A Summary of the COLIEE 2019 Competition

Juliano Rabelo[1,2], Mi-Young Kim[1,3], Randy Goebel[1,2], Masaharu Yoshioka[4,5,6], Yoshinobu Kano[7], and Ken Satoh[8]

[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, 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] Hokkaido University, Kita-ku, Sapporo-shi, Hokkaido, Japan
[7] Faculty of Informatics, Shizuoka University, Naka-ku, Hamamatsu-shi, Shizuoka, Japan
kano@inf.shizuoka.ac.jp
[8] National Institute of Informatics, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
ksatoh@nii.ac.jp

Abstract. We summarize the evaluation of the 6th Competition on Legal Information Extraction/Entailment (COLIEE 2019). The competition consists of four tasks: two on case law and two on 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 also includes an information retrieval task (Task 3) and an entailment/question answering task (Task 4), which attempts to confirm whether a particular statute applies to a yes/no question. Participation was open to any group in the world, based on any approach. Eleven different teams participated in the case law competition tasks, some of them in more than one task. We received results from 7 teams for Task 1 (15 runs) and 7 teams for Task 2 (18 runs). For the statute law tasks, 8 different teams participated, some in more than one task. Seven teams submitted a total of 13 runs for Task 3, and 7 teams submitted a total of 15 runs for Task 4. Here we summarize each team's approaches, our official evaluation, and some analysis of the variety of methods that produced the evaluation results.

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

1 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) is a series of evaluation competitions intended to build a research community, and to accelerate the development of the state of the art for information retrieval and entailment using legal texts. It is usually co-located with JURISIN, the Japanese Artificial Intelligence Society Juris-Informatics workshop series, which was created to promote community discussion on both fundamental and practical issues on legal information processing. The intention is to broadly embrace multiple disciplines, including law, social sciences, information processing, logic and philosophy, and the existing conventional “AI and law” area. In alternate years, COLIEE is organized as a workshop at the International Conference on AI and Law (ICAIL), which was the case in 2017 and 2019.

In COLIEE editions 2014 to 2017, there were two tasks (information retrieval (IR) and entailment) using Japanese Statute Law (civil law). Since COLIEE 2018, two new tasks (IR and entailment) were introduced, which use Canadian case law (Tasks 1 and 2).

Task 1 is a legal case retrieval task, and it involves reading a new case Q, and identifying supporting cases S1, S2, ..., Sn from the provided case law corpus, hypothesized to support the decision for Q. Task 2 is a legal case entailment task, which involves the identification of a paragraph or paragraphs from existing cases, which are alleged to 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 also ask the participants to submit a ranked list of relevant article results to inform a detailed discussion on the difficulty of the questions. For the entailment task (Task 4), we analyze accuracy of case analysis to expose issues with characterization of case attributes, in addition to evaluation of accuracy as in previous COLIEE tasks.

The rest of the paper is organized as follows: Sections 2, 3, 4, 5 decribe 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 identifies 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 285 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 official COLIEE test dataset has 61 cases has their golden labels, disclosed only after the competition results were published. Table 1 summarizes the properties of those datasets.

Table 1: Summary for the Case Law Retrieval Task Datasets

Property	Training	Testing
Number of base cases	285	61
Total number of candidate cases	57,000	12,200
Total number of noticed cases	1486 (2.60%)	330 (2.70%)

2.3 Approaches

Seven teams submitted a total of 15 runs for this task. Deep learning techniques and machine learning based classifiers were commonly used. More details on these alternative approaches are described below:

- **CACJ (one run)** [3] applies a machine learning based classifier using features extracted from the cases header (i.e., it does not consider any of the case contents).
- **CLArg (one run)** [17] describes an approach based on vector representation of cases, in combination with two different classifiers: random forests and k-nearest neighbours.
- **HUKB (one run)** [26] improved their previous system, used on the 2018 COLIEE edition (which was based on the use of structural information which considers a case as composed of three sections: header, facts and footer), by incorporating the use of case metadata: date, to exclude candidates more recent than the base case, and topics.
- **IITP (three runs)** [4] uses a combination of Deep Learning techniques, such as Doc2Vec, and Information Retrieval techniques, such as BM25, to tackle the task 1 challenge.
- **ILPS (three runs)** [21] combines text summarizing and a generalized language model (BERT) in order to assess pairwise relevance. To overcome a limitation of the framework on handling text fragments longer than 512 tokens, the authors apply summarization techniques over the case contents. The generated embeddings are then used as input to an MLP classifier.
- **JNLP (three runs)** [23] applies a summarization model that encodes a document into a continuous vector space, which embeds the summary properties of the document. The authors combine such encoded representation with latent and lexical features extracted from different parts of a given query and its candidates.
- **UA (three runs)** [19] developed an approach based on the use of the Universal Sentence Encoder to generate a vector representation of both the base case and each candidate, followed by the calculation of a similarity score using a cosine measure (this approach was used as the baseline for this task).

