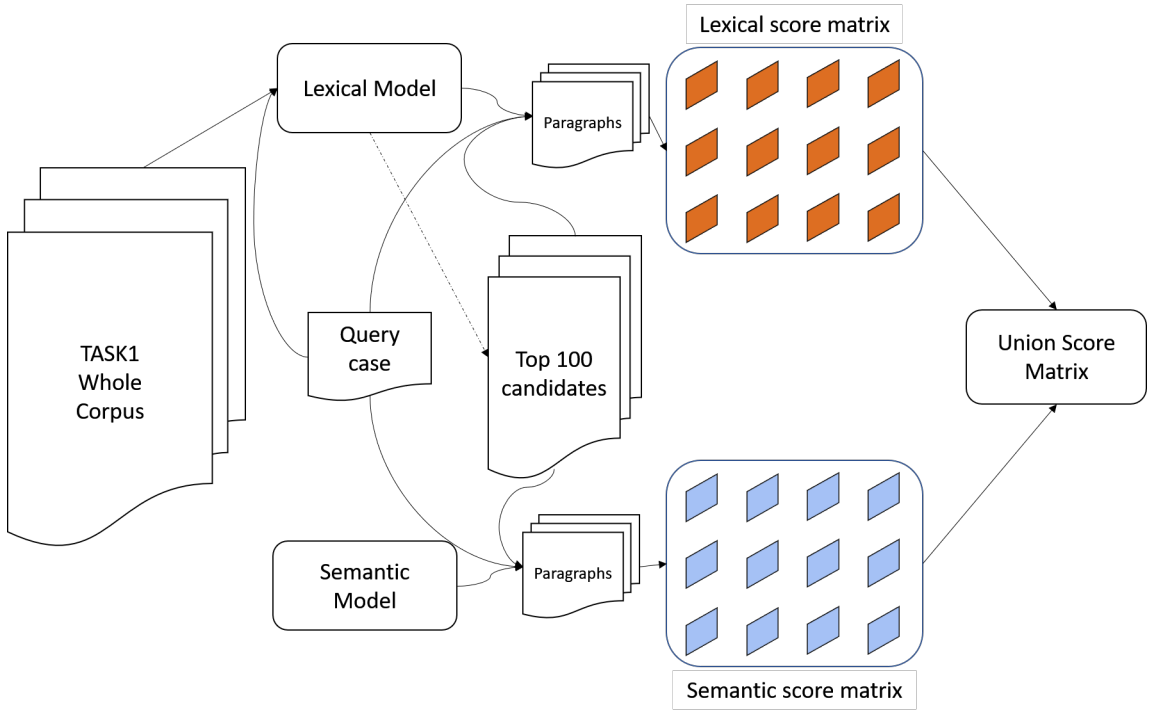

Figure 2: Demonstration of mixing lexical and semantic score.

in the candidate paragraph number [31], we can see the semantic relationship between two paragraphs through this example. For discovering the relevant paragraph, we combine the lexical similarity and the semantic similarity score.

Lexical Matching For the exact purpose of capturing lexical information, we use Rank-BM25 [1], a collection of algorithms for querying a set of documents and returning the ones most relevant to the query. For Task 1, after reducing the searching space to 100 queries, we separate the query case and also candidate cases into paragraphs. Assume that the query case has N paragraphs, and the candidate case has M paragraphs, we will calculate the lexical mapping score by Rank-BM25 for each query paragraph and every single paragraph in the candidate case. Then we obtain the matrix lexical score size NxM for each query-candidate pair. We keep these matrixes and their union with supporting score which is introduced later.

Supporting Matching As we mentioned in the Figure 1, we want to extract the semantic relationship between query paragraph and candidate paragraph. To obtain this relationship score, we use pretrained model BERT, and this model is provided by huggingface[2]. The same approach as lexical matching, we sequentially split a query and corresponding candidate cases into paragraphs, and obtain a matrix semantic score that has the same size as matrix lexical score. For more details about this supporting model, we do not use the original model provided by huggingface. Although BERT is currently one of the best, it is trained in the general domain. This actively demonstrates that BERT can not work well in a specialized domain such as law. For tackling this issue, we develop a silver dataset based on the COLIEE dataset for finetuning BERT and use this for predicting the semantic relationship between paragraphs.

Union Score To obtain the most relevant cases for a given query, we find a semantic as well as lexical relationship overlapped between the two paragraphs. For the exact purpose of developing union score we use the following formula:

$$union\_score = \alpha * score_{supporting} + (1 - \alpha) * score_{BM25} \quad (1)$$

Figure 2 shows the architecture of the system.

For Task 2, we use the same approach as Task 1. For the final runs, we use the supporting model and lexical model for 2 runs, and in the last run, we use NFSP model [6]. Using our proposed approach, Task2 is basically a binary classification with the training data as a set of sentence pairs. As a result, we can obtain more gold training data to train the supporting model. For optimizing the performance of the supporting model, after finetuning on silver, we finetune one more time on gold data of Task 2.

3.2 Task 3. The Statute Law Retrieval Task

This task involves reading a legal bar exam question Q, and extracting a subset of Japanese Civil Code Articles $A_{i|1 \leq i \leq n}$ which contains appropriate articles for answering the question, i.e. Entails($A_{i|1 \leq i \leq n}$, Q) or Entails($A_{i|1 \leq i \leq n}$, *not* Q).

As specified in [7], two main challenges of Task 3 are: (i) answering the questions which describe specific legal cases (note that the language used in statute law tends to be general), and (ii) addressing the long articles. The first challenge requires the model

[1] https://pypi.org/project/rank-bm25/
[2] https://huggingface.co/models

to have deduction ability, which is a difficult task and requires further research. Regarding the second challenge, we addressed it by performing text chunking technique on the prepared training data and self-labeled technique while finetuning pretrained models.

Training data preparation. We follow the training data preparation method proposed in [7]. Naturally, the pair of a question and each of its annotated entailing articles is considered a positive training example, and the pair of a question and any other article is considered a negative training example. However, this approach results in very big negative:positive ratio in training data. To reduce this ratio, in this phase, the maximum number of negative training examples is limited to be 150, choosing based on the rank of tf-idf scores. Please refer to [7] for more details.

Text chunking technique. We use BERT and RoBERTa, which are two prevalent language models, as pretrained models. However, they cannot handle a very long article without truncating it, because of the 512-token limitation. Besides, in Task 3, we observe that, in most cases, only a few parts of the entailing article entails the corresponding question while other parts do not. We provide an example in Table 1, where the question is entailed by only one part of the article. To address the aforementioned limitation of BERT and RoBERTa, we propose to split each article into multiple chunks using a sliding window (as sliding windows mitigate the cases where the entailing part of the article is split). Regarding training data generation, we followed the method proposed in Nguyen et al. [7], except that the pair used for training is (*question*, *chunk*) instead of (*question*, *article*). The label of pair (*question*, *chunk*) is derived from the label of its corresponding pair (*question*, *article*).

Q. R01-4-E	In cases any party who will suffer any detriment as a result of the fulfillment of a condition intentionally prevents the fulfillment of such condition, the counterparty may deem that such condition has been fulfilled.
Article 130	Part I General Provisions Chapter V Juridical Acts Section 5 Conditions and Time Limits (Prevention of Fulfillment of Conditions) **(1) If a party that would suffer a detriment as a result of the fulfillment of a condition intentionally prevents the fulfillment of that condition, the counterparty may deem that the condition has been fulfilled.** (2) If a party who would enjoy a benefit as a result of the fulfillment of a condition wrongfully has that condition fulfilled, the counterparty may deem that the condition has not been fulfilled.

Table 1: An example that only one part of the entailing article entails the corresponding question.

Self-labeled technique. Training data generated using text chunking contains noises. For example, assume that a long article A, which entails a question Q, is split into sub-articles $A_1, A_2, \ldots, A_n$, and only A_n entails Q, then the aforementioned training data generating method will label pairs $(Q, A_1), (Q, A_2), \ldots, (Q, A_n)$ as positive training examples; however, only pair (Q, A_n) should be labeled positive, while other pairs, i.e. $(Q, A_1), (Q, A_2), \ldots, (Q, A_{n-1})$, should be labeled as negative examples. Inspired by the self-labeled techniques [14], we propose to deploy a simple self-labeled technique to help mitigate noisy training examples. Specifically, first, a pretrained model finetunes on the generated training data. Next, the finetuned model predicts labels for training examples. After that, we modify the labels based on rules, and the label-modified training data is used for the next finetuning phase. This self-labeled and finetuning process can be iterated multiple times. Regarding the label modifying rules, we only keep label modification if the label is converted from positive to negative. After label modification, the number of noisy examples tends to decrease, which allows the previous-phase finetuned model to learn from a more accurate data.

Model ensembling. The outcome of each model is the prediction based on the particular characteristics the models learned during training. Since each model has its advantages and disadvantages, model ensembling, an effective machine learning approach for incorporating models, appears to help produce better predictions. In our work, we ensemble models by deploying weighted aggregation on models' predictions. Prior to the ensemble process, the outputs of multiple models were scaled by min-max normalization method, so that they were standardized in the range $[0, 1]$. It ensures that the standardized outputs of each model have a similar impact on the final prediction results. We divided the dataset into training, development, and test set. The ensemble method weights were constructed from the development set and applied to the test set.

3.3 Task 4. The Legal Textual Entailment Task

This task involves the identification of an entailment relationship between relevant articles $A_{i|1 \le i \le n}$ (which is derived from Task 3's results) and a question Q. The models are required to determine whether the relevant articles entail "Q" or "$notQ$". Given a pair of legal bar exam question and article (Q, A_i), the models return a binary value for determining whether (A_i) entails (Q). To address this task, we modified the training data preparation step, then use thed same model architecture in Task 3 (Section 3.2) for training.

Data preparation. Based on our observation, the challenge of this task is to extract the relevance between a question and articles for classification while the number of given articles is relatively small (usually 1 or 2 articles are given). We hypothesize that the model can extract information more effectively and consistently if there are more relevant articles given. Therefore, we adapt the data augmentation technique mentioned in Task 3 (which is based on tf-idf scores) to increase the number of relevant articles for each question. Besides, we also use the *text chunking* and *self-labeled* techniques introduced in Section 3.2 for dealing with long article challenge.

3.4 Task 5. Statute Law Question Answering

The goal in Task 5 of the models is to answer legal questions. In detail, with a given statement, the model needs to answer whether

Sentence Pair	NFSP Label	NMSP Label
Shall we go out? The weather is nice.	-	2
お出掛けしよ？いい天気ね。	-	2
お出掛けしよ？ The weather is nice.	-	2
Shall we go out? いい天気ね。	-	2
いい天気ね。お出掛けしよ？	-	1
The weather is nice. Shall we go out?	-	1
The weather is nice. お出掛けしよ？	1	1
いい天気ね。Shall we go out?	1	1
The weather is nice. ランダム文。	0	0
いい天気ね。Random Sentence.	0	0
The weather is nice. Random Sentence.	-	0
いい天気ね。ランダム文。	-	0

Table 2: Examples of pretraining data.

Model	Max Len.	Batch Size	#Batches	Acc.
NFSP Base	512	16	24,000	94.4%
NFSP Distilled	512	32	34,000	92.2%
NMSP Base	512	16	320,000	88.0%
NMSP Distilled	512	32	496,000	87.7%

Table 3: Parameters and performances in pretraining the models on valid set.

the statement is true or false in the legal aspect. In essence, Task 5 is constructed from Task 4, ignoring the step of retrieving the related clauses from the Japanese Civil Code.

Our novelty in Task 5's solution is the introduction of two NMSP and NFSP models. The main idea in building these two models is to use translation information as means of ambiguity reduction. We argue that a sentence in natural language can have many meanings, but in its translation, the most correct meaning will be expressed. In addition, the meaning is also determined by the context, that is, the sentences before and after the current sentence.

These models, named ParaLaw Nets [6], are pretrained on cross-lingual sentence-level tasks before being finetuned for use in the COLIEE problem. The data we use to pretrain these models is bilingual Japanese law data provided by Japanese Law Translation website [3]. We formulate the pretraining task for NFSP as a binary classification problem and NMSP as a multi-label classification problem.

We design the pretraining task to force the model to learn the semantic relationship in 2 continuous sentences crossing two languages. From original sentences as "The weather is nice. Shall we go out?", their translations "いい天気ね。お出掛けしよ？", and random sentences as "Random sentence.", "ランダム文。", we can generate the training samples as in Table 2.

These models are pretrained until the performance on the validation set does not increase. Through the results in Table 3, we see that the models have better performance on the validation set with NFSP task, base models outperform distilled models. With these numbers, we can accept the assumption that the NFSP task is more straightforward than the NMSP task.

For the finetuned task, we use a similar approach with our previous systems [5, 7]. We use the Japanese Civil Code and the data given by COLIEE's organizer as training and validation data. Working on multilingual data, we create negation rules for Japanese and removed law sentences which are represented as a list. After augmentation, we obtain 7000 sentences in two languages.

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

Data Source	#case	#paragraph	#sentence	#example
Task 1	4415	172495	626540	378720
Task 2	425	-	913	18238

Table 4: Supporting dataset.

4 EXPERIMENTS

4.1 Task 1 and Task 2. Case Law Processing

Training data As we mentioned in Section 3, we create a supporting training dataset to train BERT model and the analysis of this training dataset is as Table 4. From 4415 cases in Task 1 raw dataset, we can extract more than 170K paragraphs. However, inside these paragraphs, some of them contain a lot of french text. This actively demonstrates that we need a filter step to extract clean English text. For the purpose of avoiding noise in the dataset, we use langdetect[4] to filter and remove french text from training data. The clean paragraphs are obtained so far, we split every single paragraph into sentences (>625K in total), and from these sentences we apply some technique to generate supporting examples. The massive number of silver training examples for training BERT is over 350K examples.

Besides silver data, we utilize the data from Task 2 to extract more training dataset. As you can see in the Table 4, the number of gold examples we can extract is more than 18K.

For Task 1, we submit 3 runs as follow:

- Run1: Lexical score combine with semantic score with ratio α 7:3.
- Run2: Lexical score combine with semantic score with ratio α 3:7.
- Run3: Only supporting score.

For Task 2, we submit 3 runs as follow:

- Run1: Lexical score combine with semantic score with ratio α 7:3.
- Run2: Lexical score combine with semantic score with ratio α 7:3 and finetuning on gold training dataset.
- Run3: Lexical score combine with NFSP model's score.

Our method on Task 1 has bad performance on the test set, the particular reason for this issue is the problem when we limit the searching space from 4000 to 100, maybe lexical matching works badly in this circumstance. For Task 2, as the numbers in Table 5, our method achieves 61% on F score, and we place 5TH in COLIEE 2021.

4.2 Task 3. The Statute Law Retrieval Task

We trained and evaluated our proposed models for this task with the previous year's dataset. Macro- precisions, recalls, and F_2 scores are reported. Note that the reported F_2 scores are the F_2 scores of class-wise precision means and class-wise recall means.

[4]https://pypi.org/project/langdetect/

Run ID	Accuracy
BM25Supporting_Denoising	0.6116
BM25Supporting_Denoising_Finetune	0.6091
NFSP_BM25	0.5868

Table 5: Result on Task 2.

Chunking info.	Return	Retrieved	P	R	F2
no chunking	118	81	68.24	72.52	71.62
110/20	190	108	61.20	67.87	66.42
150/10	122	85	64.77	66.67	66.28
150/20	129	93	68.09	70.72	70.18
150/40	131	92	67.12	71.17	70.32
150/50	132	94	69.74	73.42	72.66
200/50	139	90	67.12	72.97	71.72
300/50	110	74	65.39	67.57	67.12

Table 6: (Task 3) Results of *bert-base-japanese* pretrained model finetuning with the text chunking technique. The values in the *Chunking info.* column denotes chunking settings with format <*window_size*>/<*stride*>.

We conducted multiple experiments with different settings to find the most appropriate settings when applying text chunking technique. In these experiments, we finetuned 3 epochs on *bert-base-japanese*[5] pretrained model and report results in Table 6. We use <*window_size*>/<*stride*> to denote the sliding window parameters. The results indicate that <150>/<50> seems to be the most appropriate setting for the task. We use this setting to conduct experiments relating to self-training technique. Specifically, first, we finetuned *bert-base-japanese* pretrained model with e_1 epochs, then performed self-labeling process and continued to finetune with e_2 epochs. We use <e_1>/<e_2> to denote this settings. The results in Table 7 suggests that $e_1 = 2$ seems to be the "just right" parameter for the first finetuning process (when $e_1 = 3$, the finetuned model seems to start overfitting); and as e_2 increases, F_2 tends to increase. The results demonstrate the positive impact of our proposed methods: $F2$ increased from 71.62 to 72.66 when applying the text chunking technique, and this number is 72.91 with the self-labeled technique.

We also did experiments with *bert-base-japanese-whole-word-masking*[6] and *xlm-roberta-base*[7] pretrained models, but we do not report in this paper because of paper length limitation. We performed model ensembling on the outputs of those models.

For the contest submissions, we submitted 3 runs based on the 3 proposed approaches. The model ensembling method returned the highest result among the three. We are the runner-up of this task. The results of final runs of all participants are in Table 8.

4.3 Task 4. The Legal Textual Entailment Task

Similar to Task 3, we also trained and evaluated our proposed models with the dataset in the previous year with the questions having id *R-01-** as the development set. Because of the relatively small training data, we run 5 times with each setting and report the mean and standard deviation values. For this task, most of the hyperparameters follows settings in Nguyen et al. [7] where $batch\_size = 16$ and $learning\_rate = 1e^{-5}$.

Pretrained model and Data augmentation. Firstly, we conducted the experiments to find the most suitable pretrained model for this task, and the most suitable setting for the tf-idf-based data augmentation method (Table 9). Based on the experimental results, we found that the *bert-base-japanese-whole-word-masking* pretrained model is more suitable for this task than others. Besides, the tf-idf-based augmentation data method also help increase the model performance.

Performance stability. In addition, we found that the performance of the model with a small epoch is fairly unstable. Therefore, we experimented with a bigger number of training epochs (Table 10). The experimental results demonstrate that the runs with higher epochs tend to achieve more stable accuracies, but the model

Setting	Return	Retrieved	P	R	F2
3/0	132	94	69.74	73.42	72.66
1/1	109	81	68.02	67.57	67.66
1/2	145	98	66.89	72.52	71.32
1/3	133	96	68.77	72.97	72.09
2/1	161	95	64.55	72.52	70.77
2/2	161	95	64.55	72.52	70.77
2/3	195	104	62.39	76.13	72.91
3/1	146	97	63.32	68.92	67.72
3/2	146	96	64.37	71.17	69.70
3/3	147	97	60.02	65.77	64.53

Table 7: (Task 3) Results of *bert-base-japanese* pretrained model with the self-labeled technique. The values in the *Setting* column follows the format <e_1>/<e_2>.

