COLIEE 2020: Methods for Legal Document Retrieval and Entailment

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

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

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

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

1 Introduction

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

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

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

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

2 Task 1 - Case Law Information Retrieval

2.1 Task Definition

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

2.2 Dataset

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

2.3 Approaches

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

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

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

2.4 Results

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

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

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

3 Task 2 - Case Law Entailment

3.1 Task Definition

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

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

3.2 Dataset

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

3.3 Approaches

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

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

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

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

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

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

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

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

3.4 Results

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

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

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

4 Task 3 - Statute Law Retrieval

4.1 Task Definition

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

4.2 Dataset

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

exam are used (112 questions). The number of questions classified by the number of relevant articles is as follows (1 answer: 87, 2 answers 22, 3 answers 3) [9].

4.3 Approaches

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

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

4.4 Results

Table 3 shows the evaluation results of submitted runs (due to page limit constraints, only the best run in terms of F2 is selected from each team). The official

[9] There is one question (R1-23-1: relevant articles are 554 and 1002) that have a relevant article excluded by this competition (1002). We also calculated the results by excluding this question, but there is no significant difference with official evaluation results. So we use the official evaluation results for this paper.

evaluation measures used in this task were macro average of precision, recall and F2 measure. We also calculate the mean average precision (MAP), recall at k (R_k: recall calculated by using the top k ranked documents as returned documents) by using the long ranking list (100 articles).

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

Table 3. Evaluation results (Task 3)

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

Figure 1.2 shows average of evaluation measure for all submission runs. As we can see from left part of Figure 1, there are many easy questions that almost all system can retrieve the relevant article. Easiest question is R01-12-U whose relevant article is almost same as a question. However, there are also many queries for which none of the systems can retrieve the relevant articles (Figure 1 right). R1-14-U[10] is an example of this question. The relevant article is Article 87 [11]. It is necessary to interpret the relationship between the "building and leased land" in the question as "first thing attaches a second thing" in the article. Even though BERT is good at ranking articles that take into account the context, it is difficult to estimate such interpretation that requires legal knowledge to interpret the context.

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

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

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

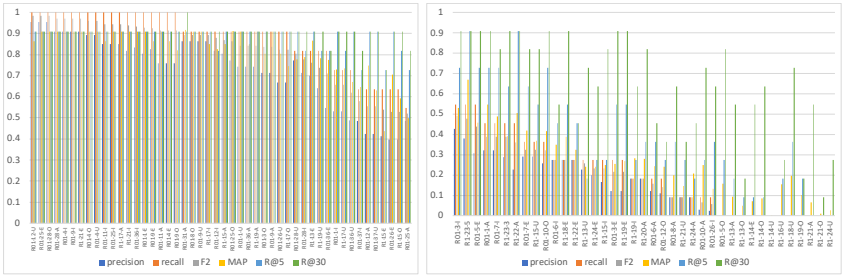

Fig. 1. Avg. of prec., rec., F2, MAP, R_5 and R_30 (questions with 1 relevant article)

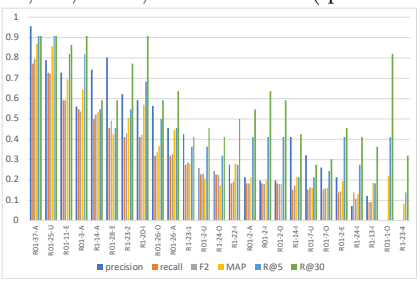

Fig. 2. Avg. of prec., rec., F2, MAP, R_5 and R_30 (questions with 1+ relevant article)

