COLIEE-2018: Evaluation of the Competition on Case Law Information Extraction and Entailment

Mi-Young Kim[1,2], Yao Lu[2], Juliano Rabelo[2], and Randy Goebel[2,3]

[1]Department of Science, Augustana Faculty, University of Alberta, Camrose, AB, Canada
[2]Alberta Machine Intelligence Institute, University of Alberta, Edmonton, AB, Canada
[3]Department of Computing Science, University of Alberta, Edmonton, AB, Canada
{miyoung2, yao1, rabelo, rgoebel}@ualberta.ca

Abstract. We summarize the evaluation of the case law component of the 5th Competition on Legal Information Extraction/Entailment 2018 (COLIEE-2018). The COLIEE-2018 task includes two tasks in each of statute law and case law. The case law component includes an information retrieval (Task 1), and the confirmation of an entailment relation between an existing case and an unseen case (Task 2). Participation was open to any group based on any approach, and the task attracted 13 teams. We received 6 submissions for Task 1 (for a total of 12 runs), and 4 submissions for the Task 2 (for a total of 8 runs).

Keywords: case law information retrieval, recognizing textual entailment, case law information entailment, AI and law, Juris-informatics

1. Introduction

The Juris-Informatics workshop series 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 exsiting conventional "AI and law" area.

Information extraction and reasoning from legal data is one of the important targets of JURISIN, including legal information representation, relation extraction, textual entailment, summarization, and their applications. Participants in the JURISIN workshops have examined a wide variety of information extraction techniques and environments with a variety of purposes, including retrieval of relevant articles, entity/relation extraction from legal cases, reference extraction from legal cases, finding relevant precedents, summarization of legal cases, and legal question answering.

During the last four years, we held four competitions on legal information extraction/entailment (COLIEE 2014-2017) on a legal data collection, and this helped establish a major experimental effort in the legal information

extraction/retrieval field. We held the fifth competition (COLIEE-2018)[1] this year, with the motivation of continuing to help create a research community of practice for the capture and use of legal information. The previous COLIEE competitions focused on the legal question answering task, based on analysis of Japanese bar law exams and Japanese legal statutes. This year, we have extended the competition to include tasks on case law.

Four tasks are included in the 2018 competition: Tasks 1 and 2 are about case law, and tasks 3 and 4 are about statute law. Here we will introduce the case law competition (Tasks 1 and 2). Task 1 is a legal case retrieval task, and it involves reading a new case Q, and extracting supporting cases S1, S2, …, Sn from the provided case law corpus, 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 the decision of a new case. In the next sections, we will describe each task in detail, explain participants' systems, and assessment results.

2. COLIEE Case Law Competition Tasks

COLIEE-2018 data is drawn from an existing collection of predominantly Federal Court of Canada case law, provided by vLex Canada (http://ca.vlex.com). Participants can choose which phase they will apply for, between the two tasks as follows:

Task 1: Legal case retrieval task. Input is an unseen legal case Q, and output is relevant cases in the given legal corpus alleged to support the decision of the input case Q.

Task 2: Confirming an entailment relation between the decision of a new case and a relevant case. Input is a decision paragraph, a short summary and the full contents from an unseen case and a relevant case. Output is selected paragraphs from a relevant case, which entail the decision of the unseen case.

2.1. Task 1: Case law retrieval task

Our goal is to explore and evaluate case law retrieval technologies that are both effective and reliable. The task investigates the performance of systems that search a set of legal cases that support a previously unseen case description. The goal of the task is to accept a query and return noticed cases in the given collection. We say a case is 'noticed' with respect to a query *iff* the case supports the decision of the query case. In this task, the query case does not include a decision, because our goal is to determine how accurately a machine can capture decision-supporting cases for a new case (with no decision).