Run ID	sid	F2	Prec	Recall
OvGU_run1	E/J	0.7302	0.6749	0.7778
JNLP.CrossLMultiLThreshold	E/J	0.7227	0.6000	0.8025
BM25.UA	E/J	0.7092	0.7531	0.7037
JNLP.CrossLBertJP	E/J	0.7090	0.6241	0.7716
R3.LLNTU	E/J	0.7047	0.6656	0.7438
R2.LLNTU	E/J	0.7039	0.6770	0.7315
R1.LLNTU	E/J	0.6875	0.6368	0.7315
JNLP.CrossLBertJPC15030C15050	E/J	0.6838	0.5535	0.7778
OvGU_run2	E/J	0.6717	0.4857	0.8025
TFIDF.UA	E/J	0.6571	0.6790	0.6543
LM.UA	E/J	0.5460	0.5679	0.5432
TR_HB	E/J	0.5226	0.3333	0.6173
HUKB-3	J	0.5224	0.2901	0.6975
HUKB-1	J	0.4732	0.2397	0.6543
TR_AV1	E/J	0.3599	0.2622	0.5123
TR_AV2	E/J	0.3369	0.1490	0.5556
HUKB-2	J	0.3258	0.3272	0.3272
OvGU_run3	E/J	0.3016	0.1570	0.7006

Table 8: (Task 3) Result of final runs on the test set, the underlined lines refer to our models.

[5]https://huggingface.co/cl-tohoku/bert-base-japanese
[6]https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking
[7]https://huggingface.co/xlm-roberta-base

Model	Origin	tf-idf1	tf-idf2	tf-idf5	tf-idf20
BertJp	55.9 ± 3.7	-	-	-	-
BertJp2	60.4 ± 4.1	61.6 ± 5.0	62.7 ± 5.7	61.3 ± 4.4	58.4 ± 2.9

Table 9: (Task 4) Model accuracies with different tf-idf augmentation setting. The name *BertJp, BertJp2* indicates that we used a pre-trained *bert-base-japanese* and *bert-base-japanese-whole-word-masking,* respectively. The name column follows the format *tf-idf<number>* where *<number>* denotes the number of augmented articles appended for each question.

# epochs	tf-idf1	tf-idf2	tf-idf5
3	61.6 ± 5.0	62.7 ± 5.7	61.3 ± 4.4
10	61.8 ± 3.0	62.9 ± 2.2	64.7 ± 2.5
20	-	61.6 ± 1.8	64.0 ± 1.3

Table 10: (Task 4) Accuracies of *BertJp2* with different training epochs.

Setting	tf-idf2	tf-idf5
1/10	62.9 ± 1.7	63.1 ± 1.9
2/10	64.3 ± 0.5	64.3 ± 0.5
3/10	62.2 ± 1.7	64.1 ± 1.0

Table 11: (Task 4) Accuracies of *BertJp2* using the *self-labeled* technique with different settings on chunking data where *window_size = 150, stride = 50.* The values in the *Setting* column denotes <e_1>/<e_2>.

can be overfitted if we increase the number of epochs too much. Besides, the augmentation data also helps the model performance to be more stable.

Long article challenge. Finally, to address the long article challenge, we conducted the experiments using the *text chunking* and the *self-labeled* techniques described in Task 3 (Table 11). Although the accuracy of models in this setting did not increase, the variant ranges are smaller. It may be because the *text chunking* and the *self-labeled* techniques help eliminate noises in training data.

The results on blind test set are shown in the Table 12 with the id "*JNLP.Enss5C15050*" refers to the model BertJp2 using augmentation data tf-idf5; "*JNLP.Enss5C15050SilverE2E10*" refers to the model BertJp2 using augmentation data tf-idf5, and <e_1>/<e_2> is 2 /10; "*JNLP.EnssBest*" refers to the enssemble of both models.

4.4 Task 5. Statute Law Question Answering

We compare our proposed models together and with other cross-lingual and multilingual baselines such as XLM-RoBERTa [2] and original BERT Multilingual [3]. In the 7000 augmented sentences, we divide the train set and validation set with the ratio of 9:1.

Our experiments show that NFSP Base and NMSP Base achieve the best performance and have stable loss decrease. NFSP Distilled, NMSP Distilled and XLM-RoBERTa fail to learn from the data and their performance equivalent to that of random sampling. BERT Multilingual is in the middle of the ranked list of models in Table 13.

Team	sid	Correct	Acc.
HUKB	HUKB-2	57	0.7037
UA	UA_parser	54	0.6667
JNLP	JNLP.Enss5C15050	51	0.6296
JNLP	JNLP.Enss5C15050SilverE2E10	51	0.6296
JNLP	JNLP.EnssBest	51	0.6296
OVGU	OVGU_run3	48	0.5926
TR	TR-Ensemble	48	0.5926
KIS	KIS1	44	0.5432
UA	UA_1st	44	0.5432

Table 12: (Task 4) Results final runs on the test set. The underlined lines refer to our submissions.

Model	Accuracy
NFSP Base	71.0%
NFSP Distilled	51.1%
NMSP Base	79.5%
NMSP Distilled	48.9%
XLM-RoBERTa	51.1%
BERT Multilingual	64.1%

Table 13: (Task 5) Performance of models on validation set.

Team	Run ID	Correct	Accuracy
	BaseLine	No 43/All 81	0.5309
JNLP	JNLP.NFSP	49	**0.6049**
UA	UA_parser	46	0.5679
JNLP	JNLP.NMSP	45	0.5556
UA	UA_dl	45	0.5556
TR	TRDistillRoberta	44	0.5432
KIS	KIS_2	41	0.5062
KIS	KIS_3	41	0.5062
UA	UA_elmo	40	0.4938
JNLP	JNLP.BERT_Multilingual	38	0.4691
KIS	KIS_1	35	0.4321
TR	TRGPT3Ada	35	0.4321
TR	TRGPT3Davinci	35	0.4321

Table 14: (Task 5) Result of final runs on the test set, the underlined lines refer to our models.

Therefore, we choose NFSP Base, NMSP Base and original BERT Multilingual as candidates for the final run.

Table 14 is the result of the models on the blind test of COLIEE-2021's organizer. On this test set, NFSP Base outperforms other methods and becomes the best system for this task. NMSP is in third place and the original BERT Multilingual has the performance below the baseline. These results again support our proposal in pretraining models using sentence-level cross-lingual information.

5 CONCLUSIONS

This paper presents JNLP Team's approaches using deep learning to the legal text processing tasks in the COLIEE-2021 competition.

Due to the limited amount of data and the difficulty of the tasks, we used pretraining methods to solve the problems. The experimental results and the performance of the models on the blind test show the reasonability and robustness of the proposed methods.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP17H06103 and JP20K20406.

REFERENCES

[1] B.Gain, D.Bandyopadhyay, T.Saikh, and A.Ekbal. 2019. Iitp@coliee 2019: Legal information retrieval using bm25 and bert.

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] J.Hudzina, T.Vacek, K.Madan, C.Tonya, and F.Schilder. 2019. Statutory entailment using similarity features and decomposable attention models.

[5] HT Nguyen, V Tran, and LM Nguyen. 2019. A deep learning approach for statute law entailment task in COLIEE-2019. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE* (2019).

[6] Ha-Thanh Nguyen, Vu Tran, Phuong Minh Nguyen, Quan Minh Bui, Chau Minh Nguyen, Binh Tran Dang, Hai-Yen Thi Vuong, Ken Satoh, and Minh Le Nguyen. 2021. ParaLaw Nets - Cross-lingual Sentence-level Pretraining for Legal Text Processing.

[7] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *arXiv preprint arXiv:2011.08071* (2020).

[8] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2019. Combining Similarity and Transformer Methods for Case Law Entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19)*. 290–296. https://doi.org/10.1145/3322640.3326741

[9] R.Hayashi and Y.Kano. 2019. Searching relevant articles for legal bar exam by doc2vec and tf-idf.

[10] R.Hoshino, N.Kiyota, and Y.Kano. 2019. Question answering system for legal bar examination using predicate argument structures focusing on exceptions.

[11] S.Wehnert, S.A.Hoque, W.Fenske, and G.Saake. 2019. Threshold-based retrieval and textual entailment detection on legal bar exam questions.

[12] T.B.Dang, T.Nguyen, and L.M.Nguyen. 2019. An approach to statute law retrieval task in coliee-2019.

[13] Vu Tran, Minh Le Nguyen, and Ken Satoh. 2019. Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*. 275–282.

[14] Isaac Triguero, Salvador García, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems* 42, 2 (2015), 245–284.

ParaLaw Nets - Cross-lingual Sentence-level Pretraining for Legal Text Processing

Ha-Thanh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Vu Tran
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Phuong Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Thi-Hai-Yen Vuong
University of Engineering and Technology, VNU
Hanoi, Vietnam

Quan Minh Bui
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Chau Minh Nguyen
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

Binh Tran Dang
Japan Advanced Institute of Science and Technology
Ishikawa, Japan

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

Ken Satoh
National Institute of Informatics
Tokyo, Japan

ABSTRACT

Ambiguity is a characteristic of natural language, which makes expression ideas flexible. However, in a domain that requires accurate statements, it becomes a barrier. Specifically, a single word can have many meanings and multiple words can have the same meaning. When translating a text into a foreign language, the translator needs to determine the exact meaning of each element in the original sentence to produce the correct translation sentence. From that observation, in this paper, we propose ParaLaw Nets, a pretrained model family using sentence-level cross-lingual information to reduce ambiguity and increase the performance in legal text processing. This approach achieved the best result in the Question Answering task of COLIEE-2021.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Applied computing** → *Law*.

KEYWORDS

pretrained model, legal text processing, cross-lingual, sentence-level

1 INTRODUCTION

Transformer [16], the architecture using encoders and decoders with the attention mechanism has become the best practice in many problems. Variations of this model continuously produce new state-of-the-art results in different tasks. The main difference between these variants lies in how the pretraining tasks are designed to take advantage of the latent information in the data. Pretrained models such as BERT [4], GPTs [1, 13, 14], ALBERT [8], ELECTRA [2], and BART [9] are all based on Transfomer but have different approaches to the pretraining tasks. Hence, proposals of pretraining tasks are essential contributions to the development of pretrained models.

Transformer-based pretrained models all need effective pretraining tasks to learn latent patterns in the data. The authors of BERT use two tasks, *masked language modeling* and *next sentence prediction* to train this model. The idea of the *masked language modeling* task is that when some words are masked, a good language model should be able to recover the original words. The *next sentence prediction* is the task that requires BERT to determine whether a sentence is the next one of another sentence.

GPT is a language model that is trained to recognize the next word of a set of given words, the same approach as N-gram language model. Based on GPT, later versions of this model with a huge number of parameters are able to perform different tasks with very few training samples (few-shot learning) or even no training samples at all (zero-shot learning). GPTs authors also introduce the concept of task conditioning which means with the same input, in different tasks the model must output differently. Language models with patterns learned from data can perform many impressive tasks.

In addition to the pretraining tasks of BERT and GPT, there are other proposals that help to improve the effectiveness or efficiency of the model. ALBERT replaces BERT's *next sentence prediction* task with *sentence order prediction*. Instead of simply concluding whether two sentences are consecutive or not, the model needs to predict the order of two consecutive sentences. ELECTRA's authors proposed *replaced token detection* as an alternative to *the masked language modeling* task. With a discriminator and a generator parallelly trained, the language model needs to find out which token is authentic, which is replaced. BART is considered as a combination of BERT and GPT. This model has both prediction and generation capabilities. A series of pretraining tasks which is applied to BART are *Token Masking*, *Token Deletion*, *Text Infilling*, *Sentence Permutation*, and *Document Rotation*.

Pretraining tasks are usually formed based on existing data structures. The language modeling tasks are based on the consecutive and co-occurrence structure of words, sentences, and paragraphs. Additionally, there exist many natural data structures that help pretrained models increase their performance. Detecting specific structures contained in data is important to formulate the corresponding tasks.

A translation of a text gives us more information about its meaning than just a set of vocabulary translated into a new language. A sentence in a language may contain many different semantics and depending on the context, the translation needs to be the most appropriate sentence in the target language with the same meaning. For example, as in Figure 1 in Japanese, こんにちは can be a midday greeting or a formal way to say "hello". In consequence, in the morning context, this sentence needs to be translated as "hello" rather than "good afternoon". Likewise, "I" in English can be translated in a multitude of ways in Japanese. Determining which is the correct translation must depend on the context of the sentence.

It is important to determine the correct context to correctly understand the meaning of a sentence when dealing with difficult documents such as the law. A correct understanding of semantic will not depend on its language of expression. Therefore, using the original version and the translation in parallel can help the model learn the semantic with better precision.

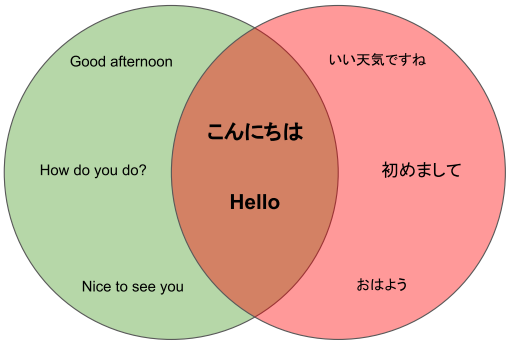

Figure 1: A single word may have multiple translations.

From such observations, we propose ParaLaw Nets, cross-lingual sentence-level pretraining models for legal document processing. The idea is to force the model to learn the context dependence from bilingual sentences. We conduct experiments on COLIEE-2021 data to verify the effectiveness of the method. Our approach is superior to other multilingual approaches such as BERT Multilingual or XLM-RoBERTa.

2 RELATED WORK

2.1 Multilingual and Crosslingual Approaches

The NLP resources are not uniform across languages. The language with the most abundant resources is English. That of other languages is usually much less. Therefore, resources developed in English will later be transferred to other languages. The multilingual nature of resources is often understood as a translation into other languages from English. The aspect of using semantic in multilingual to reduce ambiguity is also worth investigating.

The multilingual implementation was introduced for the first time when the authors of BERT [4] presented this kind of pretrained models. However, the article does not mention in detail how to build this variant of BERT. Fortunately, on their Github [1], the authors state that this model is trained with the 100 largest Wikipedia languages. Common languages are downsampled and less common languages are upsampled to ensure the patterns are learnable across different languages.

Lample et al. [7] proposed the idea of pretraining models using multiple languages. The authors use 3 tasks to train the model: *Causal Language Modeling* (CLM), *Masked Language Modeling* (MLM) and *Translation Language Modeling* (TLM). Among them, TLM is a task that requires many languages, the model uses translation knowledge between languages to fill the missing words in the blanks. According to the authors, this task forces the model to learn the alignment between languages and leverage the context of one language when the context of the other language is not complete.

XLM-RoBERTa [3] is a pretrained model that uses multilingual advantages over 100 languages. This model with a huge amount of training data on these languages achieved state-of-the-art results on different tasks on the GLUE benchmark compared with cross-language baselines. Through the article, the authors also prove the superiority of multilingual models compared with single-language models.

2.2 Pretrained Models in Legal Domain

In the NLP domains, legal document processing needs particular approaches. The legal vocabulary is different from ordinary language, and law sentences often have a complex structure. Pretrained methods for legal domains have been proved to be competitive with other methods. Most COLIEE-2020 approaches, including the best systems, use this approach [12].

The Task 1 winner, *cyber* team uses a pretrained Transformer model in their implementation for vector representation [5]. To solve Task 2, *JNLP* team [11] uses the pretrained model based on supporting information to find supporting paragraphs across the legal cases. The Task 3 winner, *LLNTU* team [6] uses BERT to classify whether an article is relevant to a given legal question or not. For Task 4, *JNLP* team also pretrains a legal language model from the case law data to generate strong contextual embedding for the model before making predictions in the statute law. With the limited data and narrow specializations, the pretrained models seem to be a competitive approach.

From observing that multilingual information can support to model the meaning of sentences, we propose Paralaw Nets, pretrained models using multilingual pair of legal sentences. In any linguistic tasks, especially in the legal domain, it is very important to understand correctly the meaning of a sentence in making predictions. By forcing the model to learn the possibility of the semantic connection between the two sentences, we believe in having a strong pretrained model.

3 PARALAW NETS

3.1 Pretraining

The general idea of this paper's approach is to utilize the hidden information that is aligned between two sentences in two different languages to train the model. Different from the token-based approach of XLM-RoBERTa, we use sentence-level approach. The semantic understanding ability of a model is judged on how well it predicts the logical order of sentences in different languages. We

[1] https://github.com/google-research

propose two approaches called *Next Foreign Sentence Prediction* (NFSP) and *Neighbor Multilingual Sentence Prediction* (NMSP).

Not only considering multilingual as an additional version of the pretrained models in English, we believe that the translation information will help the model to have a better understanding of the sentence meaning. When an idea is correctly understood by a model, this model can verify the expression of that idea in all languages. In other words, semantic is expressed through, but not limited by, the language in which it is expressed.

In the NFSP task, the model needs to read two sentences in different languages and determine if their semantic belong to two consecutive sentences in a document. To this end, the model needs to correctly understand the meaning of each sentence. This is intended to reduce ambiguity in the expression of sentences in both languages.

For example, from original sentences as "The weather is nice. Shall we go out?" and their translations "いい天気ね。お出掛けしよ？", we can create 2 positive samples in the training data as:

- The weather is nice. お出掛けしよ？
- いい天気ね。Shall we go out?

The negative samples are pairs of a sentence in the original documents and the translation of another random sentence.

NMSP shares a similar approach with NFSP but the training data is generated with more cases. In addition to bilingual pairs, we include pairs of same-language sentences in two languages. If cross-lingual factor is not considered, NFSP has the same approach as BERT's NSP task. BERT's NSP critics argue that the Transformer model can rely on co-appearance information to predict the labels of samples in this task. That is, even if the model does not know whether two sentences are consecutive or not, it can still guess the label based on the proximity of the topic. Dealing with that potential issue, our hypothesis is that if the model determines that sentence A follows sentence B, it must also know that sentence B comes before sentence A. In addition, we assign one label to help the model learn that two sentences are not contiguous in a text.

NFSP is a binary classification problem, NMSP is a multi-label classification problem with labels corresponding to the case of random sampling, normal order, reverse order, and non-contiguous.

The data used to pretrain our Paralaw Nets is bilingual legal sentences. Thanks to globalization, legal documents in other languages are often translated into English sentence by sentence. This creates a great edge for ParaLaw Nets in terms of pretraining data. The experimental models introduced in this paper are trained with Japanese-English bilingual legal data. However, this approach can be generalized to all language pairs or groups.

We pretrain models according to the tasks mentioned. The BERT multilingual base model is used as the base model for both NFSP and NMSP. The distilled versions use the configuration and architecture of BERTDistilled [15]. All models are cased configurations. Data for pretraining NFSP contains 239,000 samples, data for pretraining NMSP contains 718,000 samples. The training process is stopped when the performance of the model does not increase on the validation set. Table 1 shows the parameters and the performances in pretraining the models.