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 (precision: 0.3333, recall: 0.3443, for a threshold of 0.57 minimum similarity). The actual results of the submitted runs by all participants are shown on table 2, from which it can be seen that only 1 team could not reach the baseline.

Table 2: Results attained by all teams on the test dataset of task 1.

Team	Submission File	Precision	Recall	F1-score
JNLP	JNLP.task_1.pl.txt	0.6000	0.5545	0.5764
JNLP	JNLP.task_1.ple.txt	0.6000	0.5545	0.5764
JNLP	JNLP.task_1.p.txt	0.5934	0.5485	0.5701
ILPS	BERT_Score_0.946.txt	0.6810	0.4333	0.5296
HUKB	task1.HUKB	0.7021	0.4000	0.5097
ILPS	BM25_Rank_6.txt	0.4672	0.5182	0.4914
ILPS	BERT_Score_0.96.txt	0.8188	0.3424	0.4829
IITP	task1.IITPdocBM.txt	0.6368	0.3879	0.4821
IITP	task1.IITPBM25.txt	0.6256	0.3848	0.4765
CLArg	CLarg.txt	0.9266	0.3061	0.4601
IITP	task1.IITPd2v.txt	0.4653	0.3455	0.3965
UA	UA_0.57.txt	0.3560	0.3333	0.3443
UA	UA_0.52.txt	0.3513	0.3364	0.3437
UA	UA_0.54.txt	0.3639	0.3242	0.3429
CACJ	submit_task1_CACJ01.csv	0.2119	0.5848	0.3110

Table 2 shows JNLP attained the best result for the F1-score. CLArg had the best score when only precision is considered, whereas CACJ had the best recall score. The F1-score for CLArg and CACJ, however, were not among the best ones for this task, which shows the difficulty of finding the right balance in order to achieve good overall performance in this task.

3 Task 2 - Case Law Entailment

3.1 Task Definition

Given a base case and an extracted specific fragment together with a second case that is relevant in respect 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 181 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 must be identified is the paragraph(s) that entail the base-case-entailed fragment. The test dataset has 44 cases and was initially released without the golden labels, which were only disclosed after the competition results were published. Table 3 summarizes the properties of those datasets.

Table 3: Summary for the Case Law Entailment Task Datasets

Property	Training	Testing
Number of base cases	181	44
Total paragraphs in the related cases	5,814	1,448
Total true entailing paragraphs	202 (3.47%)	45 (3.10%)

3.3 Approaches

Seven teams submitted a total of 18 runs to this task. The most used techniques were those based on transformer methods, such as BERT [2] or ELMo [18]. More details on the approaches are show below.

- **IeLab** [9] (three runs)] used an IR-based technique which selects terms from the entailed fragments and the candidates using inverse document frequency and part of speech information.
- **IITP (three runs)** [4] describes an approach which uses BM25, an Information Retrieval technique, and Doc2Vec, a Deep Learning based technique, for this task.
- **JNLP (three runs)** has not submitted a paper describing the details of their approach for task 2, but they devised deep learning based methods for other tasks of COLIEE 2019 (e.g., [16]).
- **TRCase (one run)** [13] applies a ranking algorithm which uses word embeddings and textual similarity features to determine entailment relationships between a candidate paragraph and an entailed fragment. The authors observe that the set of selected features provide better results when applied to a ranking approach, rather than a supervised classifier.
- **TTCL (three runs)** presents an approach based on a generalized language model using BERT for the case law entailment task, and compared that approach with an SVM baseline approach. To overcome the framework

[9] This is an interesting approach worth further investigation, however the paper describing the method lacked important information and thus was not accepted for publication on the COLIEE proceedings

limitation of 512 tokens, the authors apply BERT at a sentence level, considering a paragraph to be an entailing example when one or more sentences are classified as entailing one or more sentences from the entailed fragment.