The process of executing the new query cases over the existing cases and then generating the experimental runs should be entirely automatic. In the training data, each query case is used with a pool of legal cases, and the noticed cases in the

[1] http://www.ualberta.ca/~miyoung2/COLIEE2018/

pool are produced as the answer. In test data, only query cases and a pool of case laws will be included, with no noticed case information.

The format of the COLIEE case law competition data in Task 1 is as follows:

<pair id="t1-1">
<query content_type="summary" description="The summary of the case created by human expert.">
The parties to this consolidated litigation over the drug at issue brought reciprocal motions, seeking that the opposing party be compelled to provide a further and better affidavit of documents ... (omitted)
</query>
<query content_type="fact" description="The facts of the case created by human expert.">
[1] Tabib, Prothonotary: The Rules relating to affidavits of documents should be well known by litigants. Yet it seems that parties are either not following them strictly, or are assuming that others are not ... (omitted)
</query>
<cases_noticed description="The corresponding case id in the candidate cases">
18,45,130
</cases_noticed>
<candidiate_cases description="The candidate cases indexed by id">
<candidate_case id="0"> Case cited by: 2 cases Charest v. Can. (1993)... (omitted)
</candidate_case>
<candidate_case id="1"> Case cited by: one case Chehade, Re (1994), 83 F.T.R. 154 (TD) ... (omitted)
</candidate_case>
... (omitted)
<candidate_case id="199"> Desjardins v. Can. (A.G.) (2004), 260 F.T.R. 248 (FC) MLB headnote ... (omitted)
</candidate_case>
</candidate_cases> </pair>

The above is an example of Task 1 training data where query id “t1-1” has 3 noticed cases (IDs: 18, 45, 130) out of 200 candidate cases. The test corpora will not include a <cases_noticed> tag information. Out of the given candidate cases for each query, participants are required to retrieve noticed cases. The candidate cases are made of noticed cases and randomly sampled cases from the database. For those randomly sampled cases, we use vLex Canada's developed algorithm to make sure no relevant cases are identified as sampled cases. Furthermore, two legal experts checked them manually.

2.2. Task 2: Case law entailment task

Our goal in Task 2 is to predict the decision of a new case by entailment from previous relevant cases. As a simpler version of predicting a decision, a decision of a new case and a noticed case will be given as a query. Then a case law textual entailment system must identify which paragraph in the noticed case

entails the decision, by comparing the extracting and compairing the meanings of the query and paragraph.

The task evaluation measures the performance of systems that identify a paragraph that entails the decision of an unseen case. Training data consists of a triple: a query, a noticed case, and a paragraph number of the noticed case by which the decision of the query is alledgedly entailed. The process of executing queries over the noticed cases and generating the experimental runs should be entirely automatic. Test data will include only queries and noticed cases, but no paragraph numbers.

The format of the COLIEE competition data in Task 2 is as following:

<**pair id="t2-1**">
<**query**>
<**case_description content_type="summary" description="The summary of the case created by human expert."**>
The applicant owned and operated the Inn on the Park Hotel and the Holiday Inn in Toronto ... (omitted)
</case_description>
<**case_description content_type="fact" description="The facts of the case created by human expert.">**
... </**case_description**>
<**decision description="The decision of the query case."**> The applicant submits that it is unreasonable to require the applicant to produce the information and documentation referred to in the domestic Requirement Letter within 62 days ... (omitted)
</decision>
<**cases_noticed description="The supporting case of the basic case**">
<**paragraph paragraph_id="1"**>
[1] Carruthers, C.J.P.E.I. : This appeal concerns the right of the Minister of National Revenue to request information from an individual pursuant to the provisions of s. 231.2(1) of the Income Tax Act , S.C. 1970-71-72, c. 63. Background
</paragraph>
<**paragraph paragraph_id="2">**
[2] The appellant, Hubert Pierlot, is the main officer and shareholder of Pierlot Family Farm Ltd. which carries on a farm operation in Green Meadows, Prince Edward Island.
</paragraph>
... (omitted)
<**paragraph paragraph_id="26"**>
[26] I would, therefore, dismiss the appeal. Appeal dismissed. Editor: Steven C. McMinniman/vem [End of document]
</**paragraph**>
</cases_noticed>
</query>
<**entailing_paragraph description="The paragraph id of the entailed case.">13</entailing_paragraph>**
</pair>