5 Task 4 - Statute Law Entailment

5.1 Task Definition

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

5.2 Dataset

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

5.3 Approaches

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

- **CU (two runs)** [2] uses multilingual cased BERT model for sequence classification trained and evaluated only with given relevant articles (CUGIVEN), plus additional articles returned using TFIDF (CUPLUS).
- **cyber (one runs)** [18] uses RoBERTa based method which is fine-tuned on sentence pair classification with the SNLI corpus. The resulting model fine-tuned on text pair classification with COLIEE training data.
- **GK_NLP (one run)** uses similarity measure on BERT embeddings and GloVe word embeddings with lightgbm classification model.
- **HONto (three runs)** [17] uses linear kernel SVM with TF-IDF and n-grams.
- **JNLP (three runs)** [10] uses BERT; JNLP.BERT and JNLP.TfidfBERT with the Google's original BERT_Base, JNLP.BERTLaw pretrained by American cases of 8.2M sentences/182M words in English. JNLP.BERTLaw and JNLP.BERT were fine-tuned by lawfulness classification on Augmented JAPAN Civil Code + COLIEE training data; JNLP.TfidfBERT was fine-tuned by COLIEE training data, no cross-fold validation.
- **KIS (three runs)** [4] built a range of Japanese legal dictionaries for predicate argument structures and paraphrases, which can integrate PROLEG, an legal logic language. KIS is their rule-based ensemble NLP system; KIS_2 uses SVM instead of rules in KIS; KIS_3 uses PROLEG to answer some of the questions in KIS_2.
- **LLNTU (one run)** [13] combines each query from COLIEE training dataset with all civil law articles, trains BERT-based ensemble models.
- **OvGU (three runs)** [17] uses Bidirectional LSTM and a modified Bahdanau's attention with inputing Law2Vec embeddings (baseline_attention. task4.OvGU), with the similarity measure and negations (sim_neg.task4. OvGU), adding POS of each token (POS_simneg.task4.OvGU).
- **tax-i (two runs)** [1] uses legal embeddings (FastText trained on US Caselaw) as input to a Bi-directional GRU with 128 Hidden Layers and 1 GRU layer (LEBIGRU), last hidden state of BERT base-cased was used as input to an XGBoost classifier (BERTXGB).
- **TRC3 (three runs)** [5] uses GloVe word embedding. Multee (TRC3mt) was trained phase one against single sentence NLI datasets (SNLI, MutliNLI) and then trained phase two on multiple sentence NLI datasets (OpenbookQA, COLIEE). The Text-To-Text Transfer Transformer (T5) run (TRC3t5) was fine-tuned on three denoising tasks (Civil Code Article, Civil Code Titles, Translated Wikibook Articles) and one entailment task (COLIEE).
- **UA (three runs)** [12] uses a decomposable attention model, which is a simple neural architecture for natural language inference, decomposing a problem into sub-problems (UA_attention_final), a RoBERTa trained model (UA_roberta_final), their previous model that showed the best performance in COLIEE 2019 (UA_structure).
- **UEC (three runs)** [16] translates t1 texts into an easier one (t1p) with a paraphrase dictionary, extracts subject/predicate/object tuples from the main and conditional clauses, then constructs tuple-based similarity features

for ¡t1p, t2¿ pair (UEC1 and UEC2), for both the ¡t1, t2¿ and ¡t1p, t2¿ pairs (UECplus). LightGBM is used for binary classification.

5.4 Results

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

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

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

6 Final Remarks

In this paper we summarized the results of COLIEE 2020. Task 1 deals with the retrieval of noticed cases, and Task 2 poses the problem of identifying which paragraphs of a relevant case entail a given fragment of a new case. Task 3 is about retrieving articles to decide the appropriateness of the legal question, and Task 4 is a task to entail whether the legal question is correct or not. Ten (10) different teams participated in the case law competition (most of them in both tasks). We received results from 9 teams for Task 1 (a total of 22 runs), and 8 teams for Task 2 (a total of 22 runs). Regarding the statute law tasks, there were 14 different teams participating, most in both tasks. Eleven (11) teams submitted 28 runs for Task 3, and 13 teams submitted 30 runs for Task 4.

