EPiC Series in Computing

Volume 47, 2017, Pages 1–8

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Overview of COLIEE 2017

Yoshinobu Kano[1], Mi-Young Kim[2], Randy Goebel[2], and Ken Satoh[3]

[1] Faculty of Informatics, Shizuoka University, Japan

[2] University of Alberta, Canada

[3] National Institute of Informatics, Japan

kano@inf.shizuoka.ac.jp, {miyoung2, rgoebel}@ualberta.ca, ksatoh@nii.ac.jp

Abstract

We present the evaluation of the legal question answering Competition on Legal Information Extraction/Entailment (COLIEE) 2017. The COLIEE 2017 Task consists of two sub-Tasks: legal information retrieval (Task 1), and recognizing entailment between articles and queries (Task 2). Participation was open to any group based on any approach, and the tasks attracted 10 teams. We received 9 submissions to Task 1 (for a total of 17 runs), and 8 submissions to Task 2 (for a total of 20 runs).

1 Introduction

During the last three years, we held the first, second and third competitions on legal information extraction/entailment, COLIEE 2014 , COLIEE 2015 (Kim et al., 2015), COLIEE 2016 (Kim et al., 2016), on a legal data collection, and this helped establish a major experimental effort in the legal information extraction/retrieval field. We held this fourth competition (COLIEE 2017) this year, with the motivation of continuing to help create a research community of practice for the capture and use of legal information. The COLIEE competition is about the legal question answering, which combines the two tasks of information retrieval and textual entailment.

We have previously held our COLIEE competitions in conjunction with the Japanese AI Society Juris-Informatics (JURISIN) workshop. The JURISIN workshop series was created to promote community discussion on both fundamental and practical issues on legal information processing, with the intention to embrace various backgrounds such as law, social science, information processing, logic and philosophy, and including the conventional "AI and law" area. This year's COLIEE 2017 is being held in conjunction with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017). Participating teams include HUKB(Yoshioka & Onodera, 2017), iLis7 (Heo et al., 2017), iLis9 (Jung et al., 2017), JAISTNLP (Nguyen et al., 2017), JNLP (Carvalho et al., 2017), KIS (Kano et al., 2017), NOR (Nanda et al., 2017), UA (Kim & Goebel, 2017), and NAIST (Morimoto et al., 2017).

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 1–8

2 The Legal Question Answering Task

This competition focuses on two aspects of legal information processing related to answering yes/no questions from Japanese legal bar exams (the relevant data sets have been translated from Japanese to English, and were available in both languages).

1) Task 1 of the legal question answering Task involves reading a legal bar exam question Q, and extracting a subset of Japanese Civil Code Articles S_1, S_2,..., S_n from the entire Civil Code, which are appropriate for answering the question such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and the entire Civil Code Articles, we have to retrieve the set of "S_1, S_2, ..., S_n" as the outcome of Task 1.

2) Task 2 of the legal question answering Task involves the identification of an entailment relationship such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and articles S_1, S_2, ..., S_n, we have to determine if relevant articles entail "Q" or "not Q". The answer of this track is binary: "YES"("Q") or "NO"("not Q"). This phase typically requires some information extraction (e.g., named entity identification, relation extraction, etc.), followed by any variety of methods for textual inference, to confirm entailments. Details are given in the next section.

2.1 Task 1

Our goal is to explore and evaluate legal document retrieval technologies that are both effective and reliable. The Task investigates the performance of systems that search a static set of civil law articles using previously unseen queries, and return relevant articles. We say an article is "Relevant" to a query if and only if the query sentence can be entailed from the meaning of the article. If combining the meanings of more than one article (e.g., "A," "B," and "C") can answer a query sentence, then all the articles ("A," "B,", and "C") are considered "Relevant." If a query can be answered by an article "D," and it can be also answered by another article "E" independently, we also say that both "D" and "E" are "Relevant." This Task requires the retrieval of *all* the articles that are relevant to answering a query.

Japanese civil law articles (and English translation) have been provided, and training data consists of query and relevant article pairs. The process of executing the queries over the articles and generating the experimental runs should be entirely automatic. Test data includes only queries but no relevant articles.

2.2 Task 2

The goal of Phase 2 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. The Task investigates the performance of systems that answer "Y" or "N" to previously unseen queries by somehow comparing the intersection of meanings between queries and relevant articles. Training data consists of triples: a query, relevant articles and a correct answer "Y" or "N." For the competition evaluation, the process of finding relevant articles, executing the queries over the relevant articles and generating the experimental runs should be entirely automatic. Test data includes only queries and corresponding "Y/N" labels for each query.

3 Legal Question Answering Data Corpus

The corpus of legal questions is drawn from Japanese Legal Bar exams, and the relevant Japanese Civil Law articles have been also provided.

1) Task 1 problem is to use an identified set of legal yes/no questions to retrieve relevant Civil Law articles. In this case, the correct answers have been determined by a collection of law students, and those answers are used to calibrate the performance of a program to solve Task 1.

2) Task 2 requires some method of information extraction from both the question and the articles, and then to confirm a simple entail relationship as described in the Section 2: either the articles confirms "yes" or "no" as an answer to the yes/no questions.

Participants can choose which task they will attempt, amongst the two tasks as follows:

Task 1: legal information retrieval task. Input is a bar exam "Yes/No" question and output should be relevant civil law articles.

Task 2: Recognizing Entailment between law articles and queries. Input is a question and the entire set of articles, and output should be "Yes" or "No.

Table 1 shows an example of query and relevant articles. Table 2 shows examples of sentence pairs holding different entailment relations.

Question	*A person who made a manifestation of intention which was induced by duress emanated from a third party may rescind such manifestation of intention on the basis of duress, only if the other party knew or was negligent of such fact.*
Related Article	*(Fraud or Duress) Article 96 (1)Manifestation of intention which is induced by any fraud or duress may be rescinded.(2)In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.(3)The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.*

Table 1. Example of a query and a relevant article

Question	A special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty.
Related Article	(Special Agreement Disclaiming Warranty)Article 572 Even if the seller makes a special agreement to the effect that the seller will not provide the warranties set forth from Article 560 through to the preceding Article, the seller may not be released from that responsibility with respect to any fact that the seller knew but did not disclose, and with respect to any right that the seller himself/herself created for or assigned to a third party.
Label	Yes
Question	A manager must engage in management exercising care identical to that he/she exercises for his/her own property.

Related Article	(Urgent Management of Business)Article 698 If a Manager engages in the Management of Business in order to allow a principal to escape imminent danger to the principal's person, reputation or property, the Manager shall not be liable to compensate for damages resulting from the same unless he/she has acted in bad faith or with gross negligence.
Label	No

Table 2. Examples of sentence pairs holding different entailment relations

The structure of the test corpora is derived from a general XML representation developed for use in RITEVAL, one of the tasks of the NII Testbeds and Community for Information access Research (NTCIR) project*.

The RITEVAL format was developed for the general sharing of information retrieval on a variety of domains. The format of the COLIEE competition corpora is derived from an NTCIR representation for confirmed relationships between questions and the articles and cases relevant to answering the questions, as in the following example:

<pair label="Y" id="H18-1-2">
<t1>
(Seller's Warranty in cases of Superficies or Other Rights) Article 566 (1) In cases where the subject matter of the sale is encumbered with for the purpose of a superficies, an emphyteusis, an easement, a right of retention or a pledge, if the buyer does not know the same and cannot achieve the purpose of the contract on account thereof, the buyer may cancel the contract. In such cases, if the contract cannot be cancelled, the buyer may only demand compensation for damages. (2)The provisions of the preceding paragraph shall apply mutatis mutandis in cases where an easement that was referred to as being in existence for the benefit of immovable property that is the subject matter of a sale, does not exist, and in cases where a leasehold is registered with respect to the immovable property.(3)In the cases set forth in the preceding two paragraphs, the cancellation of the contract or claim for damages must be made within one year from the time when the buyer comes to know the facts.
(Seller's Warranty in cases of Mortgage or Other Rights)Article 567(1) If the buyer loses his/her ownership of immovable property that is the object of a sale because of the exercise of an existing statutory lien or mortgage, the buyer may cancel the contract.(2)If the buyer preserves his/her ownership by incurring expenditure for costs, he/she may claim reimbursement of those costs from the seller.(3)In the cases set forth in the preceding two paragraphs, the buyer may claim compensation if he/she suffered loss.
</t1>
<t2>
There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.
</t2>
</pair>

The above is an example where query id "H18-1-2" is confirmed to be answerable from article numbers 566 and 567 (relevant to Task 1). The pair label "Y" in this example means the answer of query is "Yes," which is entailed from the relevant articles (relevant to Task 2).

* As described at http://sites.google.com/site/ntcir11riteval/

The COLIEE training data was built from the bar exam (short answer test) civil code part published from 2006-2015, and consists of 10 XML files. Each file corresponds to one year's publication. The total number of queries in the training data is 570. The test data size is 78 queries (extracted from the bar exam of 2016).

4 Evaluation Metrics

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 Task 1 are annotated, so simple information retrieval measures (precision, recall, F-measure, accuracy) can be used to rank each submission. The intention is to start to build a community of practice regarding legal 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 Task 1, evaluation measure will be precision, recall and F-measure:

- Precision = (the number of correctly retrieved articles for all queries)/(the number of retrieved articles for all queries),
- Recall = (the number of correctly retrieved articles for all queries)/(the number of relevant articles for all queries),
- F-measure = (2 x Precision x Recall)/(Precision + Recall).

For Task 2, the evaluation measure will be accuracy, with respect to whether the yes/no question was correctly confirmed:

- Accuracy = (the number of queries which were correctly confirmed as true or false)/(the number of all queries).

5 Submitted Runs and Results

Overall, 10 teams submitted their system results. Some participants submitted multiple runs for a task. We received 9 submissions to Task 1 (for a total of 17 runs), and 8 submissions to Task 2 (for a total of 20 runs).

ID	Approaches
HUKB	article structure analysis, phrase matching, rank SVM with 15 similarity scores and alignment scores, ensemble
iLis7-1	TF-IDF, LSM, LDA, Word2Vec, LSA
JAISTNLP	TF-IDF
JNLP	ranking related n-gram collections, term order probabilities, relevance disambiguation.
NOR	LDA
UA	TF-IDF, language model

Table 3. Approaches of submitted systems for IR (Task 1)

ID	Approaches
iLis9	TF-IDF, negation detection
JAISTNLP	ranking, encoding-based and attention neural network
KIS	case-role based linguistic analysis, predicate-argument structures
NAIST	Word2vec, attention neural network
NOR	CNN, LSTM
UA	Korean dependency parser, Excite Japanese/Korean machine translation, semantic dictionary, k-means clustering

Table 4. Approaches of submitted systems for textual entailment (Task 2)

Tables 3 and 4 summarize the submitted systems' techniques, including systems when corresponding proceedings exist. We present the results achieved by runs against the Information

Run	Prec.	Recall	F	L.	Run	Prec.	Recall	F	L.
HUKB-1	0.658	0.472	0.550	J	JNLP1-T	0.500	0.354	0.414	E
HUKB-2	0.586	0.490	0.534	J	KID17	0.703	0.518	0.596	E
HUKB-3	0.551	0.536	0.543	J	KIS-IE-M	0.263	0.272	0.267	J
iLis7-1	0.734	0.554	0.632	E	KIS-IE-NM	0.346	0.245	0.287	J
iLis7-2	0.654	0.500	0.567	E	NOR17	0.462	0.500	0.480	E
JAISTNLP2-1a-norerank	0.628	0.445	0.521	E	UA-LM	0.602	0.427	0.500	E
JAISTNLP2-1b-rerank	0.615	0.436	0.510	E	UA-TFIDF	0.666	0.472	0.553	E
JNLP1-R	0.686	0.536	0.602	E	VNPT	0.430	0.281	0.340	E
JNLP1-RT	0.689	0.545	0.609	E					

Table 5. IR results (Task 1) on the formal run data. Prec, F, and L stands for Precision, F-measure, and Data Language respectively. E and J stands for English and Japanese in the Language columns.

Run	Accuracy	Language	Run	Accuracy	Language
iLis7	0.564	English	KIS-YN-CM	0.538	Japanese
iLis9-1	0.576	English	KIS-YN-CS	0.589	Japanese
iLis9-2	0.538	Japanese	KIS-YN-M	0.576	Japanese
JAISTNLP2-2a-1a-norerank	0.512	English	KIS-YN-S	0.653	Japanese
JAISTNLP2-2a-1b-rerank	0.474	English	NAIST1	0.615	Japanese
JAISTNLP2-2b-1a-norerank	0.487	English	NAIST2	0.653	Japanese
JAISTNLP2-2b-1b-rerank	0.500	English	NAIST3	0.474	Japanese
JNLP1-R	0.435	English	NOR17	0.538	English
JNLP1-RT	0.487	English	UA-LM	0.717	Japanese
KIS-YN-A	0.538	Japanese	UA-TFIDF	0.692	Japanese

Table 6. Entailment results (Task 2) on the formal run data

Retrieval and Entailment tasks in Tables 5 and 6.