The above is an example of Task 2 training data, and the example says that a decision in the query was entailed from the paragraph No. 13 in the given noticed case. The decision in the query does not comprise the whole decision of the case. This is a decision for a portion of the case, and a paragraph that supports the decision should be identified in the given noticed case. The test corpora will not include the <entailing_paragraph> tag information, and participants are required to identify the paragraph number which entails the query decision.

3. Evaluation Metrics and Baselines

The measures for ranking competition participants are intended only to calibrate the set of competition submissions, rather than provide any deep performance measure. The data sets for Tasks 1 and 2 are annotated, so simple information retrieval measures (precision, recall, F-measure, accuracy) can be used to rank each submission. As noted above, the intention is to build a community of practice regarding case law textual entailment, so that the adoption and adaptation of general methods from a variety of fields is considered, and that participants share their approaches, problems, and results.

For Tasks 1 and 2, evaluation measure will be precision, recall and F-measure:

For Task 1:

Precision =(the number of correctly retrieved cases for all queries)/(the number of retrieved cases for all queries),

Recall = (the number of correctly retrieved cases for all queries)/(the number of noticed cases for all queries),

F-measure = (2 x Precision x Recall)/(Precision + Recall).

For Task 2:

Precision =(the number of correctly retrieved paragraphs for all queries)/(the number of retrieved paragraphs for all queries),

Recall = (the number of correctly retrieved paragraphs for all queries)/(the number of relevant paragraphs for all queries),

F-measure = (2 x Precision x Recall)/(Precision + Recall).

For Tasks 1 and 2, we consider the term cosine similarity as the baseline model. Table 1 presents the performances of the baseline model.

Table 1. Baseline performances of Tasks 1 and 2

Tasks	Task 1	Task 2
Precision of Term cosine similarity	0.2649	0.0405
Recall of Term cosine similarity	0.4102	0.5094
F-measure of Term cosine similarity	0.3219	0.0751

4. Submitted Runs and Results

In the overall case law competition, 13 teams registered, 6 teams submitted their system results in Task 1 (for a total of 12 runs), and 4 teams submitted their results in Task 2 (for a total of 8 runs). Some participants submitted multiple runs for a task.

The training data consists of 285 case queries for Task 1, and each query has 200 candidate noticed cases. In the Task 1 training data, each case query has average 9.87 noticed cases. For the task 2 training data, 181 queries have been provided and each query has average 48.59 candidate paragraphs for recognizing entailment relation. In the Task 2 training data, average 1.32 paragraphs have an entailment relation with a query. For the Task 1 test data, 59 case queries and 200 candidate noticed cases for each query were provided where each query has average 10.66 noticed cases. For Task 2 test data, 43 case queries were provided with average 47.19 candidate paragraphs for each query. In the Task 2 test data, average 1.23 paragraphs have an entailment relation with a query.

Tables 2 and 3 summarize a brief description of the submitted systems' techniques. We present the results achieved by runs against the Information Retrieval and Entailment subtasks in Tables 4 and 5, respectively.

Draijer and Verberne (system id: UL) [1] used a Random Forest classifier with eight different features for Task 1. The eight features are More Like This Score on Facts, More Like This Score on Summary, Doc2vec Cosine Similarity distance to Facts, Doc2vec Cosine Similarity distance to Summary, TF-IDF Euclidean distance to Facts, TF-IDF Euclidean distance to Summary, TF-IDF Cosine similarity distance to Facts, and TF-IDF Cosine similarity distance to Summary.