3.2 Finetuning

Next, we finetune the models for the lawfulness classification problem in Task 5 of the COLIEE-2021. Given a statement as a legal question, the model needs to decide whether that statement is true or false. Without the support of lexical-based retrieval systems, the model needs to really understand the meaning of the previously learned propositions, generalize them and apply that knowledge to the question. Table 2 shows examples of this task.

To strengthen the bilingual model, we use original and augmented data in both English and Japanese. Negation is the main method used to create variations of original data. The first negation rule that is matched will be used only once to avoid the negation of the negation. With English negation rules, we reuse the rules proposed by Nguyen et al [10]. Japanese negation rules are derived from basic Japanese syntax. English and Japanese negation rules are shown in Tables 3 and 4.

4 EXPERIMENTS

4.1 Experimental Setup

We do experiments to choose the best models to generate predictions on the blind test set of COLIEE-2021's organizer. Data in English includes all data provided by the organizer and a portion of the Japanese Civil Code. In the Japanese Civil Code, statements that are represented as lists are removed because their elements are often lengthy and do not express a complete semantic. In addition, it is not a valid approach if we concatenate them without carefully considering the logical semantic of the whole statement. For example, in natural language, *and/or* conjunction in a sentence may differ from the logical meaning which the sentence expresses. The process of filtering sentences is processed completely automatically based on the XML structure provided by the Japanese Law Translation website [2].

We augment the data by negation rules as described in Section 3. All full sentences in the Japanese Civil Code are considered lawful and their negations are unlawful. With the data provided by the organizers, the sentences already have labels, we create more data by creating negation of the content and reversing the labels. Data after augmentation contains 7,000 sentences, we use 10% for validation data, the rest is for training.

We experiment on the lawfulness classification problem with 6 different models including the original BERT multilingual base model from Google, XLM-RoBERTa, NFSP base, NFSP distilled, NMSP base and NMSP distilled.

4.2 Experimental Results

Pretraining the models, we observed interesting phenomena when training the models using Japanese data. If we use all the data is augmented with the rules in Table 4, all models cannot converge. To solve this problem, we use a simple curriculum learning strategy for Japanese data. We train 3 epochs using augmented data by the first three negation rules before training with the whole dataset. With only the English data, we did not encounter this problem. We believe that this is an indication of the more challenges in understanding Japanese versus understanding English for the cross-lingual models

[2] https://www.japaneselawtranslation.go.jp

Table 1: Parameters and performances in pretraining the models

Model	Max Length	Batch Size	Number of Batches	Validation Accuracy
NFSP Base	512	16	24,000	94.4%
NFSP Distilled	512	32	34,000	92.2%
NMSP Base	512	16	320,000	88.0%
NMSP Distilled	512	32	496,000	87.7%

Table 2: Examples for COLIEE-2021's Task 5

Sentence	Output
No abuse of rights is permitted.	Yes
The age of majority is reached when a person has reached the age of 12.	No

Table 3: Rules applied for negation statement generation in English [10]

Original Statement	Negation Statement Generation
contains *not*	Remove *not* from original statement
contains *shall*	Replace *shall* with *shall not*
contains *should*	Replace *should* with *should not*
contains *may*	Replace *may* with *may not*
contains *must*	Replace *must* with *must not*
contains *is*	Replace *is* with *is not*
contains *are*	Replace *are* with *are not*
contains *will be*	Replace *will be* with *will not be*
contains *can*	Replace *can* with *cannot*
contains *cannot*	Replace *cannot* with *can*
contains *with*	Replace *with* with *without*
contains *without*	Replace *without* with *with*
contains *A*	Replace *A* with *No*
contains *An*	Replace *An* with *No*

trained with our approach. Looking at Table 3 and Table 4, it can be seen that the English negations are related to the word "not", the negations of Japanese are more diverse and complex. Therefore the model needs more skills to distinguish negation.

Table 5 shows the performance of the models on the validation set. The distilled models and XLM-RoBERTa completely fail to learn in this task. Figures 2-7 plot the loss fluctuation of these models. The loss values fluctuate around 0.7 and do not decrease. The original BERT Multilingual model passes the threshold of 0.7. Although this model has a huge variant loss value after that, its accuracy is better than XLM-RoBERTa and the distilled models. NFSP Base and NMSP Base have loss reduced to below 0.7 and loss variation is much more stable.

From the experimental results, we choose three candidates for final runs: NFSP Base, NMSP Base and Original BERT Multilingual.

4.3 Final Runs Result

We run predictions on the English blind test set provided by the COLIEE-2021's organizer. Among our 3 models, NFSP Base has the best result, next is NMSP Base, and BERT Multilingual has the lowest result. We were also surprised that NFSP Base outperformed

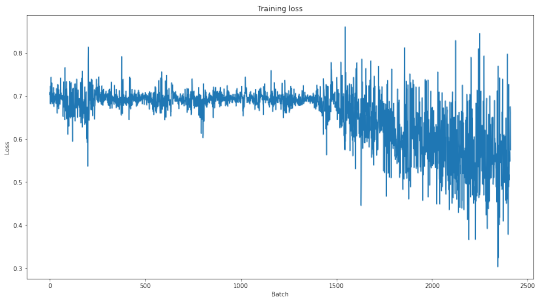

Figure 2: Loss fluctuation of BERT Multilingual.

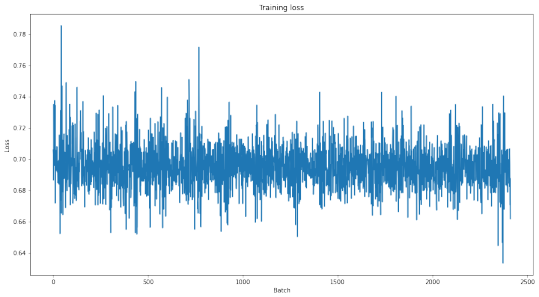

Figure 3: Loss fluctuation of XLM-RoBERTa.

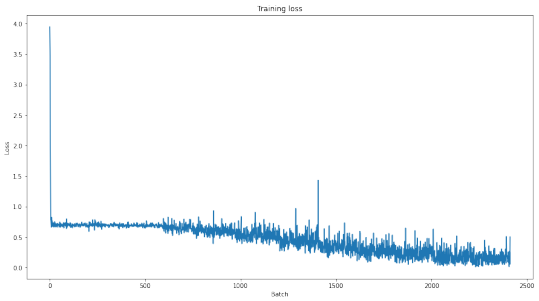

Figure 4: Loss fluctuation of NMSP Base.

Table 4: Rules applied for negation statement generation in Japanese

Original Statement	Negation Statement Generation
contains ません	Replace ません with ます
contains できる	Replace できる with できない
contains できない	Replace できない with できる
contains した	Replace した with しなかった
contains でない	Replace でない with である
contains できた	Replace できた with できなかった
contains させる	Replace させる with させない
contains ている	Replace ている with ていない
contains がない	Replace がない with がある
contains ではない	Replace ではない with ではある
contains ことがある	Replace ことがある with ことがない
contains しなければならない	Replace しなければならない with してはいけません
contains ならない	Replace ならない with なる

Table 5: Performance of models on validation set

Model	Accuracy
NFSP Base	71.0%
NFSP Distilled	51.1%
NMSP Base	79.5%
NMSP Distilled	48.9%
XLM-RoBERTa	51.1%
BERT Multilingual	64.1%

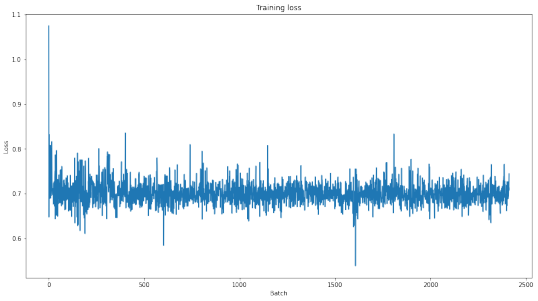

Figure 5: Loss fluctuation of NMSP Distilled.

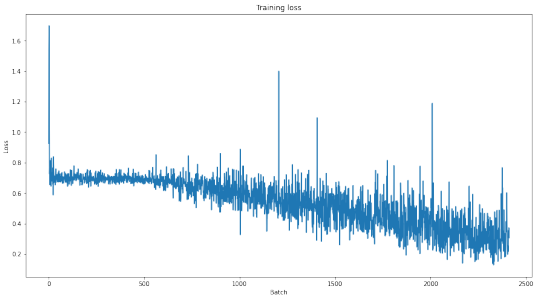

Figure 6: Loss fluctuation of NFSP Base.

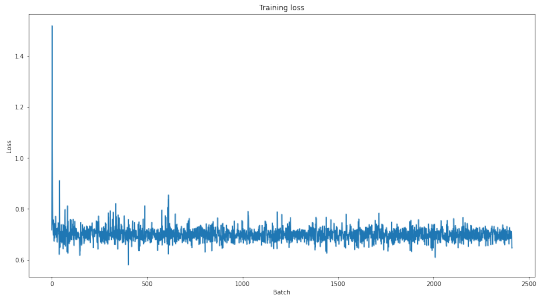

Figure 7: Loss fluctuation of NFSP Distilled.

NMSP Base and stayed first on the leaderboard. This may indicate that the test set distribution is somewhat biased against the latent features that the NFSP learned, which is not present in our validation set. However, the test set results support the notion that pretraining with cross-lingual information by our approach helps the model learn more accurately on finetuned tasks.

5 CONCLUSIONS

This paper proposes an approach using sentence-level cross-lingual information to pretrain transformer models before finetuning on the specific task. Taking advantage of cross-lingual resources in legal documents, we introduce NFSP and NMSP models which have impressive performance in our experiments as well as in COLIEE-2021's blind test. The idea of this study is applicable to problems with aligned translation data as legal text processing.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP17H06103 and JP20K20406.

REFERENCES

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

Table 6: Result of final runs on the test set, the underlined lines refer to our models

Team	Run ID	Correct	Accuracy
	BaseLine	No 43/All 81	0.5309
JNLP	JNLP.NFSP	49	**0.6049**
UA	UA_parser	46	0.5679
JNLP	JNLP.NMSP	45	0.5556
UA	UA_dl	45	0.5556
TR	TRDistillRoberta	44	0.5432
KIS	KIS_2	41	0.5062
KIS	KIS_3	41	0.5062
UA	UA_elmo	40	0.4938
JNLP	JNLP.BERT_Multilingual	38	0.4691
KIS	KIS_1	35	0.4321
TR	TRGPT3Ada	35	0.4321
TR	TRGPT3Davinci	35	0.4321

[2] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Westermann Hannes, Savelka Jaromir, and Benyekhlef Karim. 2020. Paragraph Similarity Scoring and Fine-Tuned BERT for Legal Information Retrieval and Entailment. *COLIEE 2020* (2020).

[6] Shao Hsuan-Lei, Chen Yi-Chia, and Huang Sieh-Chuen. 2020. BERT-based Ensemble Model for The Statute Law Retrieval and Legal Information Entailment. *COLIEE 2020* (2020).

[7] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* (2019).

[8] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[10] HT Nguyen, V Tran, and LM Nguyen. 2019. A deep learning approach for statute law entailment task in COLIEE-2019. *Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE* (2019).

[11] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *arXiv preprint arXiv:2011.08071* (2020).

[12] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. [n.d.]. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. ([n. d.]).

[13] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[15] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

A Pentapus Grapples with Legal Reasoning

Frank Schilder
Thomson Reuters Labs
Eagan, Minnesota, USA
frank.schilder@thomsonreuters.com

Dhivya Chinnappa
Thomson Reuters Labs
Eagan, Minnesota, USA
dhivya.chinnappa@thomsonreuters.com

Kanika Madan
Thomson Reuters Labs
Toronto, Canada
kanika.madan@thomsonreuters.com

Jinane Harmouche
Thomson Reuters Labs
Toronto, Canada
jinane.harmouche@thomsonreuters.com

Andrew Vold
Thomson Reuters Labs
Eagan, Minnesota, USA
andrew.vold@thomsonreuters.com

Hiroko Bretz
Thomson Reuters Labs
Eagan, Minnesota, USA
hiroko.bretz@thomsonreuters.com

John Hudzina
Thomson Reuters Labs
Eagan, Minnesota, USA
john.hudzina@thomsonreuters.com

ABSTRACT

This paper describes the techniques we followed for the various tasks we participated in for COLIEE-2021 competition. There were five tasks related to legal retrieval and entailment challenges for Canadian case law (in English) and Japanese Civil code. We explain the methodology we followed for each task presenting validation results. We use a variety of techniques ranging form simple metrics such as TF-IDF word overlap to the state-of-the-art embeddings models such as BERT or GPT-3.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Probabilistic reasoning**.

KEYWORDS

natural language processing, neural networks, entailment

ACM Reference Format:
Frank Schilder, Dhivya Chinnappa, Kanika Madan, Jinane Harmouche, Andrew Vold, Hiroko Bretz, and John Hudzina. 2021. A Pentapus Grapples with Legal Reasoning. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

The Competition on Legal Information Extraction/Entailment (COLIEE) has run a challenge for extraction and entailment since 2014 that examines the decision process for both case-based and statute-based legal systems. We have participated in the challenge since 2019, and this is our third year to participate. We explored several approaches for information extraction and text entailment related to both common law and civil code. In general, both legal systems provide a rationale for a decision. For common law, judges base legal decisions on past precedents via a process known as *stare decisis*. Task 1 and 2 are examples for this kind of legal retrieval and inference process using case law documents from the Federal Court of Canada (mostly in English). For civil code, judges base legal decisions on applying one or more statutes to a given situation without altering the central legal issue, i.e., *mutatis mutandis*. Task 3 and 4 as well as the recently added task 5 use Japanese bar exam questions and the Japanese Civil Code as the basis for the retrieval and entailment/question answering task.

For each legal system, COLIEE lays out a two-step process. The first step *retrieves* the relevant cases or statutes to apply to a decision. Once the system discovers the relevant text, the second step determines if the relevant text supports or *entails* the decision. In addition to the 4 tasks from the previous years, an additional question answering task was added for the Japanese Civil Code. We participated in all of the 5 tasks. Our approaches for each task are further described in the following sections:

- **Section 2 - Task 1:** Common Law Retrieval
- **Section 3 - Task 2:** Common Law Entailment
- **Section 4 - Task 3:** Civil Code Retrieval
- **Section 5 - Task 4:** Civil Code Entailment
- **Section 6 - Task 5:** Civil Code Question Answering

2 TASK 1: LEGAL CASE RETRIEVAL

The legal case retrieval task involves reading a new case Q, and extracting supporting cases S1, S2, ... Sn for the decision of Q from the entire case law corpus. Through this section, we will call the supporting cases for the decision of a new case *noticed cases*. The case law corpus used for this task predominantly includes case law documents from Federal Court of Canada, provided by Compass Law.

The case law corpus consists of 4415 legal cases. The training set includes 650 query cases (Q) with their corresponding noticed cases (S). Unlike COLIEE-2020, for a given query case the COLIEE-2021 did not provide any candidate set of legal cases where the noticed cases are present. Previously in the COLIEE-2020 legal case retrieval task, each query was given with 200 candidate legal cases. This

COLIEE 2021, June 21, 2021, São Paul, Brazil

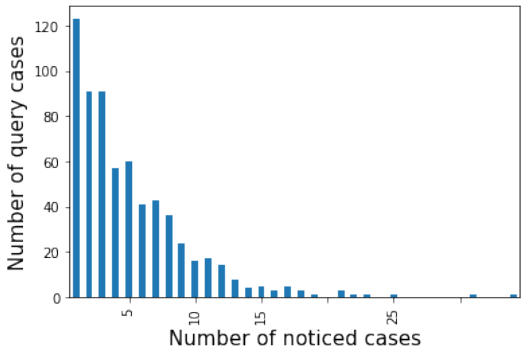

Figure 1: Frequency of noticed cases for a given case in the legal case retrieval task (Task 1).

year there was no candidate set provided per query case, making the task even more challenging. Thus, the noticed cases per query should be found in a pool of 4414 legal cases (total number of legal cases leaving out the query case), and not from a much smaller pool of 200 legal cases. The distribution of the number of noticed cases per query case is depicted in Figure 1. This frequency distribution is similar to COLIEE-2020's distribution.

2.1 Experiments

We follow a two-step process to identify noticed cases. In the first step, for each query case, we generate a subset of the entire legal corpus called the candidate set. This candidate set is supposed to include all noticed cases corresponding to the query case. In the second step, we pair the query case with each case in the candidate set. We build a binary classification model on these case pairs identifying if it is a combination of a query case and a corresponding noticed case.

Preprocessing: We processed all cases in the corpus removing all numbers, stopwords, and punctuations. We then lemmatized and lowercased each word. Additionally, we kept only the words that are greater than three characters. We decided to work only with the English contents of a file and removed all non-English words (mostly French) from the document. To this end, we removed all words that are not present in the NLTK dictionary.

Generating candidate sets: For each query we target to generate a candidate set that includes all noticed cases, keeping the size of the candidate set to the minimum. We follow three similarity measures to generate the candidate sets. The similarity measures are

(i) TF-IDF cosine similarity,
(ii) Jaccard similarity, and
(iii) LDA similarity.

Classification: After generating the candidate set, we build a binary classification model over pairs of query case and candidate cases <Q, (C_1,C_2,C_3...C_n)>. We designed the classification task such that the dataset includes all pairs that are query case and noticed cases <Q, (S_1,S_2,S_3...S_n)>, and picked a random number of query case and unnoticed case pairs <Q, ($\tilde{C}_1$,$\tilde{C}_2$,$\tilde{C}_3$...$\tilde{C}_n$)>. We developed models using logistic regression, naive Bayes, and a few tree-based classifiers.

Similarity	Rank threshold					
	20	50	100	300	500	1000
TF-IDF	0.32	0.47	0.63	0.72	0.78	0.87
Jaccard	0.29	0.38	0.50	0.57	0.65	0.75
LDA	0.21	0.35	0.54	0.66	0.77	0.86

Table 1: Ratio of noticed cases present in the candidate set with different ranking thresholds.

2.2 Results

Generating candidate sets: We begin with finding cosine similarity between pairs of a query case and all cases in the corpus. We rank the cases in the corpus, based on this similarity. Similarly, we also rank all the cases in the corpus for a given query based on Jaccard similarity and Latent Dirichlet Allocation (LDA) similarity. We present the ratio of noticed cases captured over different ranking thresholds across the three similarity methods in Table 1. Evidently, the three techniques were not able to capture all the candidate cases despite setting a ranking threshold as large as 1,000. We considered the TF-IDF similarity based candidate set for our classification experiments as they consistently capture more noticed cases over the different rank thresholds.