- **UA (three runs)** [19] proposes an approach which relies the extraction of similarity measures between the candidate paragraph and the entailed fragment; the application of BERT on those two pieces of text; use of a threshold-based classifier; and post-processing the results considering the a priori probability determined by the data distribution on the training samples.
- **UBLTM (two runs)** has not submitted a paper describing the details of their approach.

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 (precision: 0.1375, recall: 0.2444, for a threshold of 0.75 minimum similarity). The actual results of the submitted runs by all participants are shown on table 4, from which it can be seen that only 2 runs had a performance worse than the baseline score (however, the teams which sent those submissions also got better results on other runs).

Table 4: Results attained by all teams on the test dataset of task 2.

Team	Submission File	Precision	Recall	F1-score
UA	UA_0.400000.txt	0.6538	0.7556	0.7010
UA	UA_0.250000.txt	0.6364	0.7778	0.7000
IITP	task2.iitpBM25.txt	0.7045	0.6889	0.6966
UA	UA_0.300000.txt	0.6296	0.7556	0.6869
TRCase	TRCase_colie_test_submission_task2	0.6818	0.6667	0.6742
IITP	task2.iitp2docBM.txt	0.6591	0.6444	0.6517
JNLP	JNLP.task_2.lex.txt	0.5909	0.5778	0.5843
TTCL	uncased758256.txt	0.4000	0.8000	0.5333
TTCL	uncased758voted.txt	0.3882	0.7333	0.5077
TTCL	uncased758512.txt	0.3780	0.6889	0.4882
ielab	ielabsen.txt	0.4545	0.4444	0.4494
ielab	ielabphrase.txt	0.3409	0.3333	0.3371
ielab	ielabterm.txt	0.2273	0.2222	0.2247
UBLTM	UBLTM_T2_2.txt	0.1273	0.6222	0.2113
UBLTM	UBLTM_T2_1.txt	0.1182	0.5778	0.1962
JNLP	JNLP.task_2.cls-elmo.txt	0.1364	0.1333	0.1348
JNLP	JNLP.task_2.cls-elmobert.txt	0.0682	0.0667	0.0674
IITP	task2.iitp2D2v.txt	0.0455	0.0444	0.0449

From Table 4, one can see UA attained the best result for the F1-score, the official metric used in this task. However, IITP and TRCase achieved comparable results for the F1-score. It is also worth noting that IITP attained the best score considering only precision, and TTCL got the best recall score.

4 Task 3 - Statute Law Information Retrieval

4.1 Task Definition

This task involves reading a legal bar exam question Q, and identification of a subset of Japanese Civil Code Articles S_1, S_2,..., S_n from the entire Civil Code which are those appropriate for answering the question such that

$$Entails(S_1, S_2, ..., S_n, Q) \text{ or } Entails(S_1, S_2, ..., S_n, \text{ not } Q).$$

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

4.2 Dataset

For task 3, questions related to Japanese civil law were selected from the Japanese bar exam. The organizers provided a data set used for previous bar law exams, translated to English [10, 9, 8, 25] as training data (717 questions), with new questions selected from the 2018 bar exam as test data (98 questions). The number of questions classified by the number of relevant articles is listed in Table 5.

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

number of relevant article(s)	1	2	3	5	total
number of questions	80	15	2	1	98

4.3 Approaches

The following seven teams submitted 13 runs in total. Four teams (HUKB, JNLP, KIS and UA) had participated in previous editions, and three teams (DBSE, EVORA and IITP) were new competitors. Common techniques used in the system were well known IR engine mechanisms such as elasticsearch [10], Terrier [12], Indri [22], gensim [11], scikit-learn [12] with various scoring function such as TF-IDF, BM25. For the indexing, the most common method was ordinal word base indexing with stemming. Several teams use N-gram, word sequence, Word2Vec [15] and Doc2Vec[11].