Chen et al. (system id: Smartlaw) [2] proposed using association rules in both Tasks 1 and 2. They first experimented with a machine learning-based model adopting Word2Vec/Doc2Vec as features. But machine learning methods have several disadvantages for this task: first, the tasks have very limited training samples, which make current machine learning models hard to achieve good performance. Second, the space consumption of datasets and the computational cost of training exponentially increase when the size of data expands. To enhance

Table 2. Approaches of submitted systems for Task 1

ID	Run	Approaches
HUKB[4]	HUKB1	Using all case parts, building queries from the summary data and considering all candidates for the database. Returning a variable number of results for each query.
	HUKB2	Using all case parts, building queries from the summary data and considering all candidates for the database. Returning a fixed number of results for each query.
JNLP[3]	JNLP-r=2.5	Combining lexical features and latent features embedding summary properties (parameter range r is 2.5 which determines the interval for selection)
	JNLP-k=10	Combining lexical features and latent features embedding summary properties (the average number of noticed cases is 10)
Smartlaw[2]	Smartlaw	Co-occurrence association model
UA[5]	UA	Pairwise paragraph similarity based features
	UA-postproc	Pairwise paragraph similarity based features with post processing
	UA-smote	Pairwise paragraph similarity based features augmenting the training data with SMOTE
UBIRLED [6]	UBIRLED-1	k-Nearest neighbor search with TFIDF for ranking. Elimination of 75% of the lowest scoring candidate cases
	UBIRLED-2	Ranking with TFIDF then filtering candidates with scores lower than a threshold calculated by the average score of the top 5 documents, divided by 2.
	UBIRLED-3	Ranking with TFIDF and k-Nearest neighbor search, then filtering candidates with scores lower than a threshold calculated by the average score of the top 5 documents, divided by 2.
UL[1]	UL	Random Forest classifier with eight different features

the scalability of the solutions, they propose two association rule models: what is labelled as basic association rule model, and another co-occurrence association rule model. The basic association rule model considers only the similarity between the source document and the target document, and it does not leverage a

Table 3. Approaches of submitted systems for Task 2

ID	Run	Approaches
Smartlaw [2]	Smartlaw	Co-occurrence association model
UA[5]	UA	Similarity based features fed to a Random Forest classifier with 250 estimators
	UA-100	Similarity based features fed to a Random Forest classifier with 100 estimators
	UA-500	Similarity based features fed to a Random Forest classifier with 500 estimators
UBIRLED [6]	UBIRLED-1	Keywords were extracted using Python Keyphrase Extraction toolkit. Then a K-NN search with TF-IDF was used for ranking.
	UBIRLED-2	Facts, Decision, and Paragraphs were mixed together to formulate a query and then UBIRLED-1 approach was used.
	UBIRLED-3	Facts and Summary were mixed together to formulate a query and then UBIRLED-1 approach was used.
UNCC0	UNCC0	Ensemble machine learning with SMOTE resampling technique

manually labeled relevancy dictionary. The co-occurrence association rule model uses a relevancy dictionary in addition to the basic association rule model.

Tran et al. (system id: JNLP) [3] explored benefits from analyzing legal documents' summaries and logical structures for Task 1. They extended the summary of both the query and the candidates to include more attributes from fact/paragraphs. They propose to obtain document embedding information guided by the document summary. This information is used to estimate the phrasal scores for each document given their summary and paragraphs. Subsequently, they train the model with the summary acting as gold catchphrases and paragraphs acting as document sentences. After building the trained model, they generate a latent summary in continuous vector space. For the ranking of candidates, they use two selection strategies: hard top k, and flexible bound relative to score deviation.
UNCC0 applied ensemble learning using the following classifiers: logistic regression, XGBoost classifier, Random forest classifier, and Support Vector Machine classifier. They used resampling of input data using SMOTE for further training.
Yoshioka and Song (system id: HUKB) [4] built an IR system for the task 1 by using the following two steps to retrieve the referred cases: first (1) they build a ranked retrieval, using an IR system to rank candidates. Since the input queries are full text case laws consisting of several parts (summary, citations, paragraph list, etc), they experimented using different parts for building the target database and the queries. They also analyzed the effect of building one database per query (using only the given candidates for that query), and then building one database using all candidates. Their best performance was achieved when the database