Classification: For each query case, we consider as candidate case the top 1,000 cases chosen using the TF-IDF cosine similarity score. Thus we generate a large dataset of query case and candidate cases pairs <Q, (C_1,C_2,C_3...C_n)>, that includes both the noticed and unnoticed cases. We treat each query case and noticed case pair <Q, (S_1,S_2,S_3...S_n)> as a positive label and each query case and unnoticed case <Q, ($\tilde{S}_1$,$\tilde{S}_2$, $\tilde{S}_3$...$\tilde{S}_n$)> as a negative label. This dataset is highly imbalanced with a very low number of query case and noticed case pairs.

Classifier	P	R	F
Decision tree	0.56	0.97	0.71
Adaboost	0.59	0.56	0.56
Naive Bayes	0.50	0.56	0.53
Xgboost	0.60	0.65	0.62

Table 2: Results on validation set when the number of noticed cases and unnoticed cases are balanced per query case.

As the legal cases are long documents, we at first decided to work with the top 10 sentences and bottom 10 sentences of each document. However, on further investigation, we realized that the last 10 sentences summarize the legal document better than the first 10 sentences. Thus we worked with only the last 10 sentences of every legal case. We worked with several traditional machine learning classifiers to extract query case and noticed case <Q, (S_1,S_2,S_3...S_n)> pairs. Each pair is represented using the concatenation of the last 10 sentences from the query case, and the last 10 sentences from the candidate case. The classifiers used a bag-of-words of trigrams and the TF-IDF vectors as features. To address the imbalance issue, we

down-sampled the number of <Q, ($\tilde{S}_1$,$\tilde{S}_2$,$\tilde{S}_3$...$\tilde{S}_n$)> pairs. Initially, we kept all <Q, (S_1,S_2,S_3...S_n)> pairs and chose <Q, ($\tilde{S}_1$,$\tilde{S}_2$,$\tilde{S}_3$...$\tilde{S}_n$)> pairs ten times the total number of <Q, (S_1,S_2,S_3...S_n)> pairs. However, we noticed that the classifiers struggled, predicting everything as a negative pair. Hence, we downsized the <Q, ($\tilde{S}_1$,$\tilde{S}_2$,$\tilde{S}_3$...$\tilde{S}_n$)> pairs by keeping them equal to the number of <Q, (S_1,S_2,S_3...S_n)> pairs. The classification results on the validation set over various classifiers are presented in Table 2.

We followed a similar process on the test set. First, for each query case, we generated the candidate set as the top 1000 most similar cases using the TF-IDF similarity. Next, we generate pairs of cases by using the query case and each case candidate set. Next, we classify these pairs to extract noticed cases by predicting via the model build using the Xgboost classifier. We chose the Xgboost classifier over the decision tree classifier despite decision tree achieving a better F1 score (F1: 0.56 vs. 0.62), because we target a higher precision model (0.60 vs. 0.56). Our hypothesis is that a higher precision model will work well in an imbalanced label distribution setting. We build the Xgboost model with the *gbtree* booster, setting a maximum depth of 6. We set the learning rate to 0.3, following *uniform* sampling. As the classifier was built on a balanced dataset and our dataset is highly imbalanced, we submitted as noticed cases all candidate cases with a probability greater than 0.995. However, we could achieve an F-score of only 0.0047 in the test set. We ranked 11th out of the 15 submissions for task1.

As the results indicate, the task is challenging with the highest performance in the leader board with F1 0.1917. One of the reasons that made the task very challenging is the absence of candidate set for every query case unlike previous years. Apart from the text in the case document, meta-data like the year the case was released might help in improving the performance.

3 TASK 2: LEGAL CASE ENTAILMENT

Task 2 is an entailment task for legal documents. It involves the identification of a paragraph from existing cases that entails the decision of a new case. Given a decision Q of a new case and a relevant case R, a specific paragraph that entails the decision Q needs to be identified. The provided dataset was designed to make sure that the answer paragraph can not be identified merely by information retrieval techniques using some examples. Because the case R is a relevant case to Q, many paragraphs in R can be relevant to Q regardless of entailment. This task required identification of a paragraph which entails the decision of Q, requiring building a specific entailment method to compare the meaning of each paragraph in R and Q to make an entailment decision. The data is drawn from an existing collection of predominantly Federal Court of Canada case law.

3.1 Experiments and Results

The training dataset consists of 425 new cases, to each is associated a relevant case with a variable number N of paragraphs. The frequency of the number of paragraphs per relevant case is shown in Fig.2. Out of N paragraphs (x-axis), 4 paragraphs at most entail the decision fragment in the training dataset.

Our approach consists in reducing the number of potential candidate paragraphs as a ranking allowing us to then train a classifier

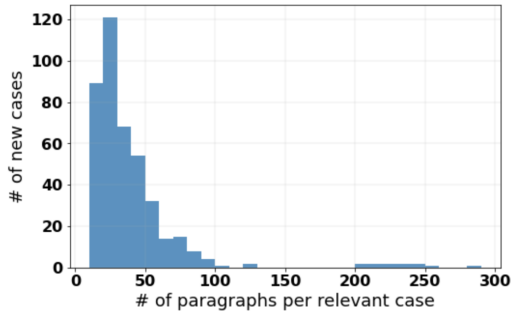

Figure 2: Count of relevant paragraphs vs. number of new cases

on a relatively balanced dataset. Similarity scores are computed between each paragraph and the decision fragment using different vectorizers, such as n-gram vectors, universal sentence encoder vectors, averaged word embedding vectors. A set of K paragraphs is selected from each relevant case as most similar to the decision fragment. The training dataset consists of 425 cases, 42 of which are isolated for test. A random forest classifier is then trained on the filtered subset of datapoints. To evaluate the generalization performance and do hyperparameter selection, among which the selection of K, we used a multi-fold cross validation procedure. The final best performing model was used to do inference on the test set provided with the task. F1-score of 0.56 has been obtained on our test set of 42 cases, which is close to 0.54, the result obtained on the unseen test (ranked 13 in the competition).

3.2 Discussion

The approach is simple as it uses hand-crafted similarity features and applies a classical random forest classifier; It is however robust and provides a baseline for more advanced techniques. Further improvements can be obtained by using different similarity metrics that focus on specific parts of the text, such as named entities, and structural similarity using part-of-speech tags.

The high class-imbalance nature of the problem and the limited data size make it challenging to train an efficient and generalizable neural-network classification model. In addition, conventional supervised classifiers fail to understand the reasoning a human being follow when facing a textual entailment problem. Zero-shot learning is a potential alternative to supervised learning, in situations where not-enough annotated data is available, and therefore could provide an alternative generalizable approach. Text-to-Text models, such as T5 (the top winner of this task), are also promising alternatives to Text-to-Label models since they can be fine-tuned in an unsupervised fashion and do not require availability of annotated data.

4 TASK 3: CIVIL CODE RETRIEVAL

Task 3 involves selecting the most relevant Japanese Civil Code articles needed to answer given legal questions. The questions are taken from Japanese legal bar exams and given in the form of yes/no questions. More than 70 percent of the questions can be answered by a single article, but some questions need multiple articles to infer

the answer, while some questions have multiple associated articles each of which can answer them independently. Selecting the right number of articles for each question is important.

4.1 Experiments

4.1.1 Word Mover's Distance. Last year, we have tried a TF-IDF based approach, but the challenge was that TF-IDF requires exact words to be present in both in the question and article pairs to obtain a high similarity score. It performed poorly especially for pairs that had the same meaning but few overlapping words. As an attempt to overcome the issue, this year we tried Word Mover's Distance (WMD). WMD allows us to assess the similarity between 2 texts even when there are no words in common.

While it solves many of the problems with TF-IDF, it still strongly favors word overlap over closeness of overall meanings. Consider the example, "The kids ate the fish for dinner."

We measured the distance of the following sentences.

id	Sentence	distance
1	The fish ate the kids for dinner.	0
2	The children had seafood for supper.	0.5679

Table 3: WMD distance to the anchor sentence

The second sentence is obviously closer in meaning, but the first sentence has the distance of zero since all the words are in common. For this reason, even though WMD is effective in retrieving relevant candidates, it is not enough to rank the most relevant ones as the highest matches.

This also makes thresholding extremely challenging. As mentioned, it is very important to select the right number of articles for each question. Since the vast majority of the questions have at most 2 correct answer (In the R02 test set, 65 had 1 answer, 14 had 2 answers, and 2 had 4 answers), in most cases listing more than 2 candidates could lead to immediate precision loss. For this reason, we limited our submission to 2 candidates per question. However, we were not successful in bringing the most relevant ones within the top 2. We will need a better ranking mechanism for next year.

4.1.2 Transformer Models. Transformer architectures like BERT, RoBERTa, T5, etc. have demonstrated powerful performance in natural language inference (NLI). For this reason, we attempted to fine-tune a pre-trained Japanese BERT entailment model by performing pairwise sequence classification with the queries and the articles corresponding to the queries as positive examples and all others as negative examples. Though the number of COLIEE queries is quite small, appending them to each article in the list of articles leads to a much larger set of data. A subset of the training data was reserved for validation, and the model was optimized to maximize the area under the ROC curve (AUC) for the validation data. Due to the size of the dataset and the lack of diversity in the text, training was a lengthy process with diminishing returns.

Due to the lower than expected performance of the model with respect to the validation data, we decided to add in the spaCy large language model to the system[5]. The system determined entailment by varying the BERT score along with the maximum cosine similarity between the query and the sentences in the articles until a maximal validation F2 score was achieved. Finally, the cosine similarity metric was used to sort the entailing articles because of its superior discriminative power as compared to the BERT model.

4.2 Results

All of our attempts at producing an entailment system proved unsuccessful. The WMD system was not sophisticated enough to understand entailing properties beyond word overlap, and the BERT system was not able to generalize to the COLIEE language domain with the few examples in the training set. The results can be seen in table 4.

Team	F2 Score	Precision	Recall	MAP
OvGU_run1	0.6749	0.7778	0.7496	0.7525
JNLP. CrossLMultiLThreshold	0.6000	0.8025	0.7947	0.7822
BM25.UA	0.7092	0.7531	0.7037	0.7555
JNLP.CrossLBertJP	0.6241	0.7716	0.7783	0.8218
R3.LLNTU	0.6656	0.7438	0.7875	0.7921
R2.LLNTU	0.6770	0.7315	0.7893	0.7822
R1.LLNTU	0.6368	0.7315	0.7893	0.7822
JNLP.CrossLBert JPC15030C15050	0.5535	0.7778	0.7737	0.8119
OvGU_run2	0.4857	0.8025	0.7571	0.7525
TFIDF.UA	0.6790	0.6543	0.7306	0.7228
LM.UA	0.5460	0.5679	0.5432	0.6422
TR_HB	0.5226	0.3333	0.6173	0.6625
HUKB-3	0.5224	0.2901	0.6975	0.6100
HUKB-1	0.4732	0.2397	0.6543	0.6128
TR_AV1	0.3599	0.2622	0.5123	0.4653
TR_AV2	0.3369	0.1490	0.5556	0.4346
HUKB-2	0.3258	0.3272	0.3272	0.4167
OvGU_run3	0.1570	0.7006	0.5557	0.5743

Table 4: Task 3 Results

OvGU_run1 is the first place winner of task 3, TR_HB is the WMD model, TR_AV1,2 are the composite BERT/spaCy models. Overall we ranked the 5th among the 6 teams competed. As seen from table 4, the first place winner surpassed all of our attempts in every metric. It is interesting to note that the simpler WMD model surpassed the more sophisticated deep learning systems, suggesting that the deep learning systems suffered from high variance. This makes sense when considering that the BERT model was thoroughly fine-tuned on a very small query set and article set. The lesson learned from these task results is that simple models using bag of word features can offer adequate initial models which generally have a high bias. Higher complexity models which operate on latent text representations can offer increased variance in prediction quality, but it is important to perform domain transfer on a much larger set of data, rather than simply fine-tuning on the COLIEE training data.

5 TASK 4: CIVIL CODE ENTAILMENT

Before last year, the entailment task has resisted deep learning approaches because of the large premise sizes and the relatively

small training set size. Last year, multiple teams applied language models [11] including BERT [4] and T5 [12]. In a follow-up on the transformer-based work, we explored several pre-training approaches that included multi-sentence entailment [15], Japanese entailment with Electra [3], and an ensemble with T5 [12].

Although last year's task 4 winner leveraged BERT [11], civil code entailment remains challenging because the system must evaluate several inter-dependent articles $[P_1, P_2, \ldots P_n]$ against a statement H to determine if H is true or false. Each civil code article represents a set of conditions, exceptions, and conclusions. H represents a set of facts and a conclusion. The system must apply the facts to the articles' conditions and determine if H came to the correct conclusion [7]. Table 5 shows a single sentence example matching the conditions with facts. In contrast, standard entailment datasets, like MultiNLI, remain comparatively simple because they focus on single sentence entailment [16].

Sentence	Sentence Text
P_1	A mandate shall terminate when the mandator or mandatary dies.
H	The mandate terminated upon the mandator's death .

Table 5: Conclusions , Conditions , and facts

In addition to the multiple sentence entailment, deep learning approaches have performed especially poorly on the Japanese Civil Code entailment tasks because the training set size is comparatively small to similar entailment tasks. Table 6 compares COLIEE 2021 civil code entailment data set to other single and multiple sentence data sets. The COLIEE data is 2 to 5 times smaller than the multiple sentence data set and 1000x smaller than the single sentence data sets. The COLIEE data set is therefore too small for supervised machine learning alone [18].

Dataset	MultiNLI	MultiRC	OpenBookQA	**COLIEE**
Train	392,702	5,131	4,957	**696**
Dev	20,000	4,848	500	**112**
Test	20,000	4,583	500	**80**

Table 6: Entailment data set sizes.

This year, we investigated two transfer-learning approaches and an ensemble in an attempt to address the premise length and data set size issues. We then implemented a multi-sentence Natural Language Inference (NLI) model, Multee [15], that applies transfer learning from a single-sentence NLI dataset to the more complex entailment task at hand.

5.1 Experiments

5.1.1 Multiple Sentence NLI. While Multee [15] no longer tops MultiRC leader-board [6], this model's architecture fits the civil code entailment task in that it weights the relevant clauses in the articles individually before applying attention to the article as a whole. Multee trains the model in two phases: single sentence pre-training and multiple sentences entailment. The first phase remained utterly unchanged compared to Multee's original experiments [15]. This phase pre-trained against the relatively massive SNLI and MultiNLI data sets and produced weights used in the second training phase.

The second phase required some modifications to Multee's original experiments because the COLIEE data set is not multiple choice. The COLIEE model implemented a binary cross-entropy loss function for a single hypothesis. The second phase's training set includes both COLIEE and OpenBookQA examples because the OpenBookAQ provided a useful generalized multi-sentence entailment data set with complete sentences in the hypothesis. To make both data sets consistent with each other, we made two changes. First, we labeled the COLIEE example as entailment and neutral for the Y and N, respectively. Second, we converted the OpenbookQA pairs in a single hypothesis format.

Both phases leveraged GloVe word embedding (680 Billion Words, 300 dimensions) instead of contextual embedding (i.e., BERT-style embeddings). Although the Multee weights each sentence, the model still performs cross-attention over the entire premise passage. The GloVe embeddings were chosen to avoid a truncated premise.

Once the second phase completed, we evaluated again the year R01 as a validation set. The validation set prediction accuracy was 0.6250.

5.1.2 Electra. Transfer learning is the task of using a model which has been trained on one task as a starting point for training on another task. The pre-training of transformer models on general corpora to be used in downstream tasks in any domain is one of the most popular methods of transfer learning in natural language processing. It has been shown that transformer models pre-trained on a general English corpus for a language modeling task lead to state-of-the-art results on other tasks such as classification, text summarization, question answering, etc. [4]. For this reason, it was hypothesized that a similar approach could be taken, except with a Japanese transformer model.

Choosing a pre-trained transformer model as a starting point for transfer learning is a challenging task, for there exist many transformer architectures, pre-training corpora, and pre-training tasks. As a rule of thumb, one can choose a model architecture produced by the research group which developed the architecture and training technique. In Japanese, however, this is difficult because the transformers have to be pre-trained by other research groups with a different corpus and training conditions, so the quality of the models and their potential for success on downstream tasks is less understood. Not only that, tokenization for Japanese is less straightforward, leading to large variance in the tokenization patterns among off-the-shelf tokenizers.

Since choosing a Japanese pre-trained transformer is not a trivial task, we propose a transformer selection process to identify the most promising architecture/tokenizer candidate. The method for obtaining the model consists of measuring the embedding distribution differences between the positive and negative examples from the Japanese training data provided for task 4. The metric used to measure this distance is the Jensen-Shannon Divergence (JSD) which is as follows [9]:

$$JSD(P||Q) = \frac{1}{2}\big(KSD(P||Q) + KSD(Q||P)\big) \quad (1)$$

where KSD is the Kullback–Leibler Divergence given by [8]:

$$KSD(P||Q) = \mathbf{E}\Big(P(x)\log(\frac{P(x)}{Q(x)})\Big) \quad (2)$$

where P(x) and Q(x) are the distributions of the length averaged embedding layer outputs for the positive and negative examples.

With JSD as a measure of transformer embedding layer output distance, we browsed various models from the huggingface transformers repository [17]. An Electra Small Discriminator produced by Cinnamon Inc. [3] was found to have the largest JSD between the positive and negative examples for its length averaged embedding layer output.

With a randomly initialized classification head on the transformer, initial training performance was unstable, suggesting that the features extracted from the attention layers did not effectively represent legal language. Also, since the COLIEE training set is so small, and domain transfer of language models takes large amounts of data, we decided to build up the model layer by layer, until a plateau in our evaluation metric was reached. We began by using the Electra embedding layer and its first layer and trained a classifier head on top of it. Once the training loss plateaued, we added the next layer and continued training. This process was repeated until the area under the ROC curve (AUC) plateaued. This occurred after training the third attention layer. We achieved a validation AUC score of 0.72.

Though the validation metrics suggested strong model performance, the shortened Electra model performed poorly on the test set with a testing accuracy of 0.5062, a lower accuracy than guessing positively for every query. There are many reasons as to why the model performed much worse on the test data. The most likely reason is that the model was overfit to the provided data. During training, the model was validated against the queries which appeared in one year's exam. Whenever the validation AUC score improved, the model was saved. This sort of model checkpointing is evidently not appropriate for tiny validation sets. The small set of data from the validation split occupies a sliver of the topic space in the legal domain. Moving forward, it would be prudent to perform pre-training on Japanese legal data, and then expand the validation set size for classification fine-tuning.

5.1.3 T5-based Ensemble. We conducted three trials with T5 to determine if legal specific pre-training could improved upon an generic pre-trained model (t5-base). Given that pre-training on domain-specific corpora tend to over-fit [12], we required an alternative submission if the pre-training experiments failed. If we could not improve upon T5-base's validation performance, then we planned to submit an ensemble model using the T5 baseline model as an alternative means to improve accuracy.