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

[11] https://radimrehurek.com/gensim/

[12] https://scikit-learn.org

- **DBSE (one run)** [24] used BM25 scoring of elasticsearch and Word2Vec [15] based similarity scoring. They finally select the one or more results from them.
- **EVORA (three runs)** [20] uses Terrier IR platform with different scoring function with two query sets (original and keyword selection) and two article database (original articles and keyword selected articles).
- **HUKB (one run)** [26] uses sentence structure analysis to extract condition part and argument part of the query and articles and compare the similarity using Indri IR system. Final results are calculated by SVMRank using those features.
- **KIS (two runs)** [5] uses Doc2Vec [11] for generating document embedding vector and calculate similarity among query and articles. They also use TF-IDF to select important keywords for generating document embedding. Final results are selected by considering the score difference between the top ranked and candidate documents.
- **IITP (two runs)** [4] uses BM25 module of gensim and tfidf module of scikit learn.
- **JNLP (two runs)** [1] proposed to use different indexing method for TF-IDF calculation (N-gram, verb-phrase, noun-phrase) and calculate similarity using cosine similarity. Final results are selected by considering the score difference between the i-th ranked document and $i+1$-th ranked one.
- **UA (two runs)** [14] uses the TF-IDF model and language model as an IR module.

The teams which participated in the previous COLIEE proposed an extension or equivalent system for Task 3, and new teams proposed methods.

4.4 Results

Table 6 shows the evaluation results of submitted runs. The official evaluation measures used in this task were macro average of F2 measure, precision, and recall. 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 UA-TFIDF achieved the best F2 score among all submitted runs. KIS_2 had the highest recall score. For the longer ranked list, EVORA is better than others.

Figures 1, 2 and 3 show an average of evaluation measure for all submission runs. As we can see from Figure 1, there are many easy questions for which almost all systems can retrieve relevant articles. Figures 2 and 3 show there are also many queries for which none of the systems can retrieve relevant articles.

One of the example of this issue is H30-1-A: “An unborn child may not be given a gift on the donor’s death.” and relevant article is Article 3 “The enjoyment of private rights shall commence at birth.” There is no common words between the query and a relevant article and it requires knowledge about relationship between “commence at birth” and “unborn” for understanding the

Table 6: Evaluation results of submitted runs (Task 3) and the corresponding organizers' run

runid	lang	ret.	rel.	F2	Prec.	Rec.	MAP	R$_5$	R$_{10}$	R$_{30}$
DBSE	E	172	54	0.466	0.454	0.493	0.512	0.512	0.620	0.669
EVORA1	E	98	56	0.533	0.571	0.529	**0.628**	**0.669**	0.744	**0.851**
EVORA2	E	98	56	0.533	0.571	0.529	0.617	0.653	0.744	0.835
EVORA3	E	98	56	0.529	0.571	0.524	0.624	0.653	**0.752**	0.835
iitpBM25	E	98	48	0.447	0.490	0.442	0.541	0.620	0.669	0.760
iitptfidf	E	98	43	0.401	0.439	0.396	0.506	0.570	0.628	0.752
JNLP-tf	E	165	64	0.534	0.459	0.582	0.598	0.653	0.686	0.769
JNLP-tfnv	E	171	61	0.505	0.403	0.562	0.575	0.595	0.653	0.769
UA-LM	E	98	48	0.452	0.490	0.447	0.541	0.554	0.636	0.727
UA-TFIDF	E	98	58	**0.549**	**0.592**	0.544	0.618	0.620	0.694	0.760
HUKB	J	98	44	0.414	0.449	0.410	0.494	0.488	0.612	0.727
KIS	J	404	69	0.503	0.423	0.613	0.562	0.628	0.711	0.835
KIS_2	J	408	**72**	0.503	0.427	**0.637**	0.540	0.653	0.744	0.835

relationship. In order to analyze the improvement of the system for such difficult questions, it is necessary to compare the retrieval performance for such difficult queries.

5 Task 4 - Statute Law Entailment

5.1 Task Definition

Task 4 requires determination of 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 only pure entailment tasks, where t1 (relevant article sentences) and t2 (problem sentence) were given. Due to the limited number of available problems, COLIEE 2017 and 2018 did not retain this style of task. In the Task 4 of COLIEE 2019, 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 (717 questions) and new questions selected from the 2018 bar exam as test data (98 questions).

5.3 Approaches

The following seven teams submitted their results (15 runs in total). Two teams (KIS and UA) had experience in submitting results in the previous campaign. We describe each system's overview below.