We performed comparisons between the top three runs in Task 2. Among 78 queries, the best run correctly answered 56 queries. When comparing the best and the second, matched answers were 45 and 41 queries for each second team. These three runs agreed in only 27 queries, where 25 were correct answers. This comparison implies that the 25 queries could be similar in surficial level, which are easy to answer correctly regardless of employed methods. It is difficult to determine why around 20 queries are non-agreed, of which at least one of these three teams answered correctly. It might be due to the difference of the methods. However, if we assume that the 25 queries are easy-to-answer, then the there are 53 that remain. This means that we can answer 26 queries correctly even in a chance rate, which is higher than the 20 non-agreed queries. Although we would need more evaluation data to obtain more stable statistic evaluation, this comparison suggests that there is still lots of opportunity to improve the competition systems in future.

There will be a live competition held in the conference as our first trial. Each team is required to run the very system used to submit the formal run results, while asked to return their answers in real time.

6 Conclusion

We have summarized the results of the COLIEE 2017 competition. Two tasks were evaluated: (1) Task 1: finding relevant articles (information retrieval) (2) Task 2: answering yes/no questions given a query (textual entailment).

There were 10 teams who participated in this competition. There were 9 submissions to Task 1 (for a total of 17 runs), and 8 submissions to Task 2 (for a total of 20 runs).

While the evaluation scores are getting higher from previous years, there are still many unresolved issues in these tasks, as suggested by the absolute evaluation scores and comparison results.

Acknowledgements

This research was partially supported by MEXT Kakenhi Japan, JST CREST Japan, and the Alberta Machine Intelligence Institute (Amii).

References

Carvalho, D. S., Tran, V., Tran, K. Van, & Nguyen, L. M. (2017). Improving Legal Information Retrieval by Distributional Composition with Term Order Probabilities. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Heo, S., Hong, K., & Rhim, Y.-Y. (2017). Legal Content Fusion for Legal Information Retrieval. In *the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Jung, B., Soh, C., Hong, K., Lim, S., & Rhim, Y.-Y. (2017). Multiple Agent Based Entailment System (MABES) for RTE. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Kano, Y., Hoshino, R., & Taniguchi, R. (2017). Analyzable Legal Yes/No Question Answering System using Linguistic Structures. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Kim, M.-Y., & Goebel, R. (2017). Two-step Cascaded Textual Entailment for Legal Bar Exam Question Answering. In *the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Kim, M.-Y., Goebel, R., Kano, Y., & Satoh, K. (2016). COLIEE-2016: Evaluation of the Competition on Legal Information Extraction/Entailment. In *Tenth International Workshop on Juris-informatics (JURISIN 2016)*.

Kim, M.-Y., Goebel, R., & Ken, S. (2015). COLIEE-2015: Evaluation of Legal Question Answering. In *Ninth International Workshop on Juris-informatics (JURISIN 2015)*. Keio University, Yokohama, Japan.

Morimoto, A., Kubo, D., Sato, M., Shindo, H., & Matsumoto, Y. (2017). Legal Question Answering System using Neural Attention. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Nanda, R., John, A. K., Caro, L. Di, Boella, G., & Robaldo, L. (2017). Legal Information Retrieval Using Topic Clustering and Neural Networks. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Nguyen, T.-S., Phan, V.-A., & Nguyen, L.-M. (2017). Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

Yoshioka, M., & Onodera, D. (2017). A Civil Code Article Information Retrieval System based on Phrase Alignment with Article Structure Analysis and Ensemble Approach. In *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), 16th International Conference on Artificial Intelligence and Law (ICAIL 2017)*.

EPiC Series in Computing

Volume 47, 2017, Pages 9–22

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

A Civil Code Article Information Retrieval System based on Phrase Alignment with Article Structure Analysis and Ensemble Approach

Masaharu Yoshioka and Daiki Onodera

Graduate School of Information Science and Technology, Hokkaido University, Hokkaido, Japan
yoshioka@ist.hokudai.ac.jp, onodera@kb.ist.hokudai.ac.jp

Abstract

In this paper, we introduce a system for COLIEE task phase 1 that retrieves relevant civil code article(s) for making correct entailment to the questions of Japanese Bar Exam. This system is an extended version of our previous system that based on legal terminology and civil code article structure. However, the performance of the previous system is not as good as best performance system of the task. In this paper, we introduce concept of phrase alignment that takes into account the civil code article structure. In addition, due to the variations of the question types, the settings that are good for particular type of questions may not be good for other types of questions. Therefore, we propose to use systems with different settings and generate final answer by aggregating the output of different systems based on ensemble approach. Finally, we also discuss the difference between English task and Japanese task based on the retrieval results of Indri, one of the state-of-the-art information retrieval system.

1 Introduction

COLIEE task phase 1 is a task to retrieve relevant articles from Japanese civil code for answering questions in Japanese Bar Exam. In COLIEE 2016[KGKS16], we propose an information retrieval (IR) system based on legal terminology and civil code article structure [OY16].

However, performance of the system is not as good as the best performance system of the COLIEE 2016 [KHJ+16]. Longest words sequence (LWS)[AMHKV99] is one of the feature that is used in the best performance system used and is not used by our system. LWS may be a strong clue for the cases that questions and articles share important phrases. However, LWS analysis for Japanese task is not work well as in the case of English task. One of the reason is difference of description style uniformity between Japanese and English texts. Japanese civil code was originally put into operation in 1890, and revised many times until now. Therefore, the description style are not so uniform compared to the English one that were translated at the same time. So we proposed a method for phrase alignment that uses normalization of the surface description as a pre-process and use extended version of Needleman-Wunsch algorithm [NW70] that allows mismatch and gap in the phrase alignment results.

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 9–22

Based on the preliminary experiment results, utilization of phrase alignment improves the retrieval performance when the questions and related articles shares similar phrases. However, usage of phrase alignment is harmful for the case that description style of the questions is totally different from the related article. In order to solve this problem, we implement retrieval systems with different settings using SVM-Rank[Joa06] and aggregate results to generate final retrieval results based on ensemble approach.

Another issue discussed in this paper is the comparison between Japanese task and English task. COLIEE task uses two different datasets. One is a Japanese dataset that uses original Japanese civil code and original Bar exam questions. The other is an English dataset that uses those texts translated by manually. Since those datasets are parallel corpus, the related articles for the question and result of entailment are same for Japanese and English datasets. However, due to the difference between description style uniformity, task difficulty between these two datasets are different. Based on the comparison between basic retrieval performance of those two datasets by using state-of-the-art IR system Indri [SMTC05][1], we confirm the retrieval performace for the English and Japanese dataset are different based on the evaluation dataset (training data classified by year). In most of the case English results are better than Japanese one. It is necessary to take into account this factor when we compare the results obtained from the Japanese dataset and the English one.

Rest of the paper are divided into four parts. Section 2 reviews our IR system for COLIEE 2016 and compare its characteristics with the best performance system in COLIEE 2016. Section 3 introduces our new IR system for COLIEE 2017. Section 4 reports the result of retrieval experiments including COLIEE 2017, and discuss its characteristics with reference to the comparative analysis between Japanese and English datasets by using Indri. Section 5 concludes this papers.

2 IR system in COLIEE 2016

2.1 Our previous IR system for COLIEE 2016

Followings are assumptions about characteristics of COLIEE 2016 phase 1 task to implement our IR system[OY16].

- Existence of important keywords that should be included in the related article
  When the question contains keywords from specific legal terminology, those keywords are more important than the others and those keywords are expected to be included in the related article.

- Existence of two different parts (condition and others) in the questions and articles
  In Japanese civil code, there are general provisions that cover wider cases and specific articles that override general provisions for particular cases. It is better to compare the description about condition part of the questions with one of the articles.

Based on these assumptions, we implemented IR system for Japanese Bar Exam question answering based on ABRIR [Yos10]. This system calculates basic similarity of question and documents by using Okapi/BM25 [RW00] and use a Boolean query to calculate penalty when the articles don't satisfy the Boolean query. Calculation of penalty is a similar concept of word overlap used in [KHJ+16]. This system uses following indexes for calculating similarity.

[1] https://www.lemurproject.org/indri/

- Index keyword type
  Japanese morphological analyzer MeCab [2] is used for splitting Japanese sentences into morphemes. From this results, we construct following three types of keywords sets for index.
  - Compound words of legal terminology (compounds)
    Keywords extracted from "(Kommentar Civil Code)" [3] are used as candidates for legal terminology. However, there are several terms that are split into morphemes. For those compound words, we compare the sequence of morphemes with candidates and select combined morphemes for index keywords.
  - Words used in legal terminology (elems)
    We split keywords of compound words into morphemes by using MeCab and use all split keywords as keywords that characterizes legal terminology.
  - All words (words)
    Words with following POS type; noun (excluding pronoun, number, non-independent, suffix), verbs, adjectives, and adverbs are used for index. We also use stop words list (e.g., general verb ((do)(become)) and general noun ((that))) for excluding meaningless keyword.
- Result of article structure analysis
  Condition parts of articles and questions are identified by pattern matching (e.g., "(If)", "(When)") to the dependency parsing results generated by CaboCha[KM02]. When both question and article have condition parts, similarity between condition parts, rest parts similarity and all parts are calculated separately. In such a case linear combination of those three similarity are used as a similarity score. On the contrary, when there is no condition part for question or article, similarity between all parts are used as a similarity score.

Similarity scores between questions and articles are calculated by using Okapi/BM25.

$$\sum_{t \in q} \left[log \frac{N}{df_t} \right] \cdot \frac{(k_1+1)tf_t}{k_1 \frac{L_d}{L_{ave}} + tf_t} \cdot \frac{(k_3+1)tf_q}{k_3 + tf_q} \tag{1}$$

where, N is the count of all articles of civil code, df_t is the document frequency of the term, L_d is the document length, L_{ave} is the average length of all documents, tf_t is the term frequency in the document, tf_q is the term frequency in the question, and k_1 and k_3 are control parameters ($k_1 = 1, k_3 = 1000$ are used in the system).

For the penalty calculation, almost same equation (removing factors related to the articles) is used. β is control parameters for balancing the factor between similarity score and penalty score.

$$Penalty(w) = \beta \cdot \left[log \frac{N}{df_t} \right] \cdot \frac{(k_3+1)tf_q}{k3 + tf_q} \tag{2}$$

Another extension for this task is handling mutatis mutandis articles. In Japanese civil codes, there are specific types of articles that describe certain juristic act by referring to the article with similar or equal effect, such as "AXY ("A" shall apply mutatis mutandis to the

[2] http://taku910.github.io/mecab/

[3] https://ja.wikibooks.org/wiki/%e3%82%b3%e3%83%b3%e3 %83%a1%e3%83%b3%e3%82 %bf%e3%83%bc%e3%83%ab%e6%b0%91%e6%b3%95

case from Article X to Article Y)". Since these articles don't have description about such effect explicitly, it is difficult to retrieve such articles. In order to solve this problem, we construct virtual articles by combining description about the case description in the mutatis mutandis articles and referred article. For example, if the previous article is article number Z, virtual articles (X+Z,$\cdots$,Y+Z) are constructed; contents of X+Z is "A" + whole contents of article X. At the retrieval time, when such a virtual article(X+Z) is top ranked one, the system returns two articles (X and Z), instead of top one article for the usual article case.

We also try to use Japanese WordNet [BIF+09] for query term expansion. However, such query term expansion is not so appropriate for the case that articles and questions share important keywords. As a result, retrieval performance of the system with query term expansion was worse than the system without expansion.

In the COLIEE 2016, varieties of control parameters sets were examined by using training data and the best performance system of COLIEE 2016 uses "elems" (words used in legal terminology) for calculating similarity by equation 1 and uses "words" for calculating penalty by equation 2.

2.2 The best performance system in COLIEE 2016

Our system described in previous section is a second-best system in COLIEE 2016 in terms of F-measure. In order to clarify the issues for the improvement, we briefly review the best performance system [KHJ+16]. One of the main differences between the best performance system and ours are datasets. This system uses English dataset and ours uses Japanese dataset. For the features for calculating similarity, most of the features used in the system are variations of features used in our system. However, there are several features and techniques that were not used in our previous system.

For the lexical similarity, they use Longest Words Sequence (LWS)[AMHKV99] as a feature. This feature works well when the questions and related articles shares important phrases to identify the similarity. For the syntactic similarity they use role similarity based on the sentence parts such as subject, verb and object.