For each trial, we fine-tuned on the COLIEE dataset using the year Heisei 30 (H30) for early stopping and Reiwa 1 (R01) as the validation sets. Each trial used the same hyper-parameter setting for fine-tuning. The trials stopped after 3 epochs with no accuracy improvement in the H30 development set. The learning rate rate was 5e-5 and the batch size was 16.

The main differences in each trial are as follows. The first trial evaluated the T5-base embedding and only fined-tuned on the COLIEE dataset. The second trial added a span corruption fine-tuning tasks on English translations of Japanese, Quebec, and Louisiana Civil Code Statues. The final trial pre-trained legal specific embeddings from scratch and then fine-tune on the COLIEE tasks.

The final trial attempted to pre-train on relevant content and task types to build an embedding from scratch. The pre-training use the same hyper-parameter setting as T5-base: 768 dimension encoding layers, 64 dimension key, query, value projections, 3072 dimension feed forward layers, 12 hidden layers, 12 decoder layers, 12 attention heads, 32 relative attention buckets. Unlike the t5-base model a new tokenizer was trained from civil code statues and US case summaries with a 16000 word-piece vocabulary. The pre-training ran the following tasks with included both legal content and general entailment tasks:

- **Span Corruption:** Japanese Civil Code
- **Span Corruption:** Civil Code from Quebec & Louisiana
- **Span Corruption:** 796K US Legal Summary Sentences
- **Natural Language Inference:** MNLI & MuiltRC
- **Span Extraction:** Requisite and Effectuation Extraction [10]

Table 7 shows the results for the initial trials. Neither legal specific approach could improve upon the generalized T5-base model. As a result, we submitted an majority vote ensemble using the COLIEE fine-tuned T5 model, Multee, and Electra.

Trial	Dataset	Val. Acc.
T5-base Fine-Tuning	COLIEE	0.5405
Civil-Code Fine-Tuning	COLIEE + Civil Code	0.5135
Scratch	Legal NLI Multi-task	0.46847

Table 7: T5 Trials

5.2 Results

Table 8 show our results from the R02 test set. While both Multee and the ensemble models score above the baseline, neither accurately captured the communicative intent of the legal sentence (placing tied for 5th overall). Our approaches leveraged language model and or attention based mechanisms to model the similarity between the hypothesis and premises. While these approaches appear mimic legal reasoning by capturing similarity between the premise and hypothesis, they do not capture the Article's intent [1].

All our language experiments represent attempt model legal knowledge based on text, either legal or generalize content. While we attempt to incorporate additional legal text in civil and case law with pre-training, the legal specific embedding only lead to over-fitting of the model (Table 7).

6 TASK 5: CIVIL CODE QUESTION ANSWERING

The newly introduced question answering task is based on task 3 and 4 and takes the query from task 4 without any information about the relevant articles. While training, relevant articles and other data sources could be used but at test time, only the query is given. Note that the hypothesis is not a question but a statement that is determined to be true or false.

Team	sid	Ranking	Correct	Accuracy
BaseLine	Yes 43/All	-	43	0.5309
BaseLine	No 38/All	-	38	0.4691
HUKB	HUKB2	1st	57	0.7037
TR	TR-Ensemble	5th tied	48	0.5926
TR	TR-MTE	5th tied	48	0.5926
TR	TR_Electra	9th	41	0.5062

Table 8: Task 4 Results Compared to Best Run

In our experiments, we wanted to explore whether a recently released massive language model (LM) called GPT-3 [2] may contain the ability to do complex legal reasoning. GPT-3 has shown impressive results in generating fiction stories as well as showing a strong performance in question answering tasks. The trained model is massive and contains 175B parameters.

The cost for training GPT-3 was very high but using it in the generation mode is computationally not very expensive. The output of the GPT-3 model can be achieved via the OpenAI[1] interface and one has to pay for the number of tokens used as prompt and the tokens being generated.

The GPT-3 api offers different engines that vary with respect to the parameters used for training. The largest model (davinci) is also the most expensive model whereas the ada model was the least expensive but fastest one ($0.06 vs. $0.0008/token).

In contrast to this massive LM, we used also a sentence based LM [13] based on siamese and triplet network structures that provides efficient sentence similarity metrics for sentences. We used the distilled version of RoBERTa [14] for paraphrase detection as another baseline system relying on a LM that was not trained on any domain-specific data. The model is smaller than the original RoBERTa-base model and is significantly smaller than the GPT-3 model (distillroberta-base has only 6 layers, 768 dimensions and 12 heads, totaling 82M parameters). Hence, it was not necessary to run the experiments on a GPU, a CPU was sufficient for running our experiments.

Our experiments showed that

- a GPT-3 based model is not capable of doing legal question answering and cannot beat the random baseline.
- there is no difference between the smallest GPT-3 model (i.e., ada) and the largest model trained with 175 billion parameters (i.e., davinci).
- a sentence-based RoBERTa model shows a small improvement over the baseline approach.

6.1 Experiments

6.1.1 GPT-3. In order to utilize the GPT-3 model with a few-shot learning model, we (a) created a short prompt, (b) transformed the query into an actual question and (c) added the answer by starting with *Yes* or *No* and then continuing with an explanation.

The explanation was basically the relevant articles that were mentioned in the training data. We created two question/explanation pairs and used this text as input for the GPT-3 model in order to allow for a few-shot learning scenario (See Figure 3).

[1]https://beta.openai.com/

This bot answers Japanese bar exam questions.

Question: Is it true that a special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty?
Answer: Yes, because even if the seller makes a special agreement to the effect that the seller does not warrant in the case prescribed in the main clause of Article 562, paragraph (1) or Article 565, the seller may not be released from that responsibility with respect to any fact that the seller knew but did not disclose, and with respect to any right that the seller personally created for or assigned to a third party.

Question: Is it true that there is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage?
Answer: No, because the provisions of the preceding three Articles apply mutatis mutandis if the right transferred by the seller to the buyer does not conform to the terms of the contract (including the case in which the seller fails to transfer part of a right that belongs to another person).

Figure 3: Prompt for GPT-3 triggering 2-shot learning

Given a prompt and and two examples, GPT-3 achieves state-of-the-art results for trivia question answering, but legal question answering is too complex for the capabilities of this LM trained on general web content that likely contained some legal content. Future research could focus on pre-training GPT-3 (or GPT-2) with specific legal content in order to achieve a better result.

Although the largest model trained with 175B parameters (i.e., davinci) did not outperform the smaller model (i.e., ada), we saw differences in the fluency and even direct references to the Japanese Civil code. The following example explanations generated produced both the correct answer (i.e., *Y*) but the explanation is incorrect although the davinci model sounds as if it could be a valid answer.

Question: In the cases where a lessor is the owner of the subject matter in a contract for lease, the ownership of the subject matter shall not be transfered to the lessee, on the other hand, in the cases where a lender is the owner of the subject matter in a contract for loan for consumption, the ownership of the subject matter shall be transfered to the borrower.

ada: Yes, because the provisions of the preceding two Articles apply mutatis mutandis if the right transferred by the seller to the buyer does not conform to the terms of the contract (including the case in which the seller fails to transfer part of a right that belongs to another person)

davinci: Yes, because in Article 579, paragraph (2), if a contract for lease is not terminated before the expiration of the term

Team	sid	Correct	Accuracy
BaseLine	No 43/All	43	0.5309
BaseLine	Yes 38/All	38	0.4691
JNLP	JNLP.NFSP	49	0.6049
UA	UA_parser	46	0.5679
JNLP	JNLP.NMSP	45	0.5556
UA	UA_dl	45	0.5556
TR	TRDistillRoberta	44	0.5432
KIS	KIS_2	41	0.5062
KIS	KIS_3	41	0.5062
UA	UA_elmo	40	0.4938
JNLP	JNLP.task5.BERT_Multilingual	38	0.4691
KIS	KIS_1	35	0.4321
TR	TRGPT3Ada	35	0.4321
TR	TRGPT3Davinci	35	0.4321

Table 9: Task 5 results

> of the contract, the lessor has the right to cancel the lease and to reclaim the subject matter, however in the case of lender and borrower concerning a contract for loan for...

6.1.2 DistillRoBERTa. We utilized the sentence BERT implementation[2] that contained the pre-trained DistillRoBERTa LM trained on paraphrase text. Given a question, the article with the highest similarity score was determined. Based on previous data we determined a threshold score that we used for the new test run (i.e.,> 0.728).

6.2 Results

The results on the test data set show that GPT-3 does not encode any complex legal knowledge that is required for this inference task. Moreover, there was no difference between the GPT-3 system with the highest number of parameters (i.e., 175 billion) and the smallest number of parameters. This would indicate that an even larger (and more expensive to train) LM may also not lead to a better performing system for this tasks.

The result for the sentence-based BERT system, on the other hand, shows a reasonable performance going slightly beyond a random baseline and our team reached 3rd place overall for this task with this submission. Given that the system does not incorporate any legal knowledge or was not pre-trained on legal data, this straight-forward similarity based approach has some promise and future research should explore how more domain-specific sentence based LMs could improve the current performance of such a system.

7 SUMMARY

We participated in all 5 tasks this year including the new question answering task 5. We tested various baseline system ranging from text similarity features to using massive LM such as GPT-3 with few-shot learning to further pre-training and fine-tuning T5.

For task 1, We follow a two-step process. In the first step, for each query case, we generate a subset of the entire legal corpus called the candidate set. This candidate set is supposed to include all noticed cases corresponding to the query case. In the second step, we pair the query case with each case in the candidate set. We build a binary classification model on these case pairs identifying if it is a combination of a query case and a corresponding noticed case.

For task 2 on legal case entailment, we combined various textual similarity metrics and empirically selected a score in order to identify the best K paragraphs. A random forest algorithm was trained on these K paragraphs in order to solve the entailment problem.

The Japanese civil code retrieval task 3 was addressed by our solution via a new similarity metric in contrast to last year's attempt of using TF-IDF based cosine similarity. This year, we used the Word Mover's Distance (WMD) in order to capture semantic similarity even without any word overlap necessary between two sentences.

As a second approach to task 3 we also experimented with a transformer-based system by fine-tuning Japanese BERT on past data consisting on article/query pairs defined as a binary classification task. In addition, we also explored the use of the spaCy large language model.

For task 4, the Japanese Civil Code entailment task, we tested out several pre-training approaches that included multi-sentence entailment [15], Japanese entailment with Electra [3], and an ensemble with T5 [12].

Finally, we used two LMs for task 5 of Japanese Civil Code question answering. The first LM was GPT-3, a massive LM with up to 175B parameters, and the second one was a distilled version of the RoBERTa-base LM. The GPT-3 based system consists of a few-shot learning approach to generating postive/negative answers with an explanation, whereas the DistillRoBERTa based approach made the entailment decision based on a learned threshold.

ACKNOWLEDGMENTS

We would like to thank the authors of "Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts" [10], Truong-Son Nguyen, Le-Minh Nguyen, Satoshi Tojo, Ken Satoh and Akira Shimazu for their generosity to make their RRE data publicly available.

REFERENCES

[1] Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 5185–5198. https://www.aclweb.org/anthology/2020.acl-main.463

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[3] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. arXiv:2003.10555 [cs.CL]

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (Oct. 2018). http://arxiv.org/abs/1810.04805 arXiv: 1810.04805.

[5] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python.* https://doi.org/10.5281/zenodo.1212303

[6] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 252–262. https://doi.org/10.18653/v1/N18-1023

[7] Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2019. Statute Law Information Retrieval and Entailment. In *Proceedings of ICAIL 2019.* https://doi.org/10.1145/

[2] https://github.com/UKPLab/sentence-transformers

3322640.3326742
[8] Solomon Kullback. 1959. *Information Theory and Statistics*. Wiley, New York.
[9] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* 37, 1 (Jan. 1991), 145–151. https://doi.org/10.1109/18.61115
[10] Truong-Son Nguyen, Le-Minh Nguyen, Satoshi Tojo, Ken Satoh, and Akira Shimazu. 2018. Recurrent Neural Network-Based Models for Recognizing Requisite and Effectuation Parts in Legal Texts. *Artif. Intell. Law* 26, 2 (June 2018), 169–199. https://doi.org/10.1007/s10506-018-9225-1
[11] Juliano Rabelo, Kim Mi-Young, Randy Goebel, Yoshioka Masaharu, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. In *Proceedings of the International Workshop on Juris-Informatics 2020 (JURISIN 2020)*. 114–127.
[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html
[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084
[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108 (2019).
[15] Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing Entailment for Multi-Hop Question Answering Tasks. In *Proceedings of NAACL 2019*. https://doi.org/10.18653/v1/N19-1302
[16] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of NAACL-HLT 2018* (New Orleans, Louisiana). http://aclweb.org/anthology/N18-1101
[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6
[18] Masaharu Yoshioka, Yoshinobu Kano, Naoki Kiyota, and Ken Satoh. 2018. Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018. https://sites.ualberta.ca/~rabelo/COLIEE2019/COLIEE2018_SL_summary.pdf

Using Contextual Word Embeddings and Graph Embeddings for Legal Textual Entailment Classification

Sabine Wehnert
sabine.wehnert@gei.de
Georg Eckert Institute
Leibniz Institute for International
Textbook Research
Germany
Otto von Guericke University
Magdeburg
Germany

Shipra Dureja
Libin Kutty
Viju Sudhi
<firstname>.<lastname>@st.ovgu.de
Otto von Guericke University
Magdeburg
Germany

Ernesto W. De Luca
deluca@gei.de
Georg Eckert Institute
Leibniz Institute for International
Textbook Research
Germany
Otto von Guericke University
Magdeburg
Germany

ABSTRACT

Textual entailment classification is one of the hardest tasks for the Natural Language Processing community. In particular, working on entailment with legal statutes comes with an increased difficulty, for example in terms of different abstraction levels, terminology and required domain knowledge to solve this task. In course of the COLIEE competition, we develop two approaches to classify entailment. The first approach combines Sentence-BERT embeddings with a graph neural network, while the second approach uses the domain-specific model LEGAL-BERT, further trained on the competition's retrieval task and fine-tuned for entailment classification. In this work, we discuss why of all our submissions, the LEGAL-BERT runs may have outperformed the graph-based approach.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Information systems** → **Document representation**; *Language models*; *Similarity measures*; Question answering; • **Computing methodologies** → Neural networks.

KEYWORDS

contextual word embeddings, graph embeddings, entailment classification

1 INTRODUCTION

In this work, we develop two approaches for legal textual entailment classification on the English version of the Japanese Civil Code. This research is part of task 4 of the Competition on Legal Information Extraction/Entailment (COLIEE). The task consists of two texts which are compared to decide on a binary entailment relationship. In this case we have a query and one or multiple associated articles from the English version of the Japanese Civil Code.

In general, textual entailment classification requires capabilities which are normally attributed to humans who can acquire a deep knowledge of the legal domain to understand and interpret legal texts to reason about their relationship and lawfulness. Such reasoning capabilities are yet to be developed on a machine, for example as a decision support in specific legal cases. International activities make it hard - even for legal professionals - to oversee all legislations which may be relevant for a specific case and to determine compliance with the law via entailment. Therefore, we perform research on this topic towards the goal of pushing the limits of current Legal AI approaches. With the advent of deep learning, there are many models which are tested on natural language inference tasks, and the same development exists in the COLIEE competition. Although their decision making is hard to understand for a human, deep learning approaches have consistently achieved good results in the past years on this task. They are often outperforming more explainable methods and because they are not trusted in the legal domain, the research and detailed analysis of their strengths and weaknesses is important to understand future research directions.

In particular, the BERT model (Bidirectional Encoder Representations from Transformers) has achieved good scores in the past COLIEE editions. Aside from ongoing state-of-the-art performance of BERT variants on many tasks in natural language processing, BERT offers contextual word embeddings which are an advancement of distributional semantic approaches. Previous approaches often failed to correctly encode the contextual meaning of a word. Therefore, using BERT in the COLIEE competition to overcome challenges, such as term mismatch and different abstraction levels of the two documents to be compared can be helpful. Therefore, we rely in both our approaches on some variant of the BERT model. Nowadays, it is almost a standard procedure to choose a domain-specific pre-trained BERT model and then to fine-tune it on a downstream task.

Using only a BERT model though will not solve this task and has been done before. Our work is motivated by the recent advancements in graph neural networks, which can also be combined with the BERT model. Since the relationship between the query and article is a relevance relationship, the data can be transformed into a graph format to encode structural information about the relationships between nodes and their individual features. Another major challenge in the COLIEE competition is the size of the dataset, with 806 instances to train a model. A commonly mentioned drawback of deep neural networks is the data size which is required to learn meaningful feature representations. In our work, we study also data

COLIEE 2021, June 21, 2021, Online

augmentation and enrichment to work with our graph-based and BERT-based deep learning approaches despite the small dataset size. Based on our experiments, we offer insights in our result analysis, discuss some challenges we faced in this competition and draw conclusions for future research.

With this paper, we make the following contributions:

- We employ an ensemble of graph neural networks together with features from Sentence-BERT and metadata of the Civil Code for the task.
- We perform pre-training on the statute law retrieval task and data decomposition to improve the learning of a domain-specific model called LEGAL-BERT.

After the introduction, the remainder of this work is structured as follows: In Section 2, we collect approaches of contextual word embeddings and graph embeddings for entailment classification. In Section 3, we describe the concepts of our approaches for the ensemble of Graph neural networks (GNNs) in our submitted run 1 and LEGAL-BERT in run 2 and 3 of the competition. Section 4 contains our evaluation setting, results and their analysis within a discussion. We conclude this work and indicate future research in Section 5.

2 RELATED WORK

2.1 Contextual Word Embeddings

With the growth in applications for Natural Language Processing (NLP), various fields of software technology such as machine translation, text recognition and text generation have seen a large development in the area of deep learning models adapting to these tasks [7]. Substantial progress in the area of learning embeddings for dense representations of the textual data has been made. Some of them are CoVe [12] (Contextual Word Vectors), ELMo [15] (Embeddings from Language Model), Cross-View Training [3] (CVT), ULMFiT [8] (Universal Language Model Fine-tuning for Text Classification), GPT [17] (Generative Pre-training Transformer), BERT [4], ALBERT [10] (A Lite BERT) and RoBERTa [11] (Robustly optimized BERT approach). These dense representations are usually learned by training on auxiliary tasks, such as masked language modeling (MLM), next sentence prediction (NSP), machine translation, and transcription. Contextual word embeddings - once learned - can be further fine-tuned for downstream tasks, such as classification, with relatively less effort. The language model BERT [4] and its variants [10, 11] have emerged as the most convenient choice for a model concerning these downstream tasks since they condition a word's embedding on the surrounding context. This makes them and similar approaches perform significantly better than other models which learn static embeddings as a dense representation of the textual data.