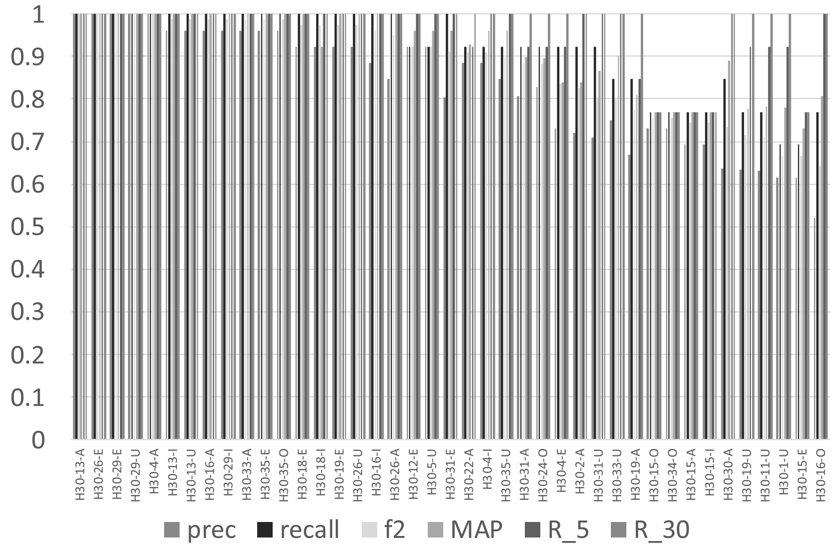

Fig. 1: Averages of precision, recall, F2, MAP, R_5, and R_30 for easy questions with a single relevant article

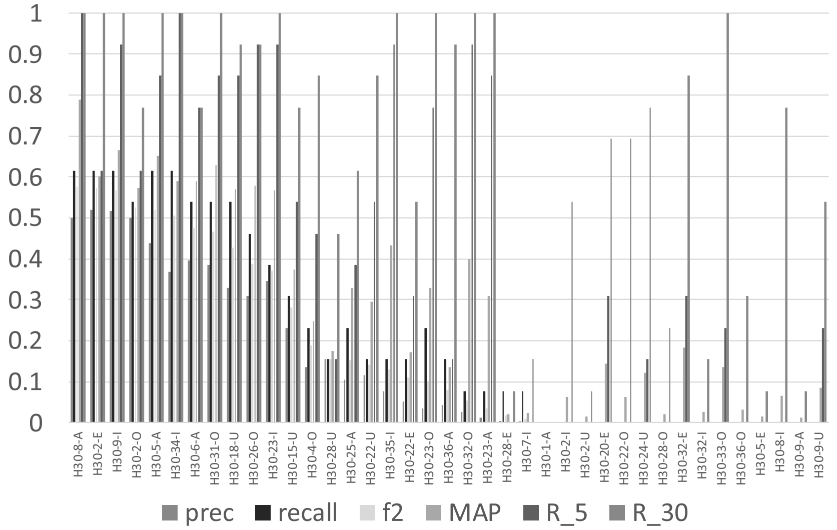

Fig. 2: Averages of precision, recall, F2, MAP, R_5, and R_30 for non-easy questions with a single relevant article

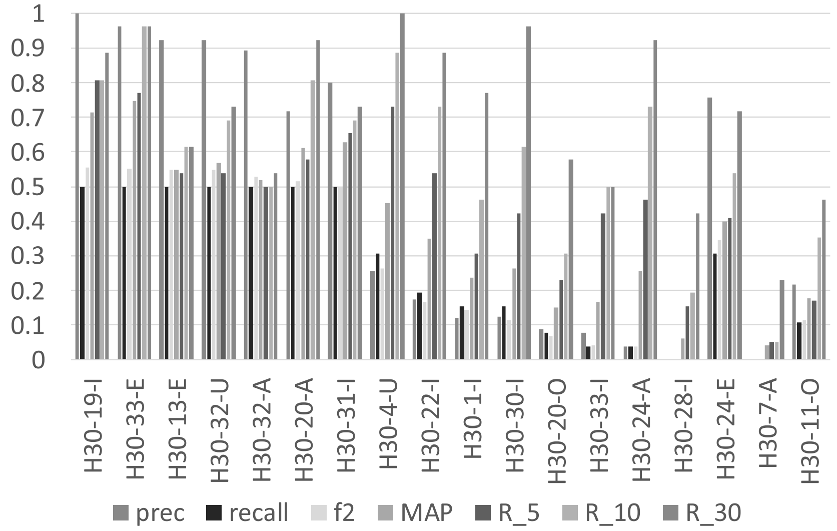

Fig. 3: Averages of precision, recall, F2, MAP, R_5, and R_30 for non-easy questions with multiple relevant articles