Another important difference is that the best performance system uses ensemble approach that uses different features and machine learning methods.

3 IR system for COLIEE 2017

3.1 Features for new IR system

Based on the discussion in the previous system, we implement new IR system with following features.

- Article structure analysis
  In the previous system, only condition parts are identified by the article structure analysis. However, based on the analysis between the questions and related articles, we also identify the sentence parts that describes juristic act as main arguments in addition to the condition parts.

- Phrase alignment instead of LWS
  LWS is a strong clue to identify related articles when the questions and related articles shares important phrases. However, since description style of articles in Japanese civil code are not so uniform compared to English translated one, it is not useful to use LWS.

In this paper, we propose a new phrase alignment method that uses normalization of the term and allows gap and mismatch between the texts of a question and an article.

- Parameter tuning by using SVM-Rank[Joa06][4]
  In the previous experiments, parameter tuning related to the control parameters were conducted based on the exhaustive search based on generate and test for the training data. In order to avoid such exhaustive search, we use SVM-Rank with linear kernel to estimate the appropriate linear combination of the features.

- Ensemble approach to generate final answers
  Due to the varieties of the question types, it is not so easy to select useful features for all questions. For example, features related to LWS is a strong clue when the questions and related articles share the important phrases. However, it is harmful when the description style of the question is totally different from the related article. Therefore, it is not easy to make a simple IR system for all types of queries. In order to solve this problem, varieties of IR system that uses different feature sets are implemented and final results are generated by aggregating the result of these systems.

Figure 1 shows the workflow of the proposed system.

3.2 Article Structure Analysis

As a result of article structure analysis, the previous IR system extract condition parts from questions and related answers. However, it is not so effective to improve retrieval performance of the IR system. In order to find out good method to utilize article structure analysis, we reviewed the pairs of questions and related articles in Japanese Bar Exam datasets and found that there are two types of questions in the datasets.

- Question related to the appropriateness of juristic act
  These questions are simple type and comparison between questions and articles about the juristic act and condition are also important. Example of this question is H26-1-C “” (“A will made by an adult ward may be rescinded by guardian of the adult ward.”).

- Question related to the appropriateness of conditions for juristic act
  These questions are variation of the previous question. In this case, importance of the condition part may vary based on the detailed question type. For example, the questions are asked to check the existence of exceptional case of general provision, contents of a condition part are not so important. On the contrary, if the question check the appropriateness of the condition for the juristic act, a condition part are more important than previous question(H26-1-C). H19-17-1 “” (“There are cases when compensation of damages may be demanded besides demand for the enforcement of performance.”) is an example to check the existence of exceptional cases.

Therefore, we decide to extract two parts (i.e., main arguments that describe juristic act and condition) from the articles and questions to calculate the similarity.

Followings are examples of patterns to identify condition parts and main argument parts.

- Condition parts
  “”(if ...),“”(in case of), “”(case), “”(limited to) , ...

[4] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Figure 1: Workflow of the Proposed System

- Main argument parts
  “”(deemed to), “”(shall), “ ”(execute), “”(can), ...

3.3 Phrase Alignment

Longest Words Sequence(LWS) is one of a feature used in the best performance system of COLIEE 2016 for the English dataset. Therefore, we apply the same technique to the Japanese dataset. However, there are many cases that results of LWS may not extract important phrases for the questions and related articles pair, even though they use similar vocabulary.

One of the characteristic difference between Japanese and English datasets is description style uniformity. Since Japanese civil law was originally put into operation in 1890 and revised its contents year by year. As a result, there are varieties of description for the same concept; especially for the case of noun phrases related to verbs (e.g., “”-“”(transfer), “”-“”(refund)). In addition, such nouns can be used as original verbs by combining general verbs “” (do). For example, “” can be described as “”(“” + “”) or “”(“” + “” + “”). In such a case, since surface of those two terms are different, simple LWS cannot identify the similarity between questions and related articles.

Therefore, following two surface expression normalization methods are applied as a preprocess of the phrase alignment.

- Verb-noun normalization
  The list of verb-noun pairs (32 pairs including “”(transfer) and “”(refund)) are collected by using civil code articles and questions in the training data. All verbs are replaced with corresponding nouns(e.g., “” is replaced with “”) before the alignment.

- Positive expression normalization
  Since this retrieval task is find related article(s) that justify the description is correct or not, there are many cases that juristic act of related articles are contradictory to the question (it means entailment of the question is wrong). In order to support matching between those contradictory cases, sentence expression is normalized as positive expression. In this experiment, we use following two normalization word pairs; “”(ineffective) to “”(effective) and “”(relieve) to “”(responsible).

However, this normalization is not good enough to use LWS, because there are several cases that the usage of (postpositional particle) in a question is different from one in a related article. Therefore, we introduce sequence alignment algorithm that allows gaps and mismatch for this phrase alignment. Needleman-Wunsch algorithm [NW70] is one of the well-known algorithm used for DNA sequence alignment. In this framework, the user defines the similarity score between the elements and gap penalty and find out most appropriate matching by selecting an alignment pair with highest score.

In this experiment, similarity score between elements are calculated by surface expression, POS, and information obtained from dependency parser CaboCha[KM02]. Since we conduct exact matching between questions and articles in this phrase alignment, similarity score equals to 0 when the surface expression of question and article is different. In addition, alignments of contents words (nouns excluding number, pronoun and stop words (e.g., “”(that), “”(thing); verbs; adjectives, and adverbs) are more important than other words, we set basic similarity score for the contents words (Sim_c) and others (Sim_o) and others, 10 and 2 respectively.

In addition, since Japanese sentence have main argument at the end of the sentence, matching of the pairs close to the root node is more important than matching far from the root

node. In order to represent such difference, we introduce depth level discount $disc_{lv}$ for the matching. lv is a dependency nodes that exists between root node of the question and the corresponding parts of the question. In this experiment, basic similarity scores are discounted by 0.8 ($disc_{lv} = 0.8^{lv}$). We also set the gap penalty 0, because there are many cases that length of article and questions are not similar. In such cases, introduce positive numbers of gap penalty tends to calculate higher score for similar length article. Figure 2 shows an example of phrase matching.

Question
成年被後見人がした遺言は，Bが取り消すことができる

⇩ Dependency parser

	lv
成年 被 後見人 が-D	*4*
した-D	*3*
遺言は，-D	*2*
Bが-D	*3*
取り消す-D	*2*
こと が-D	*1*
できる	*0*

lv represents depth level between root node(できる) and words

Article:
成年 被 後見人 の 法律 行為 は 、 取り消す こと が できる

⇩ Similarity score table for finding highest score path using dynamic programming

lv	4	4		1	1	0
POS	Noun	Pre		Noun	PP	Verb
	成年	被	…	こと	が	できる
成年	4.1	0	…	0	0	0
被	0	0.82	…	0	0	0
…	…	…	…	…	…	…
こと	0	0	…	1.6	0	0
が	0	0	…	0	1.6	0
できる	0	0	…	0	0	10

Pre = Prefix, PP= Postpositional Particle
Score of "成年"=10(Sim_c)×0.8⁴($disc_{lv}^4$)
Score of "こと" (that) = 2(Sim_o)×0.8,
because it is a non-content noun.

Figure 2: Example of phrase matching

In this experiments, we conduct following three types of phrase matching. For all cases, the system returns phrase alignment results with score and the score is used for a similarity measure.

- Condition parts of question and articles
- Main argument parts of question and articles
- All (condition and main argument) parts of a question and title of the article
  Title of the article contains general description about the article and it is also used in

the questions. However, there are many articles that don't have such description in the articles. This phrase matching is helpful to find related article whose title phrases are shared in the question.

3.4 Utilization of SVM-Rank

In addition to the phrase alignment, we also use features based on the similarity measures used in the previous system. In the previous system, similarity score based on the article structure analysis results (condition, rest, all) and index keyword types (compound words, elems, words) are aggregated by using control parameter by using generate and test approach. In order to avoid such an exhaustive search method, we use SVM-Rank for calculating the final similarity score by aggregating these measures. Based on the preliminary experiment, we don't use compound words index in this system.

For the synonym expansion, we make a list of synonym for the keywords that only exist in the question (not in the articles). For making the list, we use Japanese word net. However, simple usage of the all synonym is not good in the previous system, we use word2vec [MSC+13] [5] trained by Japanese Wikipedia and 1,486 judicial precedent related to the civil code downloaded from the web site of Courts in Japan [6]. In this experiment, we select top 40 words whose word embedding vectors are close to the original keyword as candidates for the keyword expansion. From the candidates, we select synonyms of the original keyword defined in Japanese WordNet as query term replacement candidates. When the question have such (a) keyword(s), the system replace such keyword(s) with synonyms. In addition, Verb-noun normalization discussed in Section 3.3 is also applied for making index of the article and parsed results of the question.

In this experiment following 15 features are used. All similarity measures are normalized into (0..1) range by dividing the highest scores of the feature for each question.

- Alignment score based on phrase alignment
  1. Condition parts of an article and a question
  2. Main argument parts of an article and a question
  3. Title of an article and all parts of a question
- Similarity score based on Okapi/BM25 (equation 1)
  4. An article and question by using elems
  5. An article and question by using words
  6. Condition parts of an article and a question by using elems
  7. Condition parts of an article and a question by using words
  8. Main parts of an article and a question by using elems
  9. Main parts of an article and a question by using words
- Similarity score based on word overlap (equation 2: $\beta = 1$)
  10. An article and question by using elems
  11. An article and question by using words

[5] https://code.google.com/archive/p/word2vec/
[6] http://www.courts.go.jp

12. Condition parts of an article and a question by using elems
13. Condition parts of an article and a question by using words
14. Main parts of an article and a question by using elems
15. Main parts of an article and a question by using words

Another issue is related to the training data used for the SVM-Rank. In the training data, there are several questions and related article(s) pair that is difficult to retrieve by using features defined above. For example, it is difficult to retrieve related articles of the question that requires high level semantic matching; for example, H23-2-O requires semantic matching such as “ ”(manifestation of intention)-“”(contract offer) and “”(effective)-“”(revoke). Another example is second and third related articles to the question. For example, for the question H23-1-3 has two related articles (96 and 709). Article 96 shares many keywords in the question (“(intention)”, “(manifestation)”, “(fraud)”, “(person)”), but article 709 shares only one keyword (“(intentionally)”). On the contrary there are several non-related articles that shares those keywords (e.g., article 101 shares many keywords in the question (“ (intention)”, “(manifestation)”, “(fraud)”, “ (recieve)”, “(case)”)). In such a case, training data about article 709 should be ranked higher than article 101 is harmful for the training process. Therefore, we conduct first training process to identify such outliers by using 10 cross fold validation (a set of questions for one year corresponds to one fold). In this experiment, questions whose rank of related articles are larger than 50 are excluded from the training data. In addition, articles whose rank are larger than 50 are also treated as non-relevant article for the training process. In this case, 17 questions are removed from final training data and 20 articles are marked as non-relevant articles in the training data.

In addition, we also make following 5 SVM models that use restricted numbers of features for generating varieties of answers for ensemble.

- Model without alignment(-A) (4-15)
- Model without Okapi/BM25(-O) (1-3,10-15)
- Model without overlap(-o) (1-9)
- Model without words(-w) (1-4,6,8,10,12,14)
- Model without elems(-e) (1-3,5,7,9,11,13,15)

3.5 Generating Final Answers

Final answers are generated by aggregating the results from 6 SVM models (1 full features SVM model and 5 restricted features SVM models). Final score of an article for a question is calculated as a summation of score from 6 SVM models. Top 1 ranked articles are selected as candidate answer for the question. In addition to the top 1 ranked articles, 2nd rank article that was ranked 1st at least one or more SVM models are also treated as candidates.

Then we check the possibility to add a mutatis mutandis article. Instead of making virtual article that combines description about the case description in the mutatis mutandis articles and referred article, the system checks the possibility when such referred articles are selected as candidates. Since most of the mutatis mutandis article have such description for refering to the related articles as “AXY(“A” shall apply mutatis mutandis to the case from Article X to Article Y)”, existence of topic keywords “A” in the question is important. For example, when the result of the question is article X and have keyword “A”, the sytem returns this mutatis

mutandis article in addition to X. In order to conduct this process, we make a mutatis mutandis article database that have information about a refered article, a mutatis mutandis article and topic keywords extracted by the pattern "A"(about A).

4 Experiment

4.1 Evaluation of the Proposed System

In order to evaluate the effectiveness of the system, we conduct experiments by using COLIEE 2017 training data and final test data.