One noticeable characteristic of BERT is that it performs better on domain-specific tasks when pre-trained with data of that specific domain. Various examples include *BioBERT-cased*, *PubMedBERT-uncased* which both perform better for bio-medical data than the original BERT model as discussed by Gu et al. [5]. Similarly, LEGAL-BERT and its variants on legal sub-domains can perform better than the standard BERT on domain-specific tasks as summarized by Chalkidis et al. [2]. For LEGAL-BERT pre-training is carried out on a collection of several fields of English legal text like contracts, court cases, and legislation. The *legal-bert-base-uncased* model [2] is similar to the standard English *bert-base-uncased* model [4] in its neural network architecture. It has 12 layers, 768 hidden units, 12 attention heads, and 110M parameters. The pre-training is carried out with 1 million training steps on batch sizes of 256 with a maximum sequence length of 512 starting with a learning rate of 1e-4. It also has a similar training procedure to *bert-base-uncased*.

In the previous year of the COLIEE competition, several teams used pre-trained BERT-based models with variations to address this entailment task [16]. The team *CU* submitted two such models. For the first run, they selected the *bert-multilingual-cased* model for sequence classification and then fine-tuned it on training data provided by COLIEE organizers. The other model was additionally trained on articles obtained from a term frequency - inverse document frequency (TF-IDF) model for the retrieval task 3. A closely related approach is the submission of Team *CYBER*. They use a pre-trained RoBERTa instead of *bert-base-uncased* and fine-tune it on the SNLI dataset followed by the COLIEE dataset to entail relevant articles. The team JNLP [14] also focused on the BERT-based approach to submit three runs, one of them gained the winning accuracy in COLIEE 2020 for task 4. We use their winning run as a motivation for our runs 2 and 3. They use domain-specific pre-training of BERT with American case law data with a corpus of 8.2M sentences. This model was then fine-tuned for addressing the lawfulness classification problem using additional augmented data of the English version of the Japanese Civil Code and COLIEE training data. To summarize this part, the choice of a suitable pre-trained model with subsequent fine-tuning can have a big impact on the success of a BERT-based approach for entailment classification.

2.2 Graph Embeddings

Graph neural networks (GNNs) are popular for data which can be represented by relations in a graph format. GNNs are used in many fields, such as computer vision, natural language processing and combinatorial optimization. In the field of NLP, GNNs solve tasks such as text classification, question answering and entity retrieval. Yao et al. [20] have used a graph convolutional network for text classification by forming a graph from word co-occurrences and document-word relations. They achieve scores which beat state-of-the-art methods on standard text classification benchmark datasets. De Cao et al. [1] use GNNs for question answering with named entities as nodes and edges as relations between the nodes. Xu and Yang [19] build a coreference resolver by encoding text with a BERT model and forwarding it to a fully connected layer, which is later concatenated with another feature representation obtained from a GNN. In particular, they receive the other GNN-based representation by combining the BERT encoding also as a feature with a syntactic dependency graph. This is then the input to a relational graph convolutional network. Since we also combine a GNN with a BERT model, this approach is the most related to our first run. To the best of our knowledge, we are the first team using a GNN approach in the COLIEE competition task 4.

BERT-based approaches prove to be an effective solution for the past COLIEE edition. Moreover, domain-specific models may achieve better results than their standard variants. Graph neural networks have not been widely explored in the COLIEE competition.

Table 1: Methods for each run for task 4

Run	Method
OvGU_run1	Ensemble of Graph Neural Networks
OvGU_run2	LEGAL-BERT
OvGU_run3	LEGAL-BERT

Hence, we focused on the use of LEGAL-BERT and GNNs for the COLIEE 2021 challenge on the statute law entailment task.

3 STATUTE ENTAILMENT TASK

We develop two different approaches with three different runs, as described in Table 1. The first technique is an ensemble of GNNs, while the second and the third runs use LEGAL-BERT with different training approaches.

3.1 Ensemble of Graph Networks

Our graph consists of a set of nodes and edges, where each node represents either a query or an article. Edges are the connections between nodes. In the context of classification tasks, such a graph is often encoded by a neural network. This results in a graph embedding, which is then used as a feature, for example of a linear classification layer. Standard graph embedding approaches focus on the structure of the graph only. However, encoding external knowledge into such a graph as node features and using them while creating a graph embedding can be helpful, especially in cases of a rather small graph. This is particularly interesting for the statute entailment task with limited training data, and where the contextual meaning of queries and articles can be found in their content, but also within the relation between them. Graph approaches can model abstract relations which cannot be easily characterized, such as the entailment relationship. So we decide to use graphs in connection to contextual word embeddings to encode relationships between a query and positively or negatively entailed articles. In our implementation, we form a bipartite directed graph between the article and query nodes and try to learn the relation between them. Bipartite graphs can be divided into two subgraphs with each sub-graph having no connections within itself but connections with the nodes of the other subgraph. We choose this type of graph because we cannot directly model a relationship between the queries, however, the training data indicates a positive or negative entailment relationship between a query and its relevant articles. Since graphs can encode multiple relations to one node, we assume this is a good approach with the COLIEE dataset, where queries have multiple relevant articles and learning semantic relations within each of these query-article pairs can be beneficial for entailment classification.

Figure 1 shows a workflow of how the entailment is done with the graph neural network. Each node in the graph is represented by some features. We use extracted metadata from the section titles for each article to enrich its content before further processing with the language model. Table 2 gives an example of the metadata we used. With Sentence-BERT[1] [18], we generate sentence embeddings from the content of the queries and of the enriched articles and consider them as a feature for the corresponding node in the graph. We used the pre-trained *paraphrase-distilroberta-base-v1* model to create the sentence embeddings because it is claimed to work well on natural language inference tasks and was trained on millions of paraphrase pairs[2]. In previous experiments, we also employed LEGAL-BERT [2] in the Sentence-BERT architecture (accuracy on the validation set: 49.55%), but the performance was not comparable to the *paraphrase-distilroberta-base-v1* model, since LEGAL-BERT was not trained with the Siamese architecture of Sentence-BERT on a natural language inference task. The node embeddings are formed only for the query nodes and are based on the implementation by Morris et al. [13]. The resulting query node embedding also encodes information considering relevant articles as direct neighbor nodes. We then we use the node embeddings for the downstream task of query node classification for entailment. Graph neural networks employ a message passing technique, where the message from one node is passed to another node, embedding the neighborhood information of the node with an aggregation function. Here we use the average aggregation function for message passing. In the implementation by He et al. [6], they have used a large scale bipartite graph to get an efficient representation with intra- and inter-message aggregation. We use the inter-message aggregation method to just pass article information to the queries and not to other article nodes.

The below formula shows how we calculate embeddings for a query node which is in line with the work by Morris et al. [13], except for the difference that we concatenate the neighbour information instead of adding them, which has been done in the original implementation.

$$x_q = \sigma(concatenate(W_1 x_q, \frac{1}{|N|}\sum_{j=1}^{N} e_{q,a,j} W_2 x_{a,j})), \qquad (1)$$

where x_q represents query features as the node embedding, x_a represents relevant article features, N is the number of relevant articles, W represents weights and $e_{a,q,j}$ represents the edge weight between query and article. We give equal importance to each article, therefore setting the edge weight to 1. For σ, we selected the rectified linear unit as an activation function. We train two different models in a same way with different parts of the dataset and combine their result to get entailment. We train one model with the training dataset except for instances starting with the pair ID "R01-*" and another model with the full training dataset and finally take the average of their softmax values for determining the final prediction.

To summarize, we used a novel approach by employing sentence embeddings and node embeddings together to solve the entailment task. Also, we enriched the data with structural information of the Civil Code. In the following, we present the approach of using LEGAL-BERT for the two other runs.

3.2 LEGAL-BERT

For our runs 2 and 3, we consider a pre-trained model called *legal-bert-base-uncased*[3] [2] as our default choice for this task. More

[1] https://github.com/UKPLab/sentence-transformers

[2] https://www.sbert.net/docs/pre-trained_models.html#paraphrase-identification

[3] https://huggingface.co/nlpaueb/legal-bert-base-uncased

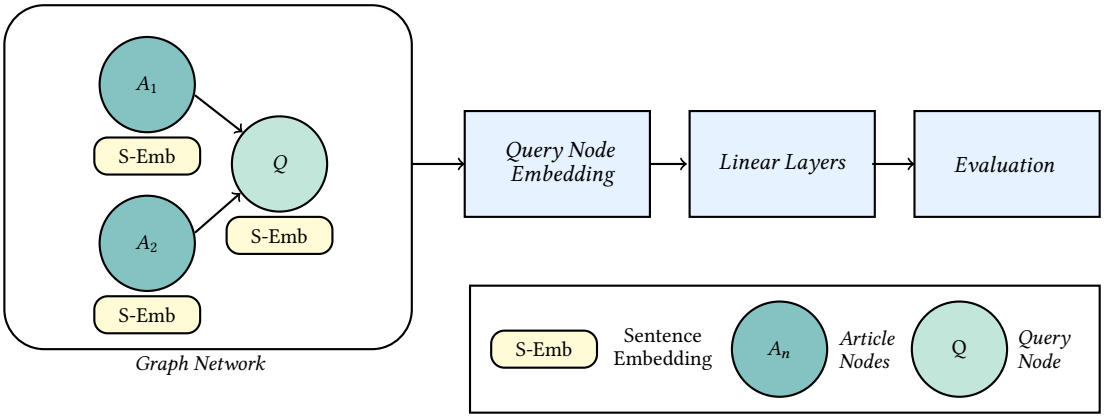

Figure 1: Workflow for the Ensemble of Graph Methods

Table 2: Example of metadata for Article 567 of the Civil Code

Training data
Article 567: (1) If the seller delivers the subject matter .. ground of the loss or damage. In such a case, the buyer may not refuse to pay the price. (2) The preceding paragraph also applies if the seller tenders .. tender of the performance due to any grounds not attributable to either party.
Metadata
Part: III Claims *Chapter:* II Contracts *Section:* 3 Sale *Sub-section:* (Transfer of Risk for Loss of Subject Matter)

details about how that pre-trained model is obtained have been shared in Section 2.1. We also participated in task 3 of COLIEE, the statute law retrieval task, and fine-tuned the aforementioned BERT model for that purpose. In that work, LEGAL-BERT outperformed the regular BERT model (*bert-base-uncased*), so we did not employ the regular model on task 4 anymore. The query-article pairs in the training data are the same for both tasks, but for task 4, we have additional entailment labels which are not required in task 3. However, since we performed task 3 also as a classification task, we used a different set of labels that described the relevance of an article to a query rather than the entailment labels. For task 3, except for the classification head, the encoder part of *legal-bert-base-uncased* is trained on the query-article pairs. In short, we transformed the originally imbalanced dataset in task 3 by decomposing all relevant articles into separate instances, in addition to keeping the original instance with potentially multiple relevant articles also in the training data. This increases the count of examples which contain relevant articles. Then, per query, we add the top 50 most similar, but not relevant articles to reduce the overall amount of irrelevant articles and thus include not obvious cases in the training phase for each query. We leveraged this trained encoder for task 4 and re-initialize the classification head with new trainable parameters.

In detail, the following steps were done to preprocess the training data:

(1) **Data decomposition:** The training data consists of query-article(s) pairs, where each query has one or more relevant articles. To extend the dataset for training, we split the training instances such that every instance has just one relevant article to which the given query is entailed. This increases the number of training instances and aids the model during training.
(2) **Data augmentation with non-relevant articles:** Further, for each instance in the new decomposed training dataset, we checked the cosine similarity of the relevant article in that instance with all the articles in the Civil Code, except for the ones which were already marked relevant by another instance of the same query. We used TF-IDF vectorization to compute the similarity and picked the top 50 articles. These are the potential non-relevant articles for the query but closely related to its relevant articles. This extension is motivated by the implementation by Nguyen et al. [14], however, they considered query-article similarity instead of the article-article similarity that we propose. We proposed the article-article similarity because we believe that articles are more closely associated with each other than they are with the queries, such that model can benefit from non-relevant examples that can have a similar terminology to the relevant ones.
(3) **Augmenting the dataset:** On top of this decomposed and augmented dataset, we also consider the training instances

in the training dataset in their original form. This also introduces duplicate instances but ensures that the original query-article(s) pair could still influence the model.

Note that both steps 2 and 3 for augmenting data are performed only for task 3, the statute retrieval task, from which we adopt the encoder model, and for task 4 we only perform that data decomposition step. Anyway, we describe all three steps as we adopted the task 3 encoder for task 4.

An example of our query-article instance in the training dataset is given in Table 3 for the query of pair id *H27-22-1*:

Query Q: *"In the case of a contract for sale of a specified thing, if the performance of the delivery has become impossible due to reason attributable to the seller, the effect of the contract of sale shall be lost by operation of law, then the buyer shall be relieved of liability for payment of the purchase money."*

We adopt two different training techniques to generate the run 2 and run 3 of our submission as depicted in Figure 2. The reason for this approach is that in previous experiments, we observed differences in model performance depending on the training and test split. We also experiment with different hyper-parameters, the results of this are discussed in the evaluation section. For both runs, we create a training and validation split (all queries starting with the ID "R01-*" for the validation set). For run 2, we fine-tune the above model on the training split and evaluate its performance on the validation set. Having achieved satisfying results, we further train the fine-tuned model on the validation split. However, for our run 3, we completely train the reinitialized classification head with the encoder base from task 3 on the entire training split including the validation split. Each run is trained using the same set of hyper-parameters that resulted in the best performance for the validation split.

To sum up, we re-used the LEGAL-BERT encoder we already had trained on the COLIEE statute law retrieval task and reinitialized its classification head. To make learning easier, we decomposed the instances with multiple articles for a query, forming additional instances. Run 2 was trained in 2 steps. The first step was to fine-tune it on our training split and then in the second step, we performed further fine-tuning on the validation split. Run 3 was fine-tuned on the full training data in one step. In the next section, we evaluate all three runs.

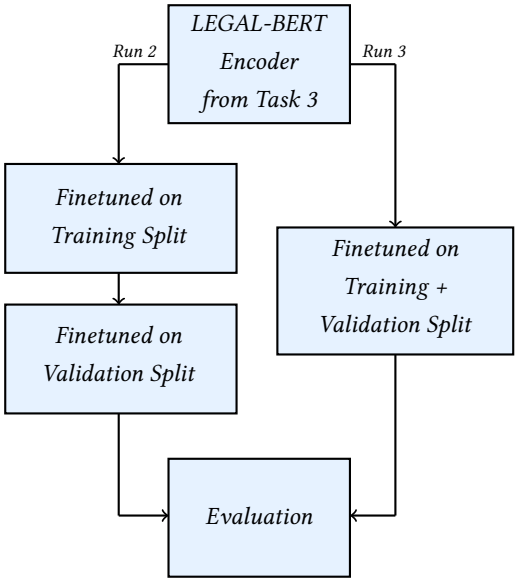

Figure 2: Fine-tuning workflow of the LEGAL-BERT encoder for run 2 and 3

4 EVALUATION

We start this section with details about the experimental setup, followed by results from the competition, previous experiments and a final discussion. For hyper-parameter optimization we evaluated our runs on the validation split of queries starting with "R01-*".

4.1 Experimental Setup

4.1.1 Graph Neural Networks. In our experiments for run 1, we first tried creating graphs by considering each word as a node and making connections between nodes (words) which are neighbors to that word in the sentence. We trained Word2Vec [9] vectors from the article descriptions in the Civil Code and the queries in the training dataset, using the *gensim*[4] library and used these as the features for each node. This way we tried creating graphs for both queries and articles and create a graph embedding for each graph, to combine their embeddings and use it for the entailment downstream task. But this approach did not seem to perform well on the validation split, with an accuracy of 56%.

To improve the results, we embed the sentences of the article and the query, respectively, using Sentence-BERT and use it as a node feature. We encode this information to get query node embeddings, as described earlier. These are then passed to the linear layer. The results have been found to outperform the above-mentioned approach of considering the Word2Vec representation for the article and query token nodes.

From experimenting with different data enrichment strategies, we found that the model tends to give better results when we add the metadata of the article while embedding it. This is shown in Table 4.

So for run 1, we use a graph neural network to create a query node embedding with 2 graph layers, then we add a linear layer with a rectified linear unit activation and a final linear layer with a softmax activation to perform the downstream task of entailment in PyTorch[5]. We evaluate our model on the validation split (all queries starting with the ID "R01-*"). After hyper-parameter optimization on the validation data, we train our model for 3 epochs as it was observed that the model was overfitting on the training set when the number of epochs was increased. We train our model with the Adam optimizer and with a batch size of 4 on the training split. We train the second model with the full training dataset including the validation split with the same hyper-parameters, so that the model makes use of all the data we have. To get the label predictions on the COLIEE 2021 test set, we take average softmax values from both models and choose the label with the higher confidence value.

[4]https://radimrehurek.com/gensim/models/word2vec.html

[5]https://pytorch.org/

Table 3: Data Decomposition to create additional instances using multiple relevant articles for each query

Queries	Articles	Label
Before Preprocessing		
Query Q	**Article 415** (1) If an obligor fails to perform ... (2) ... obligor's failure to perform the obligation **Article 542** (1) In the following cases, the obligee may ... (2) ... perform the part of the obligation **Article 543** If non-performance of an obligation is ... contract under the preceding two Articles..	Y
After Preprocessing		
Query Q	**Article 415** (1) If an obligor fails to perform ... (2) ... obligor's failure to perform the obligation	Y
Query Q	**Article 542** (1) In the following cases, the obligee may ... (2) ... perform the part of the obligation	Y
Query Q	**Article 543** If non-performance of an obligation is ... contract under the preceding two Articles..	Y

Table 4: Influence of data enrichment on COLIEE validation set

	Correct answers	Accuracy
without Metadata	66/111	0.5946
with Metadata	73/111	0.6577

4.1.2 LEGAL-BERT. For both run 2 and 3, we use the same experimental setting of hyper-parameters, albeit using different training techniques. We validate by decaying the learning rate using a decay rate of $(0.1^{(1+epoch)})$, varying the warmup steps from 5 - 20 % of the total training steps, and testing with other hyperparameters as shown in Table 5. However, warmup steps and decay rate did not have a significant impact on the performance of the LEGAL-BERT for task 4. Plausible reasons could be the small amount of training data and the fewer number of epochs used during training, resulting in a very small number of warmup steps, thereby, not providing the network enough time to adapt gradually. Also, the decay rate shrinks abruptly causing no significant effect on the learning process. For this reason, we left out the warmup steps and the decay rate for training our models. Finally, with hyper-parameter tuning, we observed that the learning rate of $5e^{-05}$ with a batch size of 8 trained for 4 epochs was the most suitable of all regarded options of our scenario.