- **UA** [14] uses condition/conclusion/exception detection rules, and negation dictionaries created manually. They translated original Japanese texts into Korean by machine translation, employed their own Korean parser and Korean resources. **UA_Ex** uses Excite machine translation service, **UA_Go** uses Google machine translation service.
- **KIS** [6] parses sentences into predicate-argument structures to compare t1/t2 pairs, detecting negations and conditions. They use an ensemble of different comparison criteria (**KIS_3module**), then adding their own synonym dictionary (**KIS_dic**) or using FrameNet (**KIS_frame**).
- **IITP** [4] uses BERT with a BERT-base model.
- **DBSE** [24] uses an ensemble of stacked LSTMs.
- **JNLP** [16] indirectly solves the original problem with a derived problem with more abundant data. They trained using a stacked GRU.
- **TR** [7] uses BERT large model with decomposable attention (**TRAttn**), and similarity features (**TRSimFeat**).
- **EVORA** [20] used deep neural networks based methods, such as embedding by FastText (**EVORA1**), LSTM (**EVORA2**) and CNN (**EVORA3**).

5.4 Results

Evaluation was based on accuracy. Table 7 shows evaluation results of Task 4 for each submitted run. Because an entailment task is essentially 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 8 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 7: Evaluation results of submitted runs (Task 4)

Team	Dataset Language	# of correct answers (98 problems in total)	Accuracy
UA_Ex	Japanese	67	0.6837
KIS_3module	Japanese	61	0.6224
IITP	English	58	0.5918
KIS_dic	Japanese	58	0.5918
UA_Go	Japanese	58	0.5918
KIS_frame	Japanese	57	0.5816
DBSE	English	56	0.5714
JNLP.t=98	English	56	0.5714
TRAttn	English	55	0.5612
TRSimFeat	English	52	0.5306
JNLP.t=85	English	51	0.5204
EVORA1	English	50	0.5102
JNLP.t=78	English	48	0.4898
EVORA3	English	47	0.4796
EVORA2	English	44	0.4490

Although some cells show better results than others, none of the current systems could have solved problem types of more complex semantics, e.g., anaphora resolution. Overall we require a more precise survey of system differences, especially which components are more or less complete solutions that produce predictably correct results.

6 Final Remarks

We have summarized the results of the COLIEE 2019 competition. In case law, 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. In statute law, 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. Eleven (11) different teams participated in the case law competition (some of them in both tasks). We received results from 7 teams for Task 1 (a total of 15 runs), and 7 teams for Task 2 (a total of 18 runs). Regarding the statute law tasks, there were 8 different teams participating, some in both tasks. 7 teams submitted 13 runs for Task 3, and 7 teams submitted 15 runs for Task 4.

[13] Alphabetical letters stand for team names; a: DBSE, b: EVORA1, c: EVORA2, d: EVORA3, e: IITP, f: JNLP: t=78, g: JNLP(t=85), h: JNLP(t=98), i: KIS_3module, j: KIS_dic, l: TRAttn, m: TRSimFeat, n: UA_Ex, o: UA_Go, respectively. Team columns stand for their number of correct answers for corresponding category.

Table 8: Technical category statistics of questions, and correct answers of submitted runs for each category in numbers of counts and percentages[13].