Table 1 shows retrieval performance (F-measure) of 6 SVM models. Numbers with bold font represents best result for each year. From this table, we confirm that there is no model that outperforms other models consistently. It means results of each SVM model has unique findings compared to the best performance system ("-A" for total) and may contribute to generate better aggregated results by ensemble learning.

	ALL	-A	-O	-P	-w	-e
H18	**0.42**	**0.42**	0.40	0.35	0.37	0.37
H19	0.51	0.51	0.47	0.49	0.47	**0.56**
H20	0.58	0.60	0.60	0.60	**0.66**	0.58
H21	0.58	**0.65**	0.55	0.55	0.52	0.55
H22	0.67	0.69	0.61	0.63	0.57	**0.71**
H23	0.57	**0.60**	**0.60**	0.55	0.54	0.58
H24	0.51	**0.54**	0.52	0.49	0.46	0.51
H25	0.63	0.64	**0.74**	0.58	0.55	0.64
H26	0.55	**0.56**	0.54	0.56	0.52	**0.56**
H27	**0.52**	0.51	0.50	0.47	0.51	0.48
H28	**0.54**	0.50	0.51	0.47	0.43	**0.54**
Total	0.55	**0.56**	0.55	0.52	0.51	0.55

Table 1: Retrieval performance of 6 SVM models (F-measure)

The t test at a significance level of 0.05 for two-sided tests were used to compare the performance between 1 full features SVM and other restricted SVM. In this case, model without overlap(-o) ($p = 0.0026$) and model without words(-w) ($p = 0.010$) are significantly worse than full features SVM.

Table 2 shows retrieval performance (Recall and F-measure) of ensemble results. Simple uses only top 1 ranked article (and mutatis mutandis article) for the answers. Rank2(2) adds top 2 ranked article that was ranked 1st at least two or more SVM models. Rank2(1) adds top 2 ranked article that was ranked 1st at least one or more SVM models. Since number of returned answers are increased for Rank2(2) and Rank2(1), recall of the results are also increased and Rank2(1) has highest recall in all datasets. However, since precision is also decreased, simple case has highest average of F-measure in this experiment.

The t test at a significance level of 0.05 for two-sided tests were used to compare the performance between 1 full features SVM and these ensemble results. In this case, simple ($p = 0.03$) is significant better than full features SVM.

From this results, we confirm the ensemble method slightly improves the performance by aggregating the results. However, further analysis is necessary to evaluate the characteristics

	simple		rank2 (1)		rank2 (2)	
	Rec	F	Rec	F	Rec	F
H18	0.36	0.42	0.38	**0.42**	**0.40**	0.40
H19	0.47	0.51	**0.49**	0.50	**0.49**	0.47
H20	0.58	0.66	0.64	**0.68**	**0.68**	0.65
H21	0.57	0.61	0.61	0.62	**0.64**	0.58
H22	0.67	0.69	0.67	0.65	**0.71**	0.64
H23	0.56	0.60	0.62	**0.62**	**0.63**	0.58
H24	0.48	**0.54**	0.51	0.53	**0.54**	0.51
H25	0.59	0.64	0.61	0.63	**0.67**	0.63
H26	0.48	0.56	0.50	0.56	**0.55**	**0.57**
H27	0.43	0.50	0.47	0.51	**0.53**	**0.53**
H28	0.46	**0.54**	0.48	0.52	**0.53**	0.53
Total	0.50	**0.57**	0.53	0.56	**0.57**	0.55

Table 2: Retrieval performance of ensemble results (Recall and F-measure)

of the method.

4.2 Discussion

One of the reason why ensemble system works better than simple system is variation of the question types. From the viewpoint of retrieval performance, there are three groups in the datasets. One is easy question that shares many terms with related articles and most of the system can easily find the related ones. Second is hard question that requires external resources to identify similarity between questions and related articles. Outliers discussed in 3.4 are examples of this group. Questions in the last group may have varieties of clues to identify relationship between a question and a related article. For those questions, output of the systems vary based on the features used in the systems. Ensemble method can summarize the output of those system and can generate more stable results compared to the other SVM models.

However, overall system performance for the dataset is highly affected by the mixture ratio of these questions types in the datasets. For the further analysis of the system characteristics, it is better to conduct analysis by using topics that have poor performance [Voo05].

Based on the organizers summary, retrieval performance of our system is not so good compared to the best performance of COLIEE 2017. However, as we discussed in Section 3.3, Japanese dataset is not so uniform, compared to the English dataset. In order to clarify the difference between these two datasets, we conduct simple retrieval experiments by using state-of-the-art IR system Indri [SMTC05][7]. We use Indri for English dataset without any tuning and evaluate the system by using top ranked articles as answers to the question. We also construct database for Japanese by splitting Japanese sentence into morpheme sequence and evaluate the system. Table 3 shows F-measure of English and Japanese dataset. From this result, English dataset is relatively easier than Japanese one. Especially for the test dataset of this year (H28), English one is comparatively easier than Japanese one. This may reflect that description style of English questions are more similar to the articles than that of Japanese questions. It is necessary to take into account this factor when we compare the results obtained from the Japanese dataset and the English one.

[7] https://www.lemurproject.org/indri/

	English	Japanese
H18	0.30	0.19
H19	0.49	0.49
H20	0.53	0.48
H21	0.55	0.55
H22	0.55	0.71
H23	0.54	0.58
H24	0.49	0.33
H25	0.59	0.47
H26	0.50	0.46
H27	0.59	0.44
H28	0.59	0.40
Total	0.53	0.46

Table 3: Retrieval performance of Indri for English and Japanese datasets (F-measure)

5 Conclusion and Future Works

In this paper, we propose a new IR system for COLIEE 2017. This system uses phrase matching and ensemble approach to retrieve relevant articles for the question. We confirm that ensemble approach improves the retrieval performance. However, improvement by using ensemble approach is not consistent and there are several cases that results obtained by ensemble approach is not better than single SVM models. Therefore, it is better to have a framework to analyze question to select appropriate parameter settings based on the understanding of question types. In addition, we also confirm that English dataset for this year is little bit easier than other cases. It is necessary to take into account such effects when they compare the retrieval performance based on English and Japanese datasets.

References

[AMHKV99] Helena Ahonen-Myka, Oskari Heinonen, Mika Klemettinen, and A Inkeri Verkamo. Finding co-occurring text phrases by combining sequence and frequent set discovery. In *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, pages 1–9. Citeseer, 1999.

[BIF+09] Francis Bond, Hitoshi Isahara, Sanae Fujita, Kiyotaka Uchimoto, Takayuki Kuribayashi, and Kyoko Kanzaki. Enhancing the japanese wordnet. In *Proceedings of the 7th Workshop on Asian Language Resources*, ALR7, pages 1–8, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Joa06] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[KGKS16] Mi-Young Kim, Randy Goebel, Yoshinobu Kano, and Ken Satoh. Coliee-2016: Evaluation of the competition on legal information extraction and entailment. In *The Proceedings of the 10th International Workshop on Juris-Informatics (JURISIN2016)*, 2016. Paper 11.

[KHJ+16] Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kihyun Hong, and Young-Yik Rhim. An ensemble based legal information retrieval and entailment system. In *The Proceedings of the 10th International Workshop on Juris-Informatics (JURISIN2016)*, 2016. Paper 11.

[KM02] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[NW70] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[OY16] Daiki Onodera and Masaharu Yoshioka. Civil code article information retrieval system based on legal terminology and civil code article structure. In *The Proceedings of the 10th International Workshop on Juris-Informatics (JURISIN2016)*, 2016. Paper 19.

[RW00] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In *Proceedings of TREC-8*, pages 151–162, 2000.

[SMTC05] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, pages 2–6, 2005.

[Voo05] Ellen M. Voorhees. The trec robust retrieval track. *SIGIR Forum*, 39(1):11–20, June 2005.

[Yos10] Masaharu Yoshioka. On a combination of probabilistic and boolean ir models for question answering. In Pu-Jen Cheng, Min-Yen Kan, Wai Lam, and Preslav Nakov, editors, *Information Retrieval Technology 6th Asia Infomation Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 2010 Proceedings*, pages 588–598. Springer-Verlag GmbH, 2010. LNCS6458.

EPiC Series in Computing

Volume 47, 2017, Pages 23–30

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Multiple Agent Based Entailment System (MABES) for RTE

Byungtaek Jung, Chiseung Soh, Kihyun Hong,
Seungtak Lim and Young-Yik Rhim
Intellicon Meta Lab, Seoul, Korea
jungbt@intellicon.co.kr, soh@intellicon.co.kr,
kihyun.hong@intellicon.co.kr,
stlim@intellicon.co.kr and ceo@intellicon.co.kr

Abstract

Despite growing needs of the legal artificial intelligence (AI), its development is slower than other AI domains because legal expertise is essentially required to develop legal AI systems. Legal knowledge representation on legal expertise needs to be considered to implement legal reasoning AI systems. In this paper, we present a legal reasoning methodology, which utilizes multiple expert knowledge based agents. These agents are designed to solve recognizing textual entailment (RTE) problems with syntactic and interpretative knowledge. The validity of the proposed method is provided through experiments with the COLIEE 2017 data.

1 Introduction

As tremendous amounts of documents have been produced in a digital form over the past decades, it has been crucial to gather or retrieve the necessary information. In the legal domain, numerous digitized precedents and reports are also being produced every day. However, without legal expert knowledge, it is difficult to refine the retrieved information into easy-to-use information. Many researches have been conducted on the use of digitized legal information: Crime prediction (NathS., 2006) is performed through big data based pattern analysis. For e-Discovery (MarcusR., 2008) a lot of software applications have widely been commercialized. Many legal websites provide judicial precedent searches.

There are still many issues to be addressed in the legal AI area, especially in legal information retrieval (IR) and RTE application fields. Many legal AI studies in the legal IR field have been conducted and embedded in various systems. The applications of legal information processing such as e-Discovery and legal question answering (QA) are based on legal document search and legal reasoning.

COLIEE (Competition on Legal Information Extraction/Entailment) has been held in order to deal with legal IR and RTE problems. In the IR phase, a legal information processing system needs to find

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 23–30

relevant articles for a given query. In the RTE part, the system needs to determine whether the article found in the IR phase entails the query.

In this paper, we present a multiple agent based entailment system (MABES). Legal knowledge is analyzed and structuralized to describe an interpretative legal knowledge base (LKB). The LKB based agent is constructed by ontology with triple structure. However, it is hard to create all the necessary rules for complete legal reasoning in the LKB system. We also apply complementary methods such as syntactic knowledge based agent to overcome the deficiency of the LKB based agent.

This paper is organized as follows: Section 2 reviews related works on legal information inference. Section 3 explains the multiple agent based entailment system. Section 4 describes the experimental results of the proposed method. Section 5 provides the conclusion of this paper.

2 Related Works

To tackle the legal RTE problem, some approaches were proposed. M.Y. Kim *et al.* (Kim.M.Y., XuY., GoebelR., SatohK., 2014) constructed a knowledge base (KB) for negative and antonym dictionary, and created a rule for analyzing premise and conclusion clauses of a query. They also proposed an unsupervised learning to decide entailment relationship between a query and an article with linguistic information. K. Kim *et al.* (KimK., HeoS., JungS., HongK., RhimY., 2016) proposed ensemble based classification methods for the legal RTE problem. In their approach, the entailment problem was described as a binary classification problem. A Siamese structure based convolutional neural network (CNN) and various classifiers such as a decision tree classifier were utilized to construct ensemble classifiers. They used several document similarity measures as input features of the classifiers. In order to solve the entailment problem, applying simple rules as in (Kim.M.Y., XuY., GoebelR., SatohK., 2014) is logically valid for relatively easy problems where entailment relationship is mainly dependent upon its positive and negative polarity. However, it is still difficult to solve the problem intertwined with a lot of knowledge using the simple rules. In similarity based RTE approaches, it is not needed to design entailment related rules or KB. However, it has a drawback that it might not explain the reasoning logics for the results, since it does not have explicit knowledge based rules.

3 Multiple Agent Based Entailment System

In this paper, the main target problem is the legal RTE problem to determine whether a given legal sentence entails a query sentence. We use a multi-agent based approach to solve the RTE problem. Two agents, which are the LKB based agent and the syntactic knowledge based agent, are applied: The LKB based agent determines entailment relation between a query and legal articles with ontology based on legal entailment processing knowledge. On the other hand, the syntactic knowledge (SK) based agent uses context structural information for the legal RTE.

Figure 1 shows the overall flow of MABES. First, the LKB based agent checks entailment relation. If the LKB based agent detects entailment relation, the entailment process is terminated with the result of the LKB agent. However, if the LKB based agent fails to decide the entailment relation of the input sentences, the entailment relation is determined by the SK based agent. As described in Figure 1, MABES is a cascaded RTE process using multiple agents.