A variety of methods were used for Task 1: exploitation of the case structure information, deep learning based techniques (such as transformer methods and tools such as the Universal Sentence Encoder), lexical and latent features,

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

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

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

For Task 3, we confirmed that BERT-based approach improves overall retrieval performance. However, there are still numbers of questions that are dif-

ficult to retrieve by any systems. It is better to discuss the type of information necessary to find out the relationship between question and articles for the next step. For Task 4, overall performance of the submissions is still not sufficient to use their systems in real applications, mainly due to lack of coverage for some classes of problems, such as anaphora resolution. We found this task is still a challenging one to discuss and develop deep semantic analysis issues in the real application and natural language processing in general.

Acknowledgements

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

References

1. Alberts, H., Ipek, A., Lucas, R., Wozny, P.: COLIEE 2020: Legal information retrieval & entailment with legal embeddings and boosting. In: COLIEE (2020)
2. Aydemir, A., de Castro Souza, P., Gelfman, A.: Using BERT and TF-IDF to predict entailment in law-based queries. In: COLIEE (2020)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
4. Hayashi, R., Kiyota, N., Fujita, M., Kano, Y.: Legal bar exam solver integrating legal logic language proleg and argument structure analysis with legal linguistic dictionary. In: COLIEE (2020)
5. Hudzina, J., Madan, K., Chinnappa, D., Harmouche, J., Bretz, H., Vold, A., Schilder, F.: Information extraction & entailment of common law & civil code. In: COLIEE (2020)
6. Leburu-Dingalo, T., Thuma, E., Motlogelwa, N., Mudongo, M.: Ub_Botswana at COLIEE 2020 case law retrieval. In: COLIEE (2020)
7. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: Cedr: Contextualized embeddings for document ranking. In: SIGIR (2019)
8. Mandal, A., Ghosh, K., Pal, A., Ghosh, S.: Automatic catchphrase identification from legal court case documents. In: Proceedings of the Conference on Information and Knowledge Management. p. 2187–2190. ACM, New York, NY, USA (2017)
9. Mandal, A., Ghosh, S., Ghosh, K., Mandal, S.: Significance of textual representation in legal case retrieval and entailment. In: COLIEE (2020)
10. Nguyen, H.T., Vuong, H.Y.T., Nguyen, P.M., Dang, B.T., Bui, Q.M., Vu, S.T., Nguyen, C.M., Tran, V., Satoh, K., Nguyen, M.L.: JNLP team: Deep learning for legal processing in COLIEE 2020. In: COLIEE (2020)
11. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
12. Rabelo, J., Kim, M.Y., Goebel, R.: Application of text entailment techniques in COLIEE 2020. In: COLIEE (2020)
13. Shao, H.L., chia Chen, Y., Huang, S.: BERT-based ensemble model for the statute law retrieval and legal information entailment. In: COLIEE (2020)

14. Shao, Y., Liu, B., Mao, J., Liu, Y., Zhang, M., Ma, S.: THUIR@COLIEE-2020: Leveraging semantic understanding and exact matching for legal case retrieval and entailment. In: COLIEE (2020)
15. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. pp. 2–6 (2005)
16. Suematsu, Y., Matsuyoshi, S., Utsumi, A.: Recognizing textual entailment for japanese legal text using lexical simplication and tuple-based matching and similarity features. In: COLIEE (2020)
17. Wehnert, S., Murugadas, V., Nandakumar, S., Saha, A., Khan, T.R., Urban, M., Luca, E.W.D.: Legal information retrieval and entailment detection: Hybrid approaches of traditional machine learning and deep learning. In: COLIEE (2020)
18. Westermann, H., Šavelka, J., Benyekhlef, K.: Paragraph similarity scoring and fine-tuned BERT for legal information retrieval and entailment. In: COLIEE (2020)
19. Yoshioka, M., Suzuki, Y.: HUKB at COLIEE 2020 information retrieval task. In: COLIEE (2020)
20. Yurochkin, M., Claici, S., Chien, E., Mirzazadeh, F., Solomon, J.: Hierarchical optimal transport for document representation (2019)