4.2 Results

Among our submissions, OvGU_run3 achieved the fifth-highest accuracy, followed by OvGU_run2 with the sixth-highest accuracy obtaining 48 and 45 correct answers, respectively, given 81 test queries. Since both these runs are BERT-based approaches, we witness comparable results. Although none of our runs could achieve the highest performance, OvGU_run1 is a novel graph-based approach, which obtained 36 correct answers. The results are populated in Table 6.

Table 6: Task 4 Results for COLIEE 2021

Correct	Run	Accuracy
36	OvGU_run1	0.4444
45	OvGU_run2	0.5556
48	OvGU_run3	**0.5926**

4.3 Discussion

We can see that our method performs well when we consider the validation data in Table 7. Compared to this year's test set, we could observe that LEGAL-BERT was able to generalize much better than the GNN and was able to give 48 correct answers while the GNN performed worse. Graphs are usually used to represent large amounts of data, and since the dataset is limited, it may be assumed that the GNN was not able to generalize much with the dataset and the current hyperparameter settings and was not able to correctly predict the entailment relationship with the query node embeddings. We assume that graph-based techniques can achieve better scores in the future, since our approach for run 1 does not rely on much external knowledge, which could be encoded after performing more document enrichment.

Table 7: Evaluation result on validation data

Method	Correct answers	Accuracy
GNN	73/111	0.6577
LEGAL-BERT	73/111	0.6577

We examine all three runs with the ground truth provided by the COLIEE organizers. Results confirm that our run 3 predicts more correct answers than our other two runs. We analyze our runs to find a few interesting trends. Out of 81 queries, run 1 correctly predicted a total of 36 queries of which 11 queries were not correctly predicted by any of the other runs we submitted. Similarly, for run 3 out of our 48 correct predictions, 11 queries were not correctly predicted by the other runs we made. When it comes to run 2, we had a total of 45 correct predictions where 5 queries were not correctly predicted by our other runs. For run 2 and 3, most of their predictions would have an agreement with each other, precisely

Table 5: Optimizing hyper-parameters on the validation set for task 4 COLIEE 2021

Correct	epochs	batch-size	learning_rate	warm_up	decay	Accuracy
57	4	16	$1e^{-05}$	300	$0.1^{(1+epoch)}$	0.5135
61	5	16	$1e^{-05}$	400	$0.1^{(1+epoch)}$	0.5495
63	5	8	$1e^{-05}$	600	$0.1^{(1+epoch)}$	0.5676
68	5	8	$1e^{-05}$	-	-	0.6126
71	4	8	$1e^{-05}$	-	-	0.6396
73	4	8	$5e^{-05}$	-	-	**0.6577**

56 such queries, the reason might be since both of the runs are BERT-based approaches and even share the same initial encoder. Additionally, we observed that we could not predict 11 queries correctly in any of our runs while 16 were correctly predicted across all three.

We further notice that the LEGAL-BERT-based approach performs considerably better than the graph-based approach for the test queries associated with multiple articles, such as for pair IDs "R02-16-O", "R02-24-E" with an overall of 16 such queries. Our run 1, the graph-based model, is able to correctly predict 6 such queries while both BERT-based models of run 2 and 3 predicted 10 and 11 queries correctly. We believe that this is due to our input data decomposition for runs 2 and 3, recognizing each relevant article as a separate training instance. Furthermore, Sentence-BERT has a default token limit of 128, which means that all inputs with more tokens than that will be truncated. This has possibly affected the performance of run 1 for longer articles with the relevant entailment content at the end of the article part. For instance the encoding of Article 567 with 212 tokens did not change after we added the meta-data, hence for articles with this length, that measure had no effect. This drawback of using the default settings shall be considered in future work with Sentence-BERT.

Apart from our submitted runs, we evaluate another run that we did not submit in the competition. This model predicts 54 correct entailments for 81 test queries. This submission could have been the third-highest score. Its LEGAL-BERT model is only fine-tuned on the training split we had during validation, so that we do not use the queries starting with the ID "R01-*", as mentioned earlier. The model predicts 73 correct labels on the validation set, listed as the last entry in Table 5. We attribute this effect to the unstable training / test results on the COLIEE competition over time, such that the problem type distribution of queries starting with "R01-*" could be probably skewed.

In general, while reviewing the training and test dataset provided for COLIEE 2021 task 4, we find a few discrepancies.

(1) **Incomplete article description:** For multiple queries including *H22-23-E*, *H27-23-E*, *H27-23-O*, *R01-25-U* and *R01-25-O* where *Article 617* was marked one among the relevant articles, the description in the query-article pair was incomplete in the training dataset when compared with the article description in the Civil Code.
(2) **Mentioning only selected paragraphs of the article:** In the training dataset, *Article 718* was marked as relevant for for the query *H30-29-E*. Though the article had 2 different paragraphs in the Civil Code, only the 1st paragraph was provided in the query-article pair in the training dataset. This can be neglected owing to the count of 1 on a total of 806 train instances, however, when we consider the test dataset, there are 35 such instances on a total of 81. This is particularly interesting when different paragraphs of the same article were mentioned for different queries. For query *R02-27-A*, only the 2nd paragraph of *Article 676* is mentioned while for query *R02-27-O*, the 3rd paragraph is mentioned. This brings us to the question if the selected mentioning was intentional. In this case, task 4 could extend its scope to not just checking for an entailment label, given relevant articles and a query, but also to find the most relevant paragraph(s) in the given article and then check for an entailment label with the given query.
(3) **Mismatch in article number and article description:** For the test query *R02-2-E*, the description of the relevant article seems to be incorrect. When comparing with the Civil Code, the article description of *Article 36* was used in the test query-article pair, but the article number mentioned was *Article 35*. It is also interesting to note the original article description for *Article 35* in the Civil Code has Japanese characters.
(4) **Article description in Japanese:** The article description for the test query (in the English test dataset) *R02-5-I* is given in Japanese. This puts the competitors using the English dataset at a disadvantage of potentially losing one test instance and thereby decreasing the number of correct predictions.

From the discussions and findings above, our main takeaways from this year's COLIEE task 4 are:

(1) Data decomposition of queries associated with multiple articles into multiple instances can help the neural networks to model the query-article relation better.
(2) Developing an understanding of the data distribution or the skewness in individual training dataset subsets ("H28-*", "H29-*", "R01-*", and more) can help to address the queries better. It would be interesting to have a multi-label assignment of problem categories released for the training data.

5 CONCLUSION AND FUTURE WORK

To conclude, we test graph and contextual word embeddings for task 4 of textual entailment classification in the COLIEE competition. In particular, we use graph neural networks with sentence

Table 8: Discrepancies in the training and test data of COLIEE 2021 task 4

Query	Article description in the dataset	Article description in the Civil Code
R01-25-O *(train)*	Article 617 (1) If the parties do not specify the term of a lease, either party may give a notice of termination at any time. In such cases, a lease as set forth in one of the following items terminates when the term prescribed in that item has passed after the day of the notice of termination: (i) (ii) (iii) (2) With respect to leases of land with harvest seasons, the notice of termination must be given after the end of that season and before the next start of cultivation..	Article 617 (1) If the parties do not specify the term of a lease, either party may give a notice of termination at any time. In such cases, a lease as set forth in one of the following items terminates when the term prescribed in that item has passed after the day of the notice of termination: (i) leases of land: one year; (ii) leases of buildings: three months; and (iii) leases of movables and party room: one day. (2) With respect to leases of land with harvest seasons, the notice of termination must be given after the end of that season and before the next start of cultivation..
H30-29-E *(train)*	Article 718 (1) A possessor of an animal is liable to compensate for damage that the animal inflicts on another person;provided, however, that this does not apply if the possessor managed the animal while exercising reasonable care according to the kind and nature of the animal..	Article 718 (1) A possessor of an animal is liable to compensate for damage that the animal inflicts on another person; provided, however, that this does not apply if the possessor managed the animal while exercising reasonable care according to the kind and nature of the animal. (2) A person who manages an animal on behalf of a possessor also assumes the liability referred to in the preceding paragraph.
R02-5-I *(test)*	第百二十六条　取消権は、追認をすることができる時から五年間行使しないときは、時効によって消滅する。行為の時から二十年を経過したときも、同様とする。	
R02-2-E *(test)*	Article 35 (1) With the exception of states, administrative divisions of states, and commercial companies, the formation of foreign juridical persons is not permitted; provided, however, that this does not apply to a foreign juridical persons that is permitted pursuant to the provisions of a law or treaty. (2)A foreign juridical person permitted pursuant to the provisions of the preceding paragraph possesses the same private rights as those possessed by a juridical person of the same kind that has been formed in Japan; provided, however, that this does not apply to a right that a foreign national is not entitled to enjoy or to any right for which there are special provisions in a law or treaty.	Article 35 A person that is neither an incorporated association nor an incorporated foundation must not use in its name the characters "社団法人", "財団法人", or other characters likely to be mistaken for them.
R02-27-A *(test)*	Article 676 (2) A partner may not independently exercise the rights with regard to a claim that is included in the partnership property based on that partner's interest in the claim.	Article 676 (1) If a partner has disposed of the interest of the partner with respect to the partnership property, that partner may not duly assert that disposition against the partnership or third parties that had dealings with the partnership. (2) A partner may not independently exercise the rights with regard to a claim that is included in the partnership property based on that partner's interest in the claim. (3) A partner may not seek the division of the partnership property before liquidation.
R02-27-O *(test)*	Article 676 (3) A partner may not seek the division of the partnership property before liquidation.	Article 676 (1) If a partner has disposed of the interest of the partner with respect to the partnership property, that partner may not duly assert that disposition against the partnership or third parties that had dealings with the partnership. (2) A partner may not independently exercise the rights with regard to a claim that is included in the partnership property based on that partner's interest in the claim. (3) A partner may not seek the division of the partnership property before liquidation.

embeddings in run 1 and LEGAL-BERT variations in training, with two stages in run 2 and one stage for run 3. We find that increasing the number of instances with article decomposition can help to boost the performance of our approaches. From all submitted runs, the LEGAL-BERT run 3 which was fine-tuned on all available data in one stage performed best. However, we found that the training / test split can substantially impact the model performance, which is shown by run 2, 3 and an auxiliary run which outperformed all submitted runs. For future work, we intend to focus more on the generation of sentence embeddings and their appropriate aggregation for longer articles. Furthermore, we suggest to incorporate external knowledge via more extensive document enrichment with knowledge from the web into all three approaches as an addition to the decomposition and augmentation strategies we employed this time.

REFERENCES

[1] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 2306–2317. https://doi.org/10.18653/v1/n19-1240

[2] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. *CoRR* abs/2010.02559 (2020). arXiv:2010.02559 https://arxiv.org/abs/2010.02559

[3] Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-Supervised Sequence Modeling with Cross-View Training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 1914–1925. https://doi.org/10.18653/v1/d18-1217

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[5] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *CoRR* abs/2007.15779 (2020). arXiv:2007.15779 https://arxiv.org/abs/2007.15779

[6] Chaoyang He, Tian Xie, Yu Rong, Wenbing Huang, Junzhou Huang, Xiang Ren, and Cyrus Shahabi. 2020. Cascade-BGNN: Toward Efficient Self-supervised Representation Learning on Large-scale Bipartite Graphs. arXiv:1906.11994 [cs.SI]

[7] Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science* 349, 6245 (2015), 261–266.

[8] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 328–339. https://www.aclweb.org/anthology/P18-1031/

[9] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 957–966. http://proceedings.mlr.press/v37/kusnerb15.html

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1eA7AEtvS

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[12] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. In *Advances in Neural*

Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 6294–6305. https://proceedings.neurips.cc/paper/2017/hash/20c86a628232a67e7bd46f76fba7ce12-Abstract.html

[13] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 4602–4609. https://doi.org/10.1609/aaai.v33i01.33014602

[14] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Tran Binh Dang, Quan Minh Bui, Vu Trong Sinh, Chau Minh Nguyen, Vu D. Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *CoRR* abs/2011.08071 (2020). arXiv:2011.08071 https://arxiv.org/abs/2011.08071

[15] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/n18-1202

[16] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. https://sites.ualberta.ca/~rabelo/COLIEE2021/COLIEE2020_summary.pdf. Accessed: 2021-05-09.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf, note= "Accessed: 2021-05-09".

[18] Nils Reimers and Iryna Gurevych. 2019. 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 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[19] Yinchuan Xu and Junlin Yang. 2019. Look Again at the Syntax: Relational Graph Convolutional Network for Gendered Ambiguous Pronoun Resolution. *CoRR* abs/1905.08868 (2019). arXiv:1905.08868 http://arxiv.org/abs/1905.08868

[20] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 7370–7377. https://doi.org/10.1609/aaai.v33i01.33017370

BERT-based Ensemble Methods for Information Retrieval and Legal Textual Entailment in COLIEE Statute Law Task

Masaharu Yoshioka
yoshioka@ist.hokudai.ac.jp
Faculty of Information Science and Technology, Hokkaido University
Graduate School of Information Science and Technology, Hokkaido University
Sapporo-shi, Hokkaido, Japan

Youta Suzuki*
Yasuhiro Aoki*
suzuki@eis.hokudai.ac.jp
yasu-a_01@eis.hokudai.ac.jp
Graduate School of Information Science and Technology, Hokkaido University
Sapporo-shi, Hokkaido, Japan

ABSTRACT

We have participated in three tasks (information retrieval, legal textual entailment, and combination of both systems) of Competition on Legal Information Extraction/Entailment (COLIEE) statute law tasks. We used the BERT-based system as a core module for calculating similarity and recognizing entailment. For task 3, we used BERT-based IR systems with different settings and the ordinal keyword-based IR system for making ensemble results. For task 4, we also propose a BERT-based ensemble legal textual entailment with data augmentation. Task 5 utilizes the output of task 3's results for the articles for entailment using the system developed for task 4. We discuss the characteristics of the system using evaluation results for COLIEE 2021 submissions.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; *Ensemble methods*; • **Information systems** → *Structured text search.*

KEYWORDS

Information Retrieval, Textual entailment, BERT, Ensemble method

ACM Reference Format:
Masaharu Yoshioka, Youta Suzuki, and Yasuhiro Aoki. 2021. BERT-based Ensemble Methods for Information Retrieval and Legal Textual Entailment in COLIEE Statute Law Task. In *Proceedings of COLIEE 2021 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2021).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

The Competition on Legal Information Extraction/Entailment (COLIEE) [2–5, 10] serves as a forum to discuss issues related to legal information retrieval (IR) and entailment. There are two types of tasks in COLIEE. One is a task using case law (tasks 1 and 2), and the other is a task using Japanese statute law using Japanese bar exam questions (tasks 3, 4, and 5).

COLIEE 2021, June 21, 2021, Online

HUKB participated in three tasks of Japanese statute law as follows. Task 3: information retrieval task, Task 4: textual entailment task, and Task 5: answering bar exam questions by retrieving relevant article(s) with textual entailment.

In this paper, we introduce our approach to tasks 3 and 5. For task 3, we propose a system based on that used for HUKB at COLIEE 2020. This system utilizes BERT [1] for handling semantic matching between the questions and articles by making ensemble results generated by BERT-based IR systems [6] and the ordinal keyword-based IR system Indri[8]. This year, we propose a method to construct a new article database by rewriting the articles using references among those articles and splitting them into fine-grained sentences. For task 5, we use the output of task 3 as an input of the BERT-based ensemble legal textual entailment introduced in [9]. We also discuss the characteristics of the system using the evaluation results for COLIEE 2021 submissions.

2 TASK3: INFORMATION RETRIEVAL TASK

Our system is based on the system introduced in COLIEE 2020 [11]. This system ensemble the results of the BERT-based IR system [6] [1] and Indri [8][2], which is a state-of-the-art ordinal keyword-based IR system.

The new proposal improves the performance of our system in the following ways.

(1) Reconstruction of article database
  (a) New article database by aggregating the description of reference article information
  Many articles refer to other articles. To understand the meaning of such articles, it is necessary to describe the refereed articles. Consequently, we propose a method to reconstruct an article database by making aggregated articles that combine descriptions in the original article and refereed article.
  (b) Splitting long articles with multiple conditions and judicial decision into small sentences
  Many articles discuss lists of conditions for deciding one judicial decision. Many articles also discuss multiple judicial decisions mostly separated by numbered items. We propose a method to split these decisions into pieces of short sentences that correspond to the original article. This method will be effective by removing unrelated parts from

[1] https://github.com/ku-nlp/bert-based-faqir
[2] https://www.lemurproject.org/indri/

the target article and increasing the similarity score for matching cases.

(2) New criteria for making BERT models for ensemble.
In COLIEE 2020, the best performance systems for task 3 [7] use BERT-based models. One of the significant differences between the system and our system used for COLIEE 2020 is how training data are made and balanced. Their method uses an oversampling approach for the positive cases (relevant articles) by a factor of 100 (using the same positive cases 100 times), while our method uses negative sampling that randomly selects small numbers of negative cases. In this year, we use both approaches to make BERT models and afterward ensemble the results.

2.1 Construction of New Article Database

New Article Data using Reference Information

There are many articles that refer different article, and it becomes difficult to estimate the meaning of the article using the word or word sequence. The article that explains how to apply "Mutatis-mutandis" is a type of the example for such problem. Since, such articles may not have clear judicial decision parts, it is impossible to understand the meaning of the article by the explanation. As a result, when this type of article is selected as a relevant article, (a) refereed article(s) is (are) also selected as a relevant article to judge whether a set of articles entails the given question or not. Therefore, we manually construct article texts that combines this type of article and (a) refereed article(s).

New Article Data Using Reference Information

Many articles refer to different articles, and it becomes difficult to estimate the meaning of the article using the word or word sequence. The article that explains how to apply mutatis mutandis is an example of such a problem. Because such articles may not have clear judicial decision parts, it is impossible to understand the meaning of the article by the explanation. As a result, when this type of article is selected as a relevant article, (a) refereed article(s) is (are) also deemed relevant to judge whether a set of articles entails the given question or not. Therefore, we manually construct article texts that combine this type of article and (a) refereed article(s).

Figure 1 is an example for making a combined article from the original article and refereed article (520-13). We replace words in the refereed article using words in the original article (質権の設定: Creation of a pledge). It is not difficult for humans to interpret these articles, but it is not easy to make such data automatically. Consequently, we manually construct these aggregated texts.

Because article 520-17 refers to four articles (520-13 to 520-16), we made four article data (520-17+520-13, 520-17+520-14, 520-17+520-15, 520-17+520-16) that are registered instead of 520-17. We also apply the same methods for other types of refereed articles.

After making the combined article list, we check the necessity of using such (a) refereed article(s) by comparing the training data. For example, when article X refers to Y, there is a case where the training data contains a question and article pair that uses article X without Y for entailment. This may happen when the refereed parts are not directly related to the decision, i.e., when the article has multiple conditions and/or multiple judicial decisions.