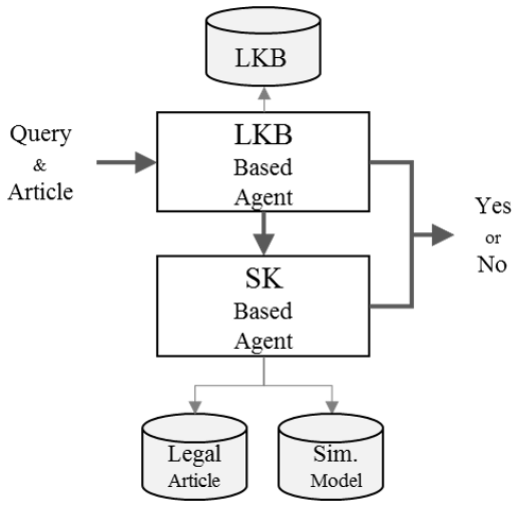

Figure 1: Overall flow diagram of MABES

4 LKB Based Approach

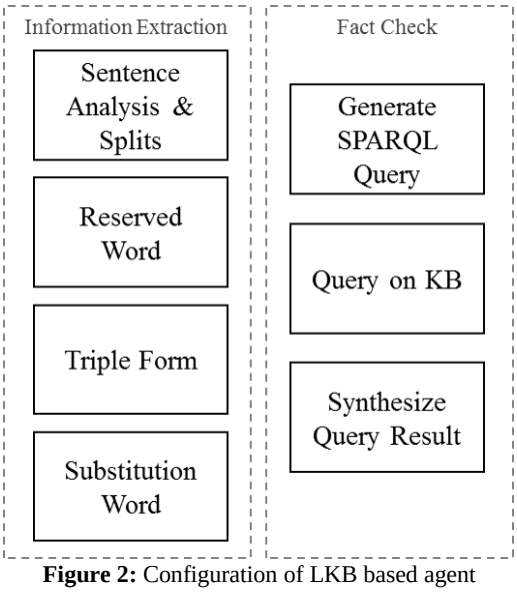

Figure 2: Configuration of LKB based agent

The LKB based agent consists of two parts, which are the information extraction and the fact check parts. First, the LKB based agent extracts the key entities from input sentences, which are triple data for reasoning. It then investigates whether the extracted triple data is described in the LKB. Figure 2 describes the overall functional structure of the LKB based agent.

4.1 Information Extraction

In the information extraction part, the input data is sequentially processed in the order of query analysis and split, reserved word process, change to triple structure, and substitution word process as shown in Figure 3. If a query has a conjunction, the sentence is split by the conjunction to produce two or more sentences. In each split sentence, main terms are extracted based on the dictionary of reserved words. A subject, an object, and a predicate are extracted from each sentence in order to construct a triple structure. If there is no object, it is processed as 'none'. The predicate has 'True' or 'False' property in the form of 'Data Property' of OWL (HitzlerP., KrötzschM., ParsiaB., Patel-SchneidePF, S.Rudolph, 2009). If the predicate includes a negative expression, 'NEG' tag is added. Finally, the triple data are substituted for proper legal terms of the LKB through the substitution word process.

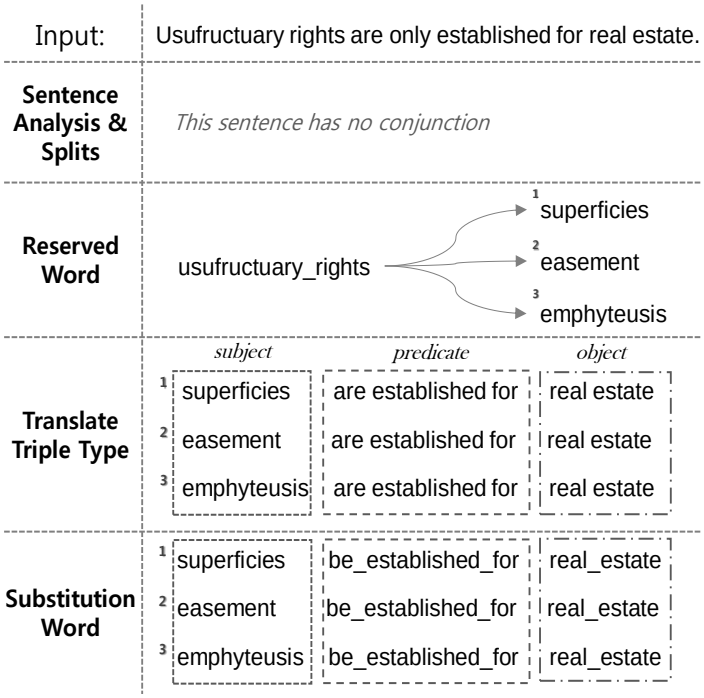

Figure 3: Data processing on the information extraction

4.2 Fact Check

In the fact check part, legal fact is analyzed using the extracted triple data from the information part. Using the ASK query format of SPARQL (HarrisS., A.Seaborne, E.Prudʾhommeaux, 2009) it is checked whether the triple data exist as a fact in the LKB. The answer of an ASK query is 'True' or 'False'. If it is 'False', it represents 'negation as failure' meaning that the triple data are actually 'False' or do not exist in the LKB. As a result, it is processed to 'Unknown'. However, if the answer is 'True', the input data has entailment relation if the triple data are correctly extracted. Each triple data in the input sentence is checked with the LKB. If results include 'False', the total result becomes 'Unknown'. Only if all results include 'True', the final result becomes 'True'.

5 Syntactic Knowledge Based Agent

Syntactic analysis module is divided into two parts, which are negation detection and similarity analysis. As the basic module of syntactic analysis, some negation detection rules are used. Negation can change the entailment relation between two documents most dramatically. Therefore, it is effective to apply negation analysis to solve RTE problems. However, if there is a big structural difference between a query and an article, negation rules may not provide reasonable results in RTE tasks; structural difference means the difference of sentence length, word usage, or parts of speech between two documents. Therefore, additional criteria are needed to determine to apply negation rules. In this paper, we use the similarity based RTE method as a complementary one.

5.1 Negation Detection

In a RTE problem, one negation expression can change the state of 'True' or 'False'. A simple pseudo code in Figure 4 describes the entailment state transition.

```
if entail(query, article) = True then
        entail(query, article) with a negation = False
else if entail(query, article) = False then
        entail(query, article) with a negation = True
end if
```

Figure 4: Equation rule

If the entailment relation of a query and an article is 'True', the addition of a negation expression can trigger the entailment state to be 'False'. If the entailment relation of a set is 'False', such an addition can lead the state to be 'True'.

Type 1	**negate verbs with not or n't -** cannot, can't, do not, …
Type 2	**using negative words -** cancel, terminate, extinguish, illegal, …
Type 3	**negate noun -** no restriction, no contract, …
Type 4	**conditional negation -** unless, except, …
Type 5	**comparative negation, kind of antonym -** adult and child, employee and employer, **…**

Table 1: Negation types

There are many ways to negate a sentence as shown in Table 1. Type 1 through Type 3 in Table 1 are comparatively easy to analyze. On the other hand, Type 4 and 5 need to be scrutinized to ascertain whether they are negative words or not.

5.2 Similarity Analysis

To calculate similarity, queries and articles should be vectorized into context vectors. The contextual vectors are usually constructed by TF-IDF (SaltonG. & McGillM., 1986), LSI (DumaisS., 2004), Word2Vec (MikolovT., ChenK., CorradoG., DeanJ., 2013), etc. Although the similarity scores can be diverse following vector space models, counts of the same or similar words are highly correlated with similarity score. In other words, higher similarity score in any vector model means small structural variance between a query and an article. Therefore, although similarity itself is not a conclusive solution

to the RTE problem, it can be an indicator of how we can deal with the RTE problem and it can be used to supplement other RTE approaches such as the negation rule based method.

6 Experiments and Results

First, we describe the test conditions of the LKB and SK based agents. Then, we show the test result of the proposed MABES, using the COLIEE 2017 data.

The LKB based agent checks whether the triple content exists in the LKB. In other words, the method is not dependent on some particular articles, but is dependent on the already established LKB. In this experiment, some of Articles 265 to 398 in Japanese Civil Code are encoded in the LKB.

In the SK based agent, we applied the negation Type 1 to Type 3 in Table 1. Note that the negation Type 4 and Type 5 in Table 1 were not used in this experiment.

The candidate models for the similarity analysis were TF-IDF, LSI, and Word2Vec and the similarity was measured by the cosine method.

6.1 Results

We used the 581 test queries in the COLIEE data set. 'True' data account for 51.6 % and 'False' data 48.4 %: 51.6 % is set as the baseline in this experiment.

First, we show the result of using the negation rules. The performance had 57.5 % of accuracy, which is 5.9 % higher than the baseline.

Correct	Incorrect	Accuracy
334	247	57.5

Table 2: Negation type result

Next, we examined vector space models. The average similarity scores of three candidate vector models are presented in Table 3. In Table 3, the similarity scores are normalized with 100. The LSI model showed that the average similarity score is 36.4 for entire data that have true entailment relation and 34.6 for false entailment relation. Except for the Word2Vec model, all models with true entailment relation had higher similarity. In particular, the TF-IDF model presented such difference clearly. Based on this test, we selected TF-IDF as our context vector model.

Additionally, similarity and model accuracy are shown in Figure 5. The TF-IDF model is showing the most noticeable trend line. It explains that the higher the similarity score, the higher the accuracy.

Model	Entail: True	Entail: False
LSI	36.4	34.6
TF-IDF	29.0	23.7
Word2Vec	45.2	46.1

Table 3: Similarity score by models

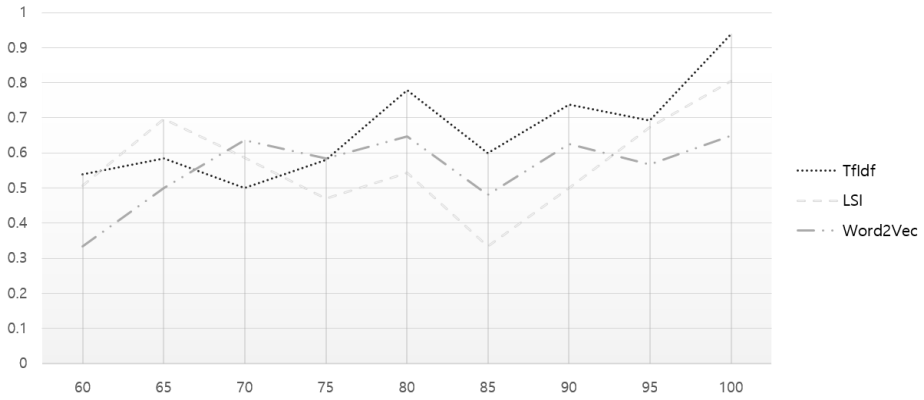

Figure 5: Model similarity and accuracy

In Table 4 combined results of the LKB and SK based agents is presented. MABES had about 8 % performance improvement compared to the baseline and 2 % higher accuracy than the negation rule based entailment.

Language	Correct	Incorrect	Accuracy
English	345	236	0.594
Japanese	325	256	0.560

Table 4: Results for training data

The final result submitted to the COLIEE 2017 competition is shown in Table 5.

Language	Correct	Incorrect	Accuracy
English	45	33	0.577
Japanese	42	36	0.538

Table 5: Final results

The experimental result using English showed better performance than the Japanese result. In English sentences it is easy to judge negative words, whereas in Japanese it is required to take into account a part-of-speech of words in order to find a negative word. Also, since Japanese language has a lot of obscure negative expressions, it is difficult to judge entailment relation through the negation rules.

7 Conclusion

In this paper, the multiple agent based entailment system was presented, which consists of the LKB and SK based agents. The LKB module solves the RTE problems with its legal knowledge base, which contains legal reasoning rules. Although the LKB module can perform complex legal reasoning, it has drawbacks to build the LKB. It requires knowledge of legal experts and time and efforts. In the syntactic analysis module, it was found that negation rules provided good results when queries and articles had high similarity scores. However, when document similarity score was low, the effectiveness of applying the negation rules was reduced.

For further improvement, we will expand the LKB with legal experts and will analyze syntactic structures that affect entailment in addition to negation rules for accurate legal RTEs.

References

Dumais, S. T. (2004). Latent semantic analysis. (pp. pp 188-230). Annual review of information science and technology, 38(1).

Harris, S., A., S., & E., P. (2009). Sparql 1.1 query language. W3C recommendation, 21(10).

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneide, P., & S., R. (2009). Owl 2 web ontology language primer. W3C recommendation 27.1.