category	#	a	%	b	%	c	%	d	%	e	%	f	%	g	%	h	%	i	%	j	%	k	%	l	%	m	%	n	%	o	%
Conditions	83	47	.57	44	.53	**34**	**.41**	38	.46	49	.59	40	.48	42	.51	46	.55	52	.63	50	.60	47	.57	48	.58	44	.53	**57**	**.69**	51	.61
Pers. role	66	36	.55	37	.56	**24**	**.36**	26	.39	40	.61	34	.52	35	.53	38	.58	43	.65	39	.59	41	.62	35	.53	34	.52	**46**	**.70**	37	.56
Pers. Reltnshp.	66	36	.55	37	.56	**24**	**.36**	26	.39	40	.61	34	.52	35	.53	38	.58	43	.65	39	.59	41	.62	35	.53	34	.52	**46**	**.70**	37	.56
Negation	44	27	.61	24	.55	20	.45	20	.45	26	.59	22	.50	**19**	**.43**	26	.59	26	.59	**29**	**.66**	23	.52	23	.52	22	.50	26	.59	24	.55
Entailment	33	18	.55	14	.42	18	.55	15	.45	16	.48	15	.45	**10**	**.30**	16	.48	**21**	**.64**	19	.58	19	.58	18	.55	16	.48	20	.61	18	.55
Dependency	28	12	.43	14	.50	**10**	**.36**	11	.39	14	.50	**10**	**.36**	17	.61	16	.57	19	.68	17	.61	18	.64	19	.68	12	.43	**21**	**.75**	17	.61
Ambiguity	26	11	.42	14	.54	**9**	**.35**	14	.54	12	.46	10	.38	15	.58	**17**	**.65**	15	.58	15	.58	11	.42	12	.46	14	.54	**17**	**.65**	13	.50
Legal terms	24	12	.50	11	.46	11	.46	14	.58	12	.50	**9**	**.38**	14	.58	**18**	**.75**	13	.54	12	.50	15	.63	15	.63	13	.54	17	.71	15	.63
Anaphora	22	12	.55	13	.59	**9**	**.41**	12	.55	13	.59	13	.59	13	.59	13	.59	16	.73	12	.55	14	.64	**9**	**.41**	**9**	**.41**	**18**	**.82**	14	.64
Verb paraphrs.	21	11	.52	12	.57	**8**	**.38**	**8**	**.38**	**15**	**.71**	10	.48	10	.48	11	.52	10	.48	13	.62	11	.52	12	.57	10	.48	13	.62	9	.43
Morpheme	18	13	.72	7	.39	**6**	**.33**	8	.44	13	.72	13	.72	11	.61	9	.50	12	.67	11	.61	13	.72	**14**	**.78**	12	.67	**14**	**.78**	**14**	**.78**
Case role	17	8	.47	10	.59	**6**	**.35**	8	.47	**12**	**.71**	7	.41	8	.47	10	.59	9	.53	7	.41	7	.41	11	.65	8	.47	9	.53	8	.47
Pred. argument	14	**5**	**.36**	7	.50	6	.43	7	.50	10	.71	7	.50	7	.50	8	.57	9	.64	7	.50	9	.64	9	.64	**5**	**.36**	**11**	**.79**	7	.50
Article search	11	5	.45	6	.55	4	.36	7	.64	8	.73	7	.64	6	.55	**3**	**.27**	8	.73	8	.73	5	.45	4	.36	4	.36	**10**	**.91**	7	.64
Paraphrase	11	4	.36	6	.55	5	.45	7	.64	3	.27	**1**	**.09**	7	.64	7	.64	9	.82	3	.27	**10**	**.91**	4	.36	7	.64	8	.73	5	.45
Itemized	8	**6**	**.75**	4	.50	5	.63	**3**	**.38**	5	.63	4	.50	**3**	**.38**	4	.50	4	.50	**6**	**.75**	4	.50	**6**	**.75**	5	.63	**6**	**.75**	5	.63
Normal terms	7	3	.43	3	.43	**2**	**.29**	**5**	**.71**	4	.57	**2**	**.29**	4	.57	4	.57	**5**	**.71**	**5**	**.71**	4	.57	3	.43	**2**	**.29**	**5**	**.71**	4	.57
Calculation	2	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	1	.50	2	1.00	2	1.00

A variety of methods were used for Task 1: classification using only features extracted from the case header, random forest and k-NN classifiers, 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, embedding summary properties, and information retrieval techniques were the main ones. For Task 2, transformer-based tools such as BERT and ELMo were prevalent, but IR techniques and textual similarity features have also been applied. 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 showed promising results this year, would not be so appropriate in a scenario where explainability is key. For future editions of COLIEE, we plan to continue to expand the data sets in order to improve the robustness of results, as well as introducing evaluation of explainability-aware tasks or requirements into the competition.