Original articles: Article 520-17 that refers Article 520-13

第五百二十条の十七　第五百二十条の十三から前条までの規定は、記名式所持人払証券を目的とする質権の設定について準用する。

Article 520-17 The provisions of Article 520-13 through the preceding Article apply mutatis mutandis to the creation of a pledge on a registered negotiable instrument payable to holder.

第五百二十条の十三　記名式所持人払証券（債権者を指名する記載がされている証券であって、その所持人に弁済をすべき旨が付記されているものをいう。以下同じ。）の譲渡は、その証券を交付しなければ、その効力を生じない。

Article 520-13 Assignment of a registered negotiable instrument payable to holder (meaning a negotiable instrument on which the name of the obligee is written with a supplementary note that payment should be made to its holder; the same applies hereinafter) does not become effective unless the instrument is delivered to the assignee.

Combined article(520-17+520-13)

記名式所持人払証券（債権者を指名する記載がされている証券であって、その所持人に弁済をすべき旨が付記されているものをいう。以下同じ。）の質権の設定は、その証券を交付しなければ、その効力を生じない。

Creation of a pledge of a registered negotiable instrument payable to holder (meaning a negotiable instrument on which the name of the obligee is written with a supplementary note that payment should be made to its holder; the same applies hereinafter) does not become effective unless the instrument is delivered to the assignee.

Figure 1: Example of making aggregated texts with a refereed article

For such cases, it is not appropriate to make such aggregated text for the article database, and we exclude such combined articles and use original texts for the article.

As a result, we construct 211 articles (there are 190, 19, and 2 combined articles with 2, 3, and 4 article combinations, respectively) from 123 articles, yielding a total number of articles of 856.

In this database, when article 520-17+520-13 is relevant, we select articles 520-17 and 520-13 as relevant. We also check the results for duplicates for when the retrieved results contain multiple entries for those articles. For example, when article 520-17+520-14 is selected as a second-rank result, we only select 520-14 as a result. When article 520-13 is selected, we skip this result because 520-13 is already included as a relevant result using 520-17+520-13.

New Article Data by Splitting the Text

Many articles discuss lists of conditions for deciding one judicial decision. Those articles have longer sentences that cover a wider vocabulary. Because of those characteristics, they have a higher possibility to share multiple words from the difficult question that

Original articles: Article 169

第百六十九条　確定判決又は確定判決と同一の効力を有するものによって確定した権利については、十年より短い時効期間の定めがあるものであっても、その時効期間は、十年とする。２　前項の規定は、確定の時に弁済期の到来していない債権については、適用しない。

Article 169 (1) The period of prescription of a right determined by a final and binding judgment or anything that has the same effect as a final and binding judgment is 10 years even if a period of prescription shorter than 10 years is provided for. (2) The provisions of the preceding paragraph do not apply to a claim which is not yet due and payable at the time when it is determined.

Split articles by paragraphs

169(1) 確定判決又は確定判決と同一の効力を有するものによって確定した権利については、十年より短い時効期間の定めがあるものであっても、その時効期間は、十年とする。

169(1) The period of prescription of a right determined by a final and binding judgment or anything that has the same effect as a final and binding judgment is 10 years even if a period of prescription shorter than 10 years is provided for.

169(2) 確定の時に弁済期の到来していない債権については、十年より短い時効期間の定めがあるものであっても、その時効期間に従う。

(2) The period of prescription of a right which is not yet due and payable at the time when it is determined is effective even if a period of prescription shorter than 10 years is provided for. provided for.

Figure 2: Example of making split texts from longer articles

shares small numbers of words for other articles. In many cases, it does not justify longer articles being relevant to such a question. In contrast, the existence of words that are not directly related to the judicial decision of the question decreases the similarity score for such articles.

We also have other problems related to the settings of the BERT. To handle longer article sentences, it is necessary to have a BERT model that can handle longer sequences. However, because of the cost of training a BERT fine-tune model, it is difficult to set the longer sequence that covers the whole contents of the articles. It is also better to use shorter sentences for calculating the similarity.

Based on this understanding, we also made an article text database by splitting whole article texts into smaller ones. Because most of such longer articles have paragraph structures with numbers, we split the text using these numbers.

Figure 2 shows an example of making split texts. For the first paragraph, it is not necessary to modify the contents. However, from the second paragraph, some cases are difficult to interpret without including the information of the preceding paragraph. For such a case, we manually rewrite the sentence to represent appropriate meanings (169 (2) in Figure 2).

By using this splitting process, we construct 1336 articles from 856 articles' data constructed by adding references. However, when there are no references to the other articles in the paragraph, the combined article number is only added for the paragraph that has a reference. For example, article 398-22 has reference to article 398-16 in paragraph 2 and mutatis mutandis reference to articles 380 and 381 in paragraph 3. In such a case, the former method makes combined articles 398-22+398-16+380 and 398-22+398-16+381, but this method makes a set of articles, 398-22(1), 398-22(2)+398-16, 398-22(3)+380, and 398-22(3)+381.

Because there are cases where it is better to compare the results of whole articles, we construct a database that uses all the reference expanded articles and split articles (2192(1336+856) articles) and exclude redundant results from the retrieval results. For example, when article 398-22(2)+398-16 is relevant, we select articles 398-22 and 398-16 as relevant. We also check the results for duplicates when the retrieved results contain multiple entries for those articles. For example, when article 398-22(3)+380 is selected as a second-rank result, we only select 380 for an additional relevant result. When article 398-22(1) is selected, we skip this result because 398-22 is already included in relevant results using 398-22(2)+398-16.

2.2 Implementation of a BERT-based Ensemble System

Our system is based on the system used for the COLIEE 2020 [11] submission. The difference between the system for COLIEE 2021 and the system for COLIEE 2020 is a method for making the document database discussed in Section 2.1 and one for making the training data for the BERT-based IR system.

In this year, we train the BERT-based IR system using training data constructed by two different methods. One is a negative sampling approach that uses 24 negative examples by selecting from non-relevant articles. The other is an oversampling approach used in the best performance system of COLIEE 2020 IR task[7]. This approach oversamples the positive cases (relevant articles) by factor 100 (using the same positive cases 100 times) to reduce the effect of imbalance between the number of positive examples and the number of negative ones. For the training data, because there is no information about which separated paragraph corresponds to the relevant part of the article, we use new article data sets that use reference information for the training data. In this training data, when two articles correspond to one combined article, the combined article is used (e.g., when the relevant articles are 520-17 and 520-13, we use one article 520-17+520-13 for the relevant article).

Three system outputs are aggregated by an ensemble approach introduced in COLIEE 2020 [11]. This approach compares the retrieval result of all submissions, and when the top-ranked retrieval results of all submissions are the same, these results are treated as easy. Because we extend this framework to aggregate two or more system outputs, we modified the process as follows.

(1) **Aggregation of the results**
Because the scoring format used by the two systems is different (Indri generates negative scores, and the BERT-based IR system generates scores from 0 to 1), it is inappropriate to simply aggregate these scores. Therefore, we used the following formula to calculate the score using the rank of

each article for the question of System$_i$ ($Rank_{i,article}$) where $i = 0, \cdots, n-1$ ($n \geq 1$ is the number of systems for aggregation).

$$score_{article} = \sigma_0^{n-1} \frac{1}{(Rank_{i,article} + \alpha) * n} \quad (1)$$

Here, α is a parameter to reduce the score of the first-rank articles compared with the second-rank articles. In addition, because BERT's task is binary classification, a difference in score of less than β is considered meaningless. Therefore, we set the rank as 800 instead of the original rank whenever the score was below β. We used $\alpha = 1$ and $\beta = 0.3$ in this experiment.

(2) **Estimation of the retrieval difficulty**
If all systems return the same article as the first-ranked relevant article, we classify the question as "easy" and return the first-ranked article in the overall result. If the results are different, we classify the question as "difficult" and return the top three ranked documents in the overall result. However, when a combined article (e.g., 520-17+520-13 introduced in Section 2.1) is a candidate for the returned results, all original articles (except redundant ones) are included in the return results. Therefore, when the top-ranked result of all systems is 520-17+520-13, the system returns 520-17 and 520-13 as candidate documents. In addition, the system may return more than three articles when the third-ranked article is a combined article.

2.3 Submitted Results

This year, we submit three results that use different ensemble settings. **HUKB-1** uses three models (BERT with oversamples, BERT with negative sampling, and Indri). **HUKB-2** uses BERT with oversamples only (no ensemble). **HUKB-3** uses two models (BERT with oversamples and Indri). For the BERT-based IR system, we use a split text database (1336 articles) as a document database, but because of the mistake of handling data, Indri used the results of the non-split database (856 articles) for the submission.

Table 4 shows evaluation results of the submitted runs. Because the performance of the BERT-based IR system (HUKB-2) is significantly worse, the overall performance of our submission is low compared with the best (OVGU_run1) and second-best systems (JNLP.CrossLMultiLThreshold). Insufficient performance of the BERT-based IR system may come from the inappropriate settings of the BERT-IR system that uses longer sentences for training by cutting the latter part of sentences because of the maximum sequence length of the BERT model. There are several cases where the most important descriptions related to judicial decisions go outside of the max sequence because of the expansion of the article. This problem happens because of the insufficient time to find appropriate settings for the BERT model. Further investigation is necessary for the insufficient performance of the BERT system.

The number of questions classified by returned articles for HUKB-1, HUKB-2, and HUKB-3 are shown in Table 2. Questions that return two articles are cases where the system selects a combined article as the top-ranked candidate. There are few questions where BERT and Indri return the same article as the top-ranked one (e.g., 6 for BERT with oversamples and Indri: HUKB-3). As a result, ensemble-based systems (HUKB-1 and HUKB-3) return more than three articles for most of the questions, degrading the precision of the returned results.

Table 1: Evaluation Results of all submitted runs

Submission ID	F2	Prec	Recall	MAP
OvGU_run1	0.7302	0.6749	0.7778	0.7496
JNLP. CrossLMultiLThreshold	0.7227	0.6000	0.8025	0.7947
HUKB-3	0.5224	0.2901	0.6975	0.6100
HUKB-1	0.4732	0.2397	0.6543	0.6128
HUKB-2	0.3258	0.3272	0.3272	0.4167

Table 2: Number of questions classified by the number of return articles

Submission ID	1	2	3	4	5
HUKB-1	2	0	67	12	
HUKB-2	73	8	0	0	
HUKB-3	6	1	64	9	1

Among them, R02-36-I is the only question that contains a relevant article (included in both HUKB-2 and HUKB-3), but the output using combined articles is not appropriate.

Because of the existence of the reference, we made combined articles with article 152 (refereed in paragraph 3). However, because this part of the article is not related to the question, it is not necessary to include this article. Instead, it is better to redefine the process to rewrite such articles with references that may be related to explain the exceptional cases.

3 TASK5: COMBINATION OF IR AND TEXTUAL ENTAILMENT TASK

We have also participated in the textual entailment task (task 4). Our method is based on the BERT-based ensemble system with data augmentation [9]. This system uses a systematic method to make the training data for understanding the syntactic structure of the questions and articles for entailment. In addition, because of the nature of the non-deterministic characteristics of the BERT fine-tuning process and the variability of the question, we propose a method to make multiple BERT fine-tuning models and select an appropriate set of ensemble models. The accuracy of our proposed method for task 4 is 0.7037, and it is the best performance among all submissions.

Because it takes time to make the system for task 4, our system cannot submit results for task 5 before the deadline, but we also made results that combine the output of task 3 and task 4.

In task 4, we submit results with three different ensemble model configurations whose accuracies are HUKB-1: 0.6790, HUKB-2: 0.7037, and HUKB-3: 0.6790. Because participants for task 5 cannot know such values before submitting the data, we generate three configurations without considering the performance of each system. Table 3 shows a configuration of three runs that combine the results of these two tasks.

Question

時効の完成猶予の効力は，その事由が生じた当事者の承継人に対しては生じない。

The postponement of expiry of prescription period is not effective against the successors of the parties with respect to whom grounds to postpone the expiry of prescription period have arisen.

Relevant article

第百五十三条　第百四十七条又は第百四十八条の規定による時効の完成猶予又は更新は、完成猶予又は更新の事由が生じた当事者及びその承継人の間においてのみ、その効力を有する。

２　第百四十九条から第百五十一条までの規定による時効の完成猶予は、完成猶予の事由が生じた当事者及びその承継人の間においてのみ、その効力を有する。

３　前条の規定による時効の更新は、更新の事由が生じた当事者及びその承継人の間においてのみ、その効力を有する。

Article 153 (1) The postponement of the expiry of prescription period or the renewal of prescription period under the provisions of Article 147 or Article 148 is effective only between the parties with respect to whom grounds to postpone the expiry of prescription period or to renew prescription period have arisen and their successors.

(2) The postponement of expiry of prescription period under the provisions of Articles 149 through 151 is effective only between the parties with respect to whom grounds to postpone the expiry of prescription period have arisen and their successors.

(3) The renewal of prescription period under the provisions of the preceding Article is effective only between the parties with respect to whom grounds to renew prescription period have arisen and their successors.

Figure 3: Example of the failure for the usage of combined article

Table 3: Configuration for making task 5 results

Submission ID	Task 3	Task 4
HUKB-1	HUKB-1	HUKB-1
HUKB-2	HUKB-2	HUKB-1
HUKB-3	HUKB-1	HUKB-2

Because these results were generated after the deadline, it is not appropriate to compare the results with others, but the accuracy of our system on HUKB-3 that uses the best performance system of Task 4 is almost comparable to the best performance system (JNLP.NFSP) of task 5.

When there are no relevant articles in the retrieved results, it is not meaningful to discuss the quality of the answer. Therefore, we classify the question using recall.

Table 4: Evaluation Results of all the submitted runs

Submission ID	Correct	Accuracy
Baseline	No43/All81	0.5309
JNLP.NFSP	49	0.6049
HUKB-3	48	0.5926
UA_parser	46	0.5679
JNLP.NMSP	45	0.5556
HUKB-1	45	0.5556
HUKB-2	45	0.5556
UA_dl	45	0.5556
TRDistillRoberta	44	0.5432

Table 5: Number of questions classified by recall

Submission ID	Recall		
	1	0.5	0
HUKB-1	50	6	25
HUKB-2	26	1	54

Table 6 shows the accuracy of each run classified by the existence of all relevant article(s) (recall = 1) in the target articles.

Table 6: Number of questions classified by recall

Submission ID	With All Relevant		Without relevant	
	Correct	All	Correct	All
HUKB-1	32 (0.64)	50	13 (0.4194)	31
HUKB-2	19 (0.7308)	26	26 (0.4727)	55
HUKB-3	29 (0.58)	50	19 (0.6129)	31

Because of the size limitation of BERT, there are cases where sentences of relevant articles are not used for the entailment. Therefore, we also calculate the performance of the question whose mean average precision (MAP)= 1 (meaning (a) relevant article(s) is (are) ranked at the top) (Table 7).

Table 7: Number of questions classified by MAP

Submission ID	MAP=1.0		MAP < 1.0	
	Correct	All	Correct	All
HUKB-1	25 (0.6757)	37	20 (0.4545)	44
HUKB-2	19 (0.7308)	26	26 (0.4727)	55
HUKB-3	24 (0.6486)	37	24 (0.5455)	44

From Table 6, we confirm the accuracy of the results with relevant articles are better than accuracy without relevant articles except HUKB-3. However, when we restrict the questions where relevant articles are top ranked (Table 7), we confirm the accuracy of the results with relevant articles are better than accuracy without relevant articles. This justifies our task 4 system can utilize the relevant article information well to obtain the better results. However, it is difficult to compare the performance of HUKB-1 and HUKB-3, because HUKB-3 has more coincidental correct answers (0.5455 24/44) for questions without relevant articles than HUKB-1 (0.4545:20/44).

4 SUMMARY

In this paper, we introduced our system for participating in tasks 3 (IR task) and 5 (combination of IR and legal textual entailment) of COLIEE 2021. For task 3, because of the insufficient performance of the BERT-based system, the overall performance of the system is not effective. It is, thus, necessary to investigate the problem of BERT-based IR system. For task 5, the overall performance is comparable to the best performance system because of the best performance system for legal textual entailment task (task 4). However, because there are many coincidental correct answers without using relevant articles, it is difficult to determine which submission is the best system in total. For future works, it is also important to discuss the system performance characteristics of the system related to the difficulty of the IR task.

ACKNOWLEDGMENT

We thank the organizers of the COLIEE for their efforts in constructing this test data. This work was partially supported by JSPS KAKENHI Grant Number 18H0333808.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[2] Yoshinobu Kano, Mi-Young Kim, Randy Goebel, and Ken Satoh. 2017. Overview of COLIEE 2017. In *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment (EPiC Series in Computing, Vol. 47)*, Ken Satoh, Mi-Young Kim, Yoshinobu Kano, Randy Goebel, and Tiago Oliveira (Eds.). EasyChair, 1–8.

[3] Mi-Young Kim, Randy Goebel, Yoshinobu Kano, and Ken Satoh. 2016. COLIEE-2016: Evaluation of the Competition on Legal Information Extraction and Entailment. In *The Proceedings of the 10th International Workshop on Juris-Informatics (JURISIN2016)*. Paper 11.

[4] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE2020:Methods for Legal Document Retrieval and Entailment. In *The Proceedings of the 14th International Workshop on Juris-Informatics (JURISIN2020)*. The Japanese Society of Artificial Intelligence„ 114–127.

[5] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. A Summary of the COLIEE 2019 Competition. In *New Frontiers in Artificial Intelligence*, Maki Sakamoto, Naoaki Okazaki, Koji Mineshima, and Ken Satoh (Eds.). Springer International Publishing, Cham, 34–49.

[6] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval Using Query-Question Similarity and BERT-Based Query-Answer Relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1113–1116. https://doi.org/10.1145/3331184.3331326

[7] Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. 2020. BERT-based Ensemble Model for The Statute Law Retrieval and Legal Information Entailment. In *The Proceedings of the 14th International Workshop on Juris-Informatics (JURISIN2020)*. The Japanese Society of Artificial Intelligence„ 223–234.

[8] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*. 2–6.

[9] Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. 2021. BERT-based Ensemble Methods with Data Augmentation for Legal Textual Entailment in COLIEE Statute Law Task. In *The Proceedings of International Conference on Artificial Intelligence and Law 2021 (ICAIL2021)*. (to appear).

[10] Masaharu Yoshioka, Yoshinobu Kano, Naoki Kiyota, and Ken Satoh. 2018. Overview of Japanese Statute Law Retrieval and Entailment Task at COLIEE-2018. In *The Proceedings of the 12th International Workshop on Juris-Informatics (JURISIN2018)*. The Japanese Society of Artificial Intelligence„ 117–128.

[11] Masaharu Yoshioka and Youta Suzuki. 2020. HUKB at COLIEE 2020 Information Retrieval Task. In *The Proceedings of the 14th International Workshop on Juris-Informatics (JURISIN2020)*. The Japanese Society of Artificial Intelligence„ 195–208.

Author Index