Kim, K., Heo, S., Jung, S., Hong, K., & Rhim, Y. Y. (2016). An ensemble based legal information retrieval and entailment system. *In Proceed-ings of the Tenth International Workshop on Juris-informatics, JURISIN 2016* (pp. pp. 167–176). JSAI-isAI.

Kim., M., Xu, Y., Goebel, R., & Satoh, K. (2014). Answering Yes/No Questions in Legal Bar Exams. (pp. pp.199–213). JSAI-isAI 2013.

Marcus, R. (2008). E-discovery beyond the federal rules.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Nath, S. V. (2006). Crime pattern detection using data mining. *WIIAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on* (pp. pp 41-44). IEEE.

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

EPiC Series in Computing

Volume 47, 2017, Pages 31–42

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models

Nguyen Truong Son, Phan Viet Anh, and Nguyen Le Minh

Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan 923-1292
nguyen.son@jaist.ac.jp, anhpv@@jaist.ac.jp, nguyenml@jaist.ac.jp

Abstract

This paper presents an end-to-end question answering system for legal texts. This system includes two main phases. In the first phase, our system will retrieve articles from Japanese Civil Code that are relevant with the given question using the cosine distance after the given question and articles are converted into vectors using TF-IDF weighting scheme. Then, a ranking model can be applied to re-rank these retrieved articles using a learning to rank algorithm and annotated corpus. In the second phase, we adapted two deep learning models, which has been proposed for the Natural language inference task, to check the entailment relationship between a question and its related articles including a sentence encoding-based model and a decomposable attention model. Experimental results show that our approaches can be a promising approach for information extraction/entailment in legal texts.

Keywords: legal question answering, information retrieval, entailment extraction, deep learning

1 Introduction

Studying legal issues from the perspective of informatics is a topic that draws interests from researchers recent years. In this task, one of the important targets is the information extraction/reasoning from legal data including legal relation extraction, textual entailment, etc. Legal question answering is also an essential task in legal issues. Recent work proposed some of the first efforts for this task [8, 15, 6].

In this work, we build a two-phase system for the legal information extraction/entailment task. In the first phase, a list of relevant articles are retrieved by compute the cosine similarity between the TF-IDF vectors of the given question and articles. A ranking model can be applied to obtain the best results. The architecture of this phase is followed the architecture of our system last year [12] by adding some n-gram indexing models ($n = 2$) beside the uni-gram indexing model.

A question in the COLIEE task is a statement that we need to check whether or not it is entailed by some articles in legal documents. Therefore, the second phase will try to check entailment relationship between a question and its related articles thanks to a classifier. Because

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 31–42

we have an annotated data set, supervised methods are preferred approaches for this task by treating this task as a binary classification task.

Recently, deep learning models show it effectiveness for NLP tasks such as machine translation, sequence labeling, text generation as well as natural language inference task. In many tasks such as named entity recognition, deep learning models can produce the state-of-the-art without using any engineering features [9]. In the natural language inference tasks [2], which try to recognize the entailment relationship between a text and a hypothesis, deep learning models also produce better results in comparison with conventional machine learning algorithms [2]. For these reasons, our motivation is how to adapt these models for question answering task in legal texts. In this work, we investigate two deep learning models including the sentence encoding-based model proposed by [2], and the decomposable attention model proposed by [13].

Experimental results show that our approaches have some promising results. First, adding bi-gram and tri-gram indexing models shows a significant improvement. Second, deep learning models could be a good approaches for entailment recognition in legal texts because without engineering features, the result is competitive with conventional algorithms.

The remainder of this paper is organized as follows. Section 2 presents briefly some related works. The architecture of Information retrieval phase and entailment checking phase are presented in Section 3 and 4. The experiments are analyzed in Section 5. Conclusions and future works are shown in Section 6.

2 Related Work

A basis for a Yes/No Arabic Question Answering System was proposed in [1] using a textual entailment method. For the task of legal information retrieval/entailment task, [8] proposed a system for answering yes/no questions in legal bar exams.

The first shared task of the legal information retrieval/extraction (COLIEE2014) was reported in [15]. There are many methods proposed to solve the task. [17] focused on exploiting reference information to build a question answering system restricted to the legal domain. [7] used ranking SVM and syntactic/semantic similarity.

In the COLIEE 2015 competition, [6] proposed using a convolutional neural network in legal question answering. D. S. Carvalho et al. [3] used lexical and morphological characteristics to extract n-gram features from the query and articles; their own relevance score was then used for Phase 1. They used the query and relevant article are represented in vector space model for Phase 2. In order to classify the queries between Yes label and No label, they used AdaBoost (basic classifier: DecisionStump). Sushimita et al. [14] used voting of three information retrieval techniques: Hiemstra, BM25 and PL2F for the information retrieval task. In [4], a keyword weighting method was proposed and snippet scoring after diving data into snippets. Tran et al., 2015 [16] proposed using hidden markov model for the legal information retrieval task.

In the COLIEE 2016, [5] proposed ensemble similarity using a least square method and linear discriminant analysis with variety of features such as lexical similarity, syntactic similarity, and semantic similarity. Among their many approaches, LSM ensemble showed best performance. Besides, for the entailment recognition task, they applied majority voting scheme with three classifiers (decision tree, linear SVM, and convolutional neural network (CNN) classifiers) using many features such as word overlap, cosine similarity, substring similarity, the Lucene similarity score, 3 different role similarity scores, 6 WordNet similarity scores, negative term ratio, and length ratio. Their system showed the best performance in COLIEE 2016.

3 Phase 1: Retrieving relevant articles

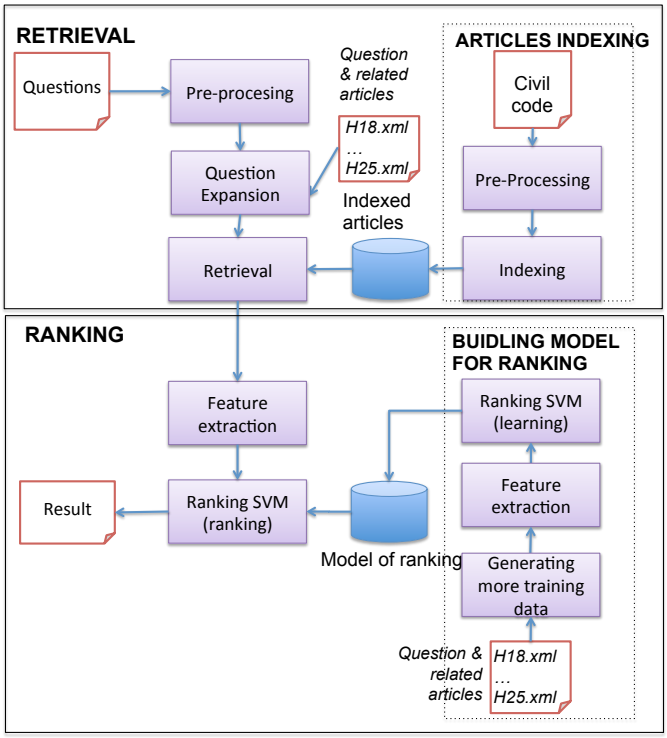

Figure 1: The architecture of information retrieval phase used in [12]

The architecture of the information retrieval system is described in Figure 1 which is followed by the architecture in [12]. It has two main steps including information retrieval and ranking. In the first step, for each given question, the system will retrieve top 100 relevant articles based on the cosine distance between the TF-IDF vectors of articles and the given question.

The ranking step will re-rank 100 relevant articles of the first step by using a ranking model which is trained by SVM rank algorithm. Finally, for each question we choose the most relevant article after the ranking step.

The details of these steps will be described in the remain of this section.

3.1 Articles Indexing

This component will convert all articles into TF-IDF vector representations, a popular method for representing documents in the information retrieval field. Moreover, to improve the performance of the system, some preprocessing steps can be applied such as stemming or removing stop words. Besides, to increase the important of long text matching, instead of using only uni-gram model, we add bi-gram and tri-gram indexing model for documents representation.

3.2 Query Expansion

The query expansion step is an option in our system. This step tries to add related terms into a given query to improve retrieve performance. Query expansion involves techniques that can find synonyms of words, and search for the synonyms as well. We used two methods for query expansion using Word2Vec and WordNet.

Query expansion using word2vec[10]: From questions and their relevant articles in training corpus, extract similar word pairs of word in question and articles base on cosine similarity of their word embedding representation. We select word-pairs that have the cosine similarly value ≥ 0.5 as the related words for query expansion. We expect this methods can retrieve articles that does not share words with the given query.

Query expansion using WordNet[11]: From questions, we expand the question by adding synonyms and hypernyms of words in that question by looking in WordNet dictionary. However, this way is not effective because each word may have it make the precision score reduce sharply.

3.3 Ranking

3.3.1 Prepare training data

Given a question, learning to rank algorithms require relevant articles and non-relevant articles for that question as the training data to build the model for ranking. However, the COLIEE training data only contains questions and relevant articles, so it is not enough for learning to rank algorithm work effective. To get more training data for learning to rank algorithm, we generate non-relevant articles of a question by selecting articles that have the low cosine similarity value based on TF-IDF vectors.

3.3.2 Features Extraction

We used the following feature list to train the ranking model.

- Cosine similarity between question and article of TF-IDF unigram vectors
- Cosine similarity between questions and articles of TF-IDF bigram vectors
- Cosine similarity between question and article of TF-IDF trigram vectors
- Cosine similarity between nouns in questions and articles
- Cosine similarity between verbs in questions and articles
- Distance feature between questions and article: Euclidean, Manhattan, Levenshtein, Jaccard

3.3.3 Training the ranking model

We train the ranking model using SVMRank tool [1] based on the above feature set. Many previous works showed that SVM Rank are an effective tools for training ranking models. [7] also used SVM-Rank for ranking the result after the retrieval phase in the COLIEE task.

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

4 Recognizing Entailment between articles and queries

Given a pair of text including a question **b** and a text **a** in which **a** is the content of some articles that relevant to question **b** which has been retrieved from the phase 1. This task tries to recognize whether or not **b** is entailed by **a** based on the meaning of the given question and related articles. By considering the content of related articles as a **text**, and the content of the given question is **hypothesis** we can solve this task using some approaches for the natural language inference task that have been developed in recent years.

We adapt two types of deep learning models to recognize the entailment relationship between law articles and questions including: (1) a sentence encoding based model that uses recurrent neural networks to encode the content of an article and the given question; and (2) a decomposable attention model which try to solve the textual entailment task by computing the alignment between words and phrases of the given question and related articles then use these alignments as features to check the entailment relationship.

Training data set: each training example in the training set is a triple of $\{\mathbf{a}^{(n)}, \mathbf{b}^{(n)}, \mathbf{y}^{(n)}\}$ where $\mathbf{a}^{(n)}$, $\mathbf{b}^{(n)}$, and $\mathbf{y}^{(n)}$ is content of related articles, a question and the label of the this training example ($\mathbf{y} \in \{Y, N\}$).

4.1 A Sentence encoding model based on Recurrent neural networks

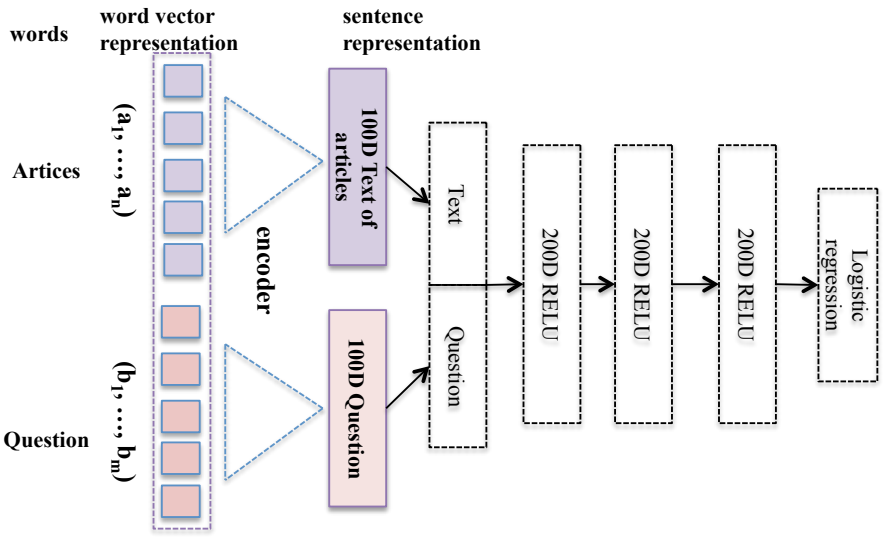

Figure 2: The sentence encoding model for recognizing the entailment between a question and the relevant articles

Figure 2 shows the architecture of the sentence encoding-based model. Let $\mathbf{a} = (a_1, a_2, ..., a_{l_a})$ and $\mathbf{b} = (b_1, b_2, ..., b_{l_b})$ be the two input texts (related articles and question) respectively. Each a_i, b_j is a word embedding vector of dimension d. l_a and l_b is the length of **a** and **b**. Then these texts are encoded into vectors (the translation step) using a sentence encoder model. After that, two these vectors are concatenated into only one vector and then this vector is fed through three layers with RELU activation. Finally, the logistic regression

layer will classify the input sentence pair to predict whether or not the given article entails the question.