used all available case parts; the queries used only the summary and the database was constructed with all candidates. In their second technique (2) from a selection of the referred cases, they choose which of those cases returned in step (1) are going to be used as their system's answer. They tried two strategies: first, select the top *n* ranked cases (*n* fixed a priori), then select a variable number of cases by checking the similarity with non-related cases.

Rabelo et al. (system id: UA) [5] modeled tasks 1 and 2 as binary classification problems. For Task 1, they constructed feature matrices by using a cosine similarity measure between paragraphs from the query case and each candidate case. Those matrices were then transformed into fixed size feature vectors via a histogram approach with pre-determined score bounds, and given to a Random Forest classifier. They also applied post processing to leverage statistical a priori knowledge. Since the dataset in Task 1 is very imbalanced, they under-sampled the dominant class and over-sampled the rarer class by synthesising samples with SMOTE. Their approach for Task 2 was also based on extracting similarity-based features from the query and noticed cases, and feeding those features to a Random Forest classifier.

Lefoane et al. (system id: UBIRLED) [6] propose an approach based on Information Retrieval and unsupervised learning to Task 1: TFIDF is used as a similarity measure between a query and candidate cases. A k-nearest neighbor search with TFIDF as a distance measure is also used. They first rank documents according to their relevance to the query, then apply filtering to exclude the lowest scoring documents from relevant cases, using a threshold value to cut off non-relevant case judgments.

In Table 4, we can see that most systems show better performance than the baseline model. The JNLP system shows the best performance combining lexical features and latent features embedding summary properties (limiting the average number of noticed cases to 10), and it achieved significant increase of the F-measure compared to other systems.

HUKB1 and HUKB2 systems extracted 194 and 191 cases as noticed cases. JNLP-r=2.5 and JNLP-k=10 systems extracted 412 and 399 cases. The Smartlaw system extracted 271 cases, UA, UA-postproc, and UA-smote systems extracted 203, 254, and 247 cases, UBIRLED-1, UBIRLED-2, and UBIRLED-3 systems extracted 392, 453, and 64 cases, and UL system extracted 190 cases. Even though JNLP systems extracted the most cases amongst the systems, they showed the best precision performance. In Task 1, many participants used machine learning classifiers, but the system which used more sophisticated features such as a combination of lexical features and latent features embedding summary properties showed the best performance in this year's competition.

Table 5 reports the results of Task 2, where UA and UA-500 showed the best performance, which is significantly better than the baseline performance. The UA and UA-500 systems used similarity-based features input to a Random Forest

Table 4. IR results(Task 1) on the formal run data

Run	Prec.	Recall	F-m.	Run	Prec.	Recall	F-m.
Baseline	0.2649	0.4102	0.3219	UA-postproc [5]	0.3484	0.4038	0.3741
HUKB1 [4]	0.4974	0.3084	0.3808	UA-smote [5]	0.3539	0.3927	0.3723
HUKB2 [4]	0.4047	0.3037	0.3470	UBIRLED-1 [6]	0.1329	0.6232	0.2191
JNLP-r=2.5[3]	0.5464	0.6550	0.5958	UBIRLED-2 [6]	0.1955	0.7202	0.3075
JNLP-k=10 [3]	**0.6763**	**0.6343**	**0.6546**	UBIRLED-3 [6]	0.5614	0.1017	0.1723
Smartlaw [2]	0.2871	0.4308	0.3446	UL [1]	0.5638	0.3021	0.3934
UA [5]	0.3725	0.3227	0.3458				