For Task 3, we found there are three types of questions in the test data (easy questions, difficult questions with vocabulary mismatch, and questions with multiple answers). Most of the submission systems are good at retrieving relevant answers for easy questions, but it is still difficult to retrieve relevant articles for other question types. It may be necessary to focus on such question types to improve the overall performance of the IR system. 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, and requires deeper analysis of semantic issues in the general application of natural language processing.

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. Dang, T.B., Nguyen, T., Nguyen, L.M.: An approach to statute law retrieval task in coliee-2019. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
3. El Hamdani, R., Troussel, A., Houvenagel, C.: Coliee case law competition task 1: The legal case retrieval task. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
4. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: Iitp@coliee 2019: Legal information retrieval using bm25 and bert. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
5. Hayashi, R., Kano, Y.: Searching relevant articles for legal bar exam by doc2vec and tf-idf. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
6. Hoshino, R., Kiyota, N., Kano, Y.: Question answering system for legal bar examination using predicate argument structures focusing on exceptions. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
7. Hudzina, J., Vacek, T., Madan, K., Tonya, C., Schilder, F.: Statutory entailment using similarity features and decomposable attention models. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
8. Kano, Y., Kim, M.Y., Goebel, R., Satoh, K.: Overview of coliee 2017. In: Satoh, K., Kim, M.Y., Kano, Y., Goebel, R., Oliveira, T. (eds.) COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment. EPiC Series in Computing, vol. 47, pp. 1–8. EasyChair (2017). https://doi.org/10.29007/fm8f, https://easychair.org/publications/paper/Fglr
9. Kim, M.Y., Goebel, R., Kano, Y., Satoh, K.: Coliee-2016: evaluation of the competition on legal information extraction and entailment. In: International Workshop on Juris-informatics (JURISIN 2016) (2016)
10. Kim, M.Y., Goebel, R., Satoh, K.: Coliee-2015: evaluation of legal question answering. In: Ninth International Workshop on Juris-informatics (JURISIN 2015) (2015)
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML'14, JMLR.org (2014), http://dl.acm.org/citation.cfm?id=3044805.3045025

12. Macdonald, C., McCreadie, R., Santos, R.L., Ounis, I.: From puppy to maturity: Experiences in developing terrier. In: Proceedings of the SIGIR 2012 Workshop in Open Source Information Retrieval. pp. 60–63 (2012)
13. Madan, K., Hudzina, J., Vacek, T., Schilder, F., Custis, T.: Textual entailment using word embeddings and linguistic similarity. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
14. Mi-Young, K., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the 17th International Conference on Artificial Intelligence and Law. ICAIL'2019 (2019)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
16. Nguyen, H.T., Tran, V., Nguyen, L.M.: A deep learning approach for statute law entailment task in coliee-2019. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
17. Paulino-Passos, G., Toni, F.: Retrieving legal cases with vector representations of text. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
18. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
19. Rabelo, J., Mi-Young, K., Goebel, R.: Combining similarity and transformer methods for case law entailment. In: Proceedings of the 17th International Conference on Artificial Intelligence and Law. ICAIL'2019 (2019)
20. Raiyani, K., Quaresma, P.: Keyword machine learning based japanese statute law retrieval and entailment task at coliee-2019. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
21. Rossi, J., Kanoulas, E.: Legal information retrieval with generalized language models. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
22. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: a language-model based search engine for complex queries. Tech. rep., in Proceedings of the International Conference on Intelligent Analysis (2005)
23. Tran, V., Nguyen, M.L., Satoh, K.: Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In: Proceedings of the 17th International Conference on Artificial Intelligence and Law. ICAIL'2019 (2019)
24. Wehnert, S., Hoque, S.A., Fenske, W., Saake, G.: Threshold-based retrieval and textual entailment detection on legal bar exam questions. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)
25. Yoshioka, M., Kano, Y., Kiyota, N., Satoh, K.: Overview of japanese statute law retrieval and entailment task at coliee-2018. In: The Proceedings of the 12th International Workshop on Juris-Informatics (JURISIN2018). pp. 117–128. The Japanese Society of Artificial Intelligence, (2018)
26. Yoshioka, M., Song, Z.: Hukb at coliee 2019 information retrieval task - utilization of metadata for relevant case retrieval. In: Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE'2019 (2019)