Below are some simple sentence encoder models which have used in [2]:

- Summation of embedding words: This method will sum the embeddings of the words in each text to obtain the vector representation.
- Simple sequence embedding models: using simple RNN or Long-short term memory to encode the input sentence.

4.2 Decomposable attention models

Figure 3 shows the architecture of the decomposable attention model for natural language inference proposed by [13] which is used to recognize the relationship between two input texts $\mathbf{a} = (a_1, a_2, ..., a_{l_a})$ and $\mathbf{b} = (b_1, b_2, ..., b_{l_b})$ by decomposing the problem into sub-problems. This model is composed from three main components: **attend**, **compare** and **aggregate**.

Attend This component will soft-align the elements of $\mathbf{a}$ and $\mathbf{b}$ using the neural attention technique. For each word a_i in $\mathbf{a}$, this step will find a sub-phrase β_i in $\mathbf{b}$ that soft-aligned to a_i and vice versa for α_j. The values of β_i and α_j are computed by equation 1 and 2.

$$\beta_i := \sum_{j=1}^{l_b} \frac{exp(e_{ij}))}{\sum_{k=1}^{l_b} exp(e_{ik})} b_j \tag{1}$$

$$\alpha_j := \sum_{i=1}^{l_a} \frac{exp(e_{ij}))}{\sum_{k=1}^{l_a} exp(e_{ik})} a_i \tag{2}$$

where e_{ij} are attention weights between words in $\mathbf{a}$ and $\mathbf{b}$ which is computed using a feed forward neural network F:

$$e_{ij} = F(a_i)^T F(b_j) \tag{3}$$

The obtained aligned phrases $\{(a_i, \beta_i)\}_{i=1}^{l_a}$ and $\{(b_j, \alpha_j)\}_{j=1}^{l_b}$ allow the model to decompose the problem into the comparison of aligned sub-phrases.

Compare This step will compare aligned phrases obtained from the previous step using a function G.

$$\begin{aligned} \mathbf{v}_{1,i} &:= G([a_i, \beta_i]) \quad \forall i \in [1, ..., l_a] \\ \mathbf{v}_{2,j} &:= G([b_j, \alpha_j]) \quad \forall j \in [1, ..., l_b] \end{aligned} \tag{4}$$

where the brackets $[\bullet, \bullet]$ denote the concatenation between two vectors. G is again a feed forward neural network with RELU activations.

Aggregate Two sets of comparison vectors $\{\mathbf{v}_{1,i}\}_{i=1}^{l_a}$ and $\{\mathbf{v}_{2,j}\}_{j=1}^{l_b}$ are aggregated into two vector $\mathbf{v}_1$ and $\mathbf{v}_2$ using summation:

$$\mathbf{v}_1 = \sum_{i=1}^{l_a} \mathbf{v}_{1,i} \qquad \mathbf{v}_2 = \sum_{j=1}^{l_b} \mathbf{v}_{1,i} \tag{5}$$

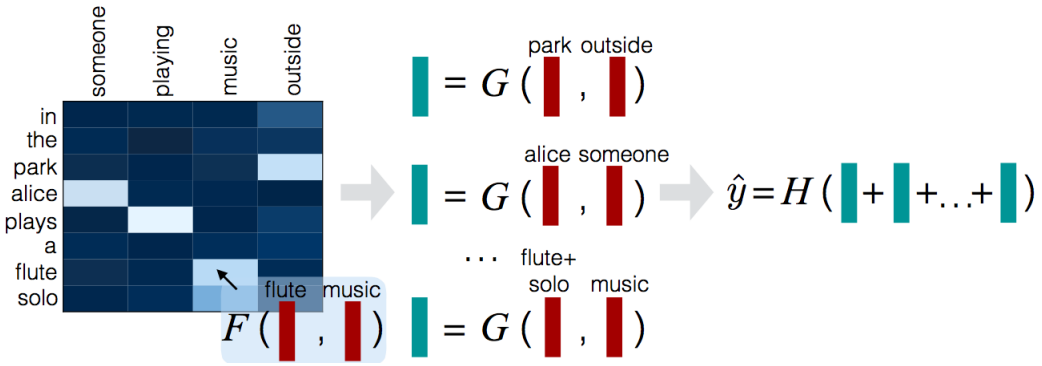

Figure 3: The decomposable attention model for recognizing textual entailment between two sentences including 3 steps: *Attend* (left), *Compare* (center) and *Aggregate* (right) [13]

Finally, a classifier H is used to predict the scores for each class. H is a feed forward network followed by a linear layer:

$$\hat{y} = H([\mathbf{v}_1, \mathbf{v}_2]) \tag{6}$$

where $\hat{y} \in \mathbb{R}^C$ is a vector that represents the scores of each classes (e.g. Yes/No). Then, the predicted class is computed by $\mathbf{argmax}_i \hat{y}_i$.

For training, the model is trained using dropout regularization with the cross-entropy loss function. During the training process, parameters of F, G, H is updated to minimize the loss function.

5 Experiments

5.1 Experimental results of phase 1

For phase one, we analysed aspects of the information retrieval task in our system including: stemming, query expansion, n-gram indexing, removing stop-words, and ranking. We conducted experiments and compared different configurations to find the configuration that produces the best performance. Table 1 shows the results of the information retrieval phase.

The results on the H18-H25 shows the contribution of the stemming and removing stop words step. When we stem or remove stop words, the performance of the system is usually better. However, these steps have negative impact on the data set H26 and H28.

Adding bi-gram indexing and tri-gram indexing models also improves the performance of our system. On all three data sets, top 3 best results are always using bi-gram or tri-gram indexing model. Experimental results in Table 2 also show that adding 2-gram and 3-gram have an important contribution for the retrieval task. For example, the retrieval performance improves 1.93%, 4.42%, 7.44% on H18-H25, H26, H28 data sets if we use the 2-gram indexing model.

However, when we use the n-gram indexing model with $n > 3$, the results do not improve but it takes more time for retrieving as well as for indexing.

Table 1 also shows that our query expansion approach does not always improve the results. Using word embedding similarity can find useful terms to the given question for but it may add many unrelated terms.

#	QUERY-EXPANSION	N-GRAM	REMOVE-STOPWORD	STEM	H18-H25	H26	H28
1		1gram			0.5096	0.5752	0.5319
2		1gram		x	0.5203	0.5487	0.5000
3		1gram	x		0.5075	0.5221	0.5532
4		1gram	x	x	0.5139	0.5133	0.4894
5		2gram			0.4989	0.5398	0.5426
6		2gram		x	0.5139	0.5487	0.5319
7		2gram	x		0.5096	0.5487	0.5638
8		2gram	x	x	0.5332	0.5575	0.5638
9		3gram			0.4968	0.5575	0.5532
10		3gram		x	0.5139	0.5752	0.5426
11		3gram	x		0.5161	0.5841	0.5638
12		3gram	x	x	0.5289	0.5664	**0.6277**$^{\textbf{(H28a)}}$
13		4gram			0.4946	**0.6018**	**0.5638**$^{\textbf{(H28b)}}$
14		4gram		x	0.5032	0.5929	0.5319
15		4gram	x		0.5139	0.5664	0.5638
16		4gram	x	x	0.5246	0.5664	0.5957
17	x	1gram			0.5203	0.5752	0.5213
18	x	1gram		x	0.5118	0.5310	0.5106
19	x	1gram	x		0.5139	0.5133	0.5426
20	x	1gram	x	x	0.5032	0.4956	**0.5213**$^{\textbf{(H28c)}}$
21	x	2gram			0.5075	0.5487	0.5745
22	x	2gram		x	0.5310	0.5487	0.5319
23	x	2gram	x		0.5246	0.5487	0.5532
24	x	2gram	x	x	**0.5439**	0.5310	**0.5851**$^{\textbf{(H28d)}}$
25	x	3gram			0.5032	0.5664	0.5532
26	x	3gram		x	0.5203	0.5752	0.5532
27	x	3gram	x		0.5246	0.5664	0.5532
28	x	3gram	x	x	0.5353	0.5664	0.5957
29	x	4gram			0.5011	0.5841	0.5638
30	x	4gram		x	0.5075	0.5841	0.5319
31	x	4gram	x		0.5268	0.5664	0.5426
32	x	4gram	x	x	0.5268	0.5575	0.5957

Table 1: Experimental results ($F_{\beta=1}$ score) of phase 1 - Information Retrieval without applying re-ranking models. H28b, H28d is the result on the test set when the performance is the best on the development set (H26) and training set respectively; H28a indicates the performance is the best on the test set but training set and development set; H28c indicates the result which we have submitted for the final official evaluation.

Table 3 shows the results after using the ranking model which is trained on the H18-H25 data set. Firstly, We retrieve relevant articles for each configuration then apply the ranking model to re-rank the retrieve articles. Experimental results show that the ranking model produces an intermediate result in comparison with the highest and the lowest results. For example, in the H26 data set, the $F_{\beta=1}$ score range is from 0.4956 to 0.6018 depend on the chosen configuration. However, after applying the ranking model, the results are 0.5221 of $F_{\beta=1}$ for all configurations.

#	N-GRAM	H18-H25	H26	H28
4	1 gram	0.5139	0.5133	0.4894
8	2 gram	0.5332	0.5575	0.5638
12	3 gram	0.5289	0.5664	0.6277
16	4 gram	0.5246	0.5664	0.5957

Table 2: Comparison between difference n-gram indexing models (all other configurations are the same: Query Expansion:No, Remove Stop words: Yes, Stemming: Yes)

This shows that the ranking model does not produce the best result.

Question set	Before ranking	After ranking
H26	0.4956∼0.6018	0.5221
H28	0.4894∼0.6277	0.5106

Table 3: Results ($F_{\beta=1}$ score) after using SVM rank model

5.2 Experimental results of phase 2

We use two available implementations [2] [3] for both two models with some adaptation for processing the input data. Both of the two tools are implemented in python language using Tensor Flow and Theano library.

Experimental settings:

- Configurations for the sentence encoding model: We experimented with two methods for sentence encoding including Word-summation and BI-LSTM. Other parameters are fixed such as word embedding size = 100, hidden layer size = 100, batch size = 8, drop-out rate = 0.2. This model is trained with training with RMSP optimizer and the training process will be stopped if the loss on the development set does not reduce after 50 epochs.
- Decomposable attention model: word embedding size=100, each feed forward network has 2 layers with hidden layer size = 200 , batch size = 8. This model is trained with Adagrad optimizer and learning rate = 0.05.

Pre-trained word embedding vectors for legal domain To train word embedding vectors for legal domain, we make a legal corpus from legal documents that crawled from website of Ministry of Justice, Japan [4]. Then, word embedding vectors are trained by using word2vec tool [10]. These embedding vectors are used for both two deep learning models in this task.

[2] https://github.com/Smerity/keras_snli
[3] https://github.com/erickrf/multiffn-nli
[4] http://www.japaneselawtranslation.go.jp: This website contain the English translation of Japanese legal documents

Results Table 4 and 5 show the results of the entailment classification task. The decomposable attention model outperforms the sentence encoding based model. Table 5 shows that the decomposable attention model usually produces the good result after 100 epoch training.

In the sentence encoding based model, using a complex network to encode input sentences does not improve the performance. For example, in 4 of 5 cases in Table 4, the BI-LSTM encoder shows lower results than word-summation encoder.

Sometimes, the model achieves the best results on the development set but it shows the bad results on the test set. It makes the parameter tuning process more difficult. We think that the size of training data and the difference between data sets are the reasons for these unstable results.