Table 5. Entailment results (Task 2) on the formal run data

Run	Prec.	Recall	F-m.	Run	Prec.	Recall	F-m.
Baseline	0.0405	0.5094	0.0751	UBIRLED-1[6]	0.0484	0.8302	0.0914
Smartlaw [2]	0.0465	0.1509	0.0711	UBIRLED-1[6]	0.0495	0.9245	0.0940
UA[5]	**0.2381**	**0.2830**	**0.2586**	UBIRLED-1[6]	0.0467	0.7925	0.0881
UA-100[5]	0.1905	0.2264	0.2069	UNCC0	0.0330	0.0566	0.0417
UA-500[5]	**0.2381**	**0.2830**	**0.2586**				

classifier with different number of estimators. Among the 8 systems, 6 systems showed better performance than the baseline model on Task 2. Task 2 was much difficult than Task 1, and even humans have difficulty in choosing the correct paragraph with the appropriate entailment relations. We can also see the task is difficult based on the low performance on all the systems.

The Tasks 1 and 2 has been newly created in this year's competition, and we think there are many rooms for improvement, such as the evaluation method of Task 2, imbalanced data set, small size set of data which have limitations in applying machine learning techniques, etc. We hope to solve these limitations step-by-step for next competition, to get more robust performances for each Task.

5. Conclusion

We have summarized the results of the COLIEE-2018 competition. Two Tasks were evaluated: (1) Task 1: retrieving noticed cases (information retrieval), and (2) Task 2: extracting paragraphs of relevant case which entail the conclusion of a new case. There were 13 teams who participated in this competition, and we received results from 7 teams. There were 6 submissions to Task 1 (for a total of 12 runs), and 4 submissions to Task 2 (for a total of 8 runs).

A variety of methods were used for Task 1: combining lexical features and latent features embedding summary properties, creating queries from the summaries of cases, and building an information retrieval system to extract noticed cases, co-occurrence association model, pairwise paragraph similarity computation, K-NN, TF-IDF, and a Random forest classifier. Various features were also proposed: features from summary properties, Word2Vec, Doc2Vec, More Like This Score, cosine similarity, Euclidean distance, etc. For Task 2, co-occurrence association model, similarity-based features fed to a random forest classifier, and ensemble machine learning with SMOTE resembling techniques were used. Even though most systems outperformed baseline, all the performances are low, and the task didn't make it easy to identify relevant useful attributes.

For future competitions, we will need to expand the data sets in order to improve the robustness of results. We also need to more deeply investigate how to extract good features for Task 2.

Acknowledgements

This research was supported by Alberta Machine Intelligence Institute (AMII). Thanks to Colin Lachance from vLex for his constant support in the development of the case law data set, and to support from Ross Intelligence and Intellicon.

References

1. Wilco Draijer and Suzan Verberne, “Case law retrieval with doc2vec and Elastic search”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **UL**)
2. Ying Chen, Yilu Zhou, Zhen Lu, Hao Sun and Wenjun Yang, “Legal Information Retrieval by Association Rules”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **Smartlaw**)
3. Vu Tran, Son Truong Nguyen and Minh Le Nguyen, “JNLP Group: Legal Information Retrieval with Summary and Logical Structure Analysis”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **JNLP**)
4. Masaharu Yoshioka and Zihao Song, “HUKB at COLIEE2018 Information Retrieval Task”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **HUKB**)
5. Juliano Rabelo, Mi-Young Kim, Housam Babiker and Randy Goebel, “Legal Information Extraction and Entailment for Statute Law and Case Law”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **UA**)

6. Moemedi Lefoane, Tshepho Koboyatshwene and Lakshmi Narasimhan, “KNN CLUSTERING APPROACH TO LEGAL PRECEDENCE RETRIEVAL”, Twelfth International Workshop on Juris-informatics (JURISIN), 2018 (System id: **UBIRLED**)