Sentence encoding method	Development set H26	H28a	H28b	H28c	H28d
Word-summation	0.6000	0.4872	0.4615	**0.5128** ♣	0.4872
BI-LSTM	0.5895	0.4359	0.4872	0.4359	0.4359

Table 4: Experimental results (accuracy) using sentence encoding-based models. H28a, H28b, H28c, H28d are obtained from phase 1 with configuration 12, 13, 20, 24 respectively. (♣) is the results we have summited for official evaluation

Epoch	Development set H26	H28a	H28b	H28c	H28d
10	0.4737	0.4615	0.4615	0.4615	0.4615
20	0.4211	0.5256	0.5256	0.4872	0.5000
30	0.4632	0.4487	0.4872	0.4359	0.4487
40	0.5895	0.4615	0.4487	0.4872	0.4872
50	0.4737	0.4359	0.4359	0.4359	0.4359
60	0.4842	0.4744	0.4615	0.4744	0.4872
70	0.5895	0.5128	0.5385	0.5513	0.5385
80	0.6000	0.5256	0.5000	0.5641	0.4744
90	0.5895	0.5641	0.5256	$0.5897^{(*)}$	0.5385
100	**0.6211**	**0.5769**	**0.5256**	**0.4872**♣	**0.5128**
110	0.4526	0.5641	$0.5513^{(*)}$	0.5513	0.5385
120	0.5053	$0.5897^{(*)}$	0.5256	0.5385	$0.5513^{(*)}$
130	0.6000	0.4231	0.4615	0.4359	0.4231
140	0.4632	0.4872	0.5128	0.5769	0.5256
150	0.4316	0.5385	0.5385	0.4872	0.5385

Table 5: Accuracy on the development set and test set of the decomposable attention model during the training process. The **bold line** indicates the results on test sets when the performance is the best on the development set. H28a, H28b, H28c, H28d are obtained from phase 1 with configuration 12, 13, 20, 24 respectively. The (*) symbols indicate the result are the best on the test set but the development set. Network setting={dropout(keep):0.8, learning rate:0.05, number node in hidden layers:200, input embedding size: 100}. (♣) is the results we have summited for official evaluation

6 Conclusion

We built a system for the legal information retrieval/entailment task. For the first phase, the most relevant article given a question was obtained based on two steps. A set of relevant articles were extracted using cosine TF-IDF vectors with some additional techniques to improve the performance such as query expansion with word embedding sources. Then, retrieved articles were re-ranked based on SVM rank to achieve the most relevant article. For the second phase, we employed two deep learning models to recognize entailments between articles and questions. One of limitation of this approach is that the size of training corpus is too small to train a stable model. In future, incorporating engineering features can be good ways to improve the performance of the system.

7 Acknowledgments

This work was supported by JAIST CREST, Japan.

References

[1] Wafa N Bdour and Natheer K Gharaibeh. Development of yes/no arabic question answering system. *International Journal of Artificial Intelligence and Applications*, 4(1):51–63, 2013.

[2] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[3] Danilo Carvarlho, Minh-Tien Nguyen, Chien Tran, and Le-Minh Nguyen. Lexical-morphological modeling for legal text analysis. In *Lecture notes in computer science:New Frontiers in Artificial Intelligence*. Springer, 2016.

[4] Y. Kano. Keyword and snippet based yes/no question answering system for coliee 2015. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.

[5] Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kihyun Hong, and Young-Yik Rhim. Ensemble based legal information retrieval and entailment system. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016.

[6] Mi-Young Kim and Ken Goebel, Randy Satoh. Coliee-2015 : Evaluation of legal question answering. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.

[7] Mi-Young Kim, Ying Xu, and Randy Goebel. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*, pages 244–258. Springer, 2014.

[8] Mi-Young Kim, Ying Xu, Randy Goebel, and Ken Satoh. Answering yes/no questions in legal bar exams. In *JSAI International Symposium on Artificial Intelligence*, pages 199–213. Springer, 2013.

[9] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[10] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[11] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[12] Truong-Son Nguyen, Viet-Anh Phan, Thanh-Huy Nguyen, Hai-Long Trieu, Ngoc-Phuong Chau, Trung-Tin Pham, and Le-Minh Nguyen. Legal information extraction/entailment using svm-ranking and tree-based convolutional neural network. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016.

[13] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[14] S. Pal Sushmita, A. Kanapala. Legal information retrieval task: Participation from ism, dhanbad. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.

[15] Satoshi Tojo. Eighth international workshop on juris-informatics (jurisin 2014). In *JSAI International Symposium on Artificial Intelligence*. Springer, 2014.

[16] Duc-Vu Tran, Viet-Anh Phan, Hai-Long Trieu, and Le-Minh Nguyen. An approach for retrieving legal texts. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.

[17] Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. Answering legal questions by mining reference information. In *JSAI International Symposium on Artificial Intelligence*, pages 214–229. Springer, 2013.

EPiC Series in Computing

Volume 47, 2017, Pages 43–56

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Improving Legal Information Retrieval by Distributional Composition with Term Order Probabilities

Danilo S. Carvalho*, Vu Duc Tran, Khanh Van Tran, and Minh Le Nguyen

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
Nomi, Ishikawa, Japan.
{danilo, vu.tran, tvkhanh, nguyenml}@jaist.ac.jp

Abstract

Legal professionals worldwide are currently trying to get up-to-pace with the explosive growth in legal document availability through digital means. This drives a need for high efficiency Legal Information Retrieval (IR) and Question Answering (QA) methods. The IR task in particular has a set of unique challenges that invite the use of semantic motivated NLP techniques. In this work, a two-stage method for Legal Information Retrieval is proposed, combining lexical statistics and distributional sentence representations in the context of Competition on Legal Information Extraction/Entailment (COLIEE). The combination is done with the use of disambiguation rules, applied over the rankings obtained through n-gram statistics. After the ranking is done, its results are evaluated for ambiguity, and disambiguation is done if a result is decided to be unreliable for a given query. Competition and experimental results indicate small gains in overall retrieval performance using the proposed approach. Additionally, an analysis of error and improvement cases is presented for a better understanding of the contributions.

1 Introduction

The ability of answering questions is a long sought goal in the field of Natural Language Processing (NLP). Legal questions, in particular, pose a big challenge to current NLP techniques, due to their often complex syntactical structure and *domain dependent* terminology. As we experience an explosive growth in legal document availability through digital means, the need for higher efficiency Legal Information Retrieval (IR) and Question Answering (QA) methods becomes critical for the practice of legal profession. Current increase in information analysis capabilities is not keeping up with such intense growth, leading to under-utilization of available legal resources and to potential for information quality issues. This also brings up the matter of professional ethics and liability on law practice, due to the fundamental importance of relevant and correct information in legal practice.

A basic step into answering a legal question is retrieving the relevant legal information, e.g., laws, facts and previous verdicts, and aligning their contents in order to decide their

*Supported by CNPq – Brazil scholarship grant

applicability to the given question. This is a challenging problem, since laws are written with abstraction in mind, to cover most possible scenarios of any predicted situation. A semantically motivated NLP framework is then needed to allow abstraction and realization of the written law. With this in mind, approaches for both abstraction [1] and realization [2] have been proposed, with *Machine Learning* (ML) techniques taking an increasingly important role in language analysis [8, 6]. ML-based *distributional semantics* approaches have recently shown promising results in general domain IR [7, 16, 15]. However, they still perform and generalize poorly in the legal domain, due to the difficulty of training under datasets of relatively small size and a wide variety of topics with different vocabulary and semantics. For this reason, an approach combining lexical statistics and distributional semantics would be appropriate to leverage both accuracy in literal matching and the possibility of limited abstraction in the form of word/sentence embeddings. However, exploration of such combination has been limited, to the best knowledge of the authors.

In this work, we propose a two-stage method for Legal Information Retrieval aimed at Question Answering, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE). The stages are: 1) Relevance analysis and 2) Relevance disambiguation. The method is based on a mixed n-gram language model for relevance analysis, complemented by distributional semantic similarity on cases in which relevance cannot be decided. A technique for obtaining sentence representations from word embeddings is used, which is able to capture semantic information from a sentence, and has advantage when relevant texts have little lexical matching.

The remainder of this paper is organized as follows: Section 2 presents related works and relevant results; Section 3 details the Legal Question Answering problem and the COLIEE competition shared task; Section 4 explains our approach to the competition problem; Section 5 presents the experimental setting, results and some discussion about the findings; Finally, Section 6 offers some concluding remarks.

2 Related Work

Recent developments in Legal Information Retrieval (LIR) and Legal Question Answering (LQA) include the work of Liu, Chen and Ho [13], which presented a method called three-phase prediction (TPP) for retrieval of relevant statutes in Taiwanese criminal law, given queries presented in non-legal language. It employed a hierarchical ranking approach for law corpora, combining several Information Retrieval techniques, as well as Machine Learning and feature selection. The use of distributional semantic representation into LQA has an interesting case in the work of Kim et. al. [10], in which a SVM-based ranking method using training features such as lemmatized words intersection, dependency pairs and TF-IDF is used for LIR, while Recognition of Textual Entailment (RTE) was performed by a binary SVM classifier, trained on a set of features including semantic similarity calculated from word2vec [14] embeddings. This method won the combined LQA (LIR + RTE) COLIEE competition in 2016.

The creation of sentence embeddings on limited training data scenarios was approached by Carvalho and Nguyen [3], using a probability table obtained from word ordering in the corpus sentences, to calculate an attention index for composition of word embeddings through a simple summation formula. The method improved overall accuracy on segmentation of patent documents from the US patent office.

In the context of the solo LIR COLIEE competition, [9] proposed an ensemble similarity using a least square method (LSM) and linear discriminant analysis (LDA ensemble) including a variety of features such as lexical similarity, syntactic similarity and semantic similarity. This

work showed the best LIR performance in 2016. Our previous main work in COLIEE [4] introduced a ranking method called R_2NC (Ranking Related N-gram Collections), based on a mixed size n-gram language model, which used links between the documents (articles) in the legal corpus to build n-gram collections for each of them, and a variant of TF-IDF scoring to rank them. It achieved a LIR 2nd place in 2015.

The method here proposed explores the use of distributional sentence representations obtained through the use of Carvalho and Nguyen's method [3] as a deciding factor in LIR for cases in which R_2NC has ambiguous rankings, i.e., arbitrarily close, or insecure scores, i.e., arbitrarily low. This is done by a set of simple rules for exchanging or adding documents in the R_2NC retrieved document list.

3 Legal Question Answering – COLIEE

Answering a legal question comprises: (i) collecting the knowledge required for understanding the given question, and then (ii) inferring the appropriate and correct answer. In the context of the *Competition on Legal Information Extraction/Entailment (COLIEE)*[1], the question is a legal statement varying from specific to general cases, and the required knowledge is embodied in the law itself, in the form of organized articles that compose a fragment of the Japanese Civil Code. The Japanese Civil Code is composed by a collection of numbered articles, each one containing a set of declarations pertaining to a specific topic under law, e.g., labor contracts, mortgages. Given the knowledge from the relevant law articles, the legal statement shall either agree or disagree with the interpretation of the articles, which leads to either affirmative or negative answer accordingly.

In COLIEE 2017, activities (i) and (ii) are separated in corresponding *phases*:

- Phase One (IR): given a legal question, retrieving relevant articles from the provided part of the Japanese Civil Code.
- Phase Two (Answering): given a question, from the system retrieved list of relevant articles to the question, deciding the entailment relationship between the retrieved articles and the provided question.

Legal text inherently distinguishes itself from other types of written communication, by the uniqueness of both its content and intent: to express rules and situations where they apply. This should be done in an abstractive way and with no ambiguity, such that the rules shall be applied only to the intended cases and no case is under conflicting rules. Those requirements certainly enforce a language with stricter terminology and syntax, a higher abstraction level, and with semantics that are foreign or even conflicting with common language use. Such characteristics make the use of *Distributed Semantics* to be *corpus specific* on legal text. However, for answering legal questions it is critical to identify the corpus specific and common senses of terms, since law is to be applied in daily life, both of them are used. Besides, another noteworthy characteristic of legal text is its preference for longer sentences, with enumeration or itemization, causing more difficulty for automatic parsing.

For this competition, we decided to focus efforts on the Information Retrieval aspect (phase one). Thus, the contributions in this work do not cover the Answering of the questions (phase two), which also deals with Recognition of Textual Entailment (RTE) between questions and articles.

[1] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2017/

4 Proposed Approach

Given a legal question, presented in natural language, the legal information retrieval comprises two consecutive stages: 1) relevance analysis and 2) relevance disambiguation. Firstly, a ranked list with a limited number of relevant articles is obtained by using R_2NC [4] (Section 4.1). Next, the first two articles in the ranked list are evaluated over ambiguity (the scores are too close from each other) and insecurity (the scores are too low), under specified thresholds. If the articles' scores are ambiguous or insecure, sentence embeddings are obtained for both the question and each article in the ranked list using *word2vec* [14] and *Term Order Probabilities* (TOP) [3] (Section 4.3). A new ranked list is obtained by calculating the highest sentence embedding cosine similarity of each pair (question, article). Finally, the two lists are compared through a set of rules, and a decisive set of relevant articles is selected. This process was developed after preliminary experiments past competition experience indicated a very high correlation of ambiguous or low scoring articles and retrieval mistakes. A diagram of the overall process flow is shown in Fig. 1. The next sections present each stage in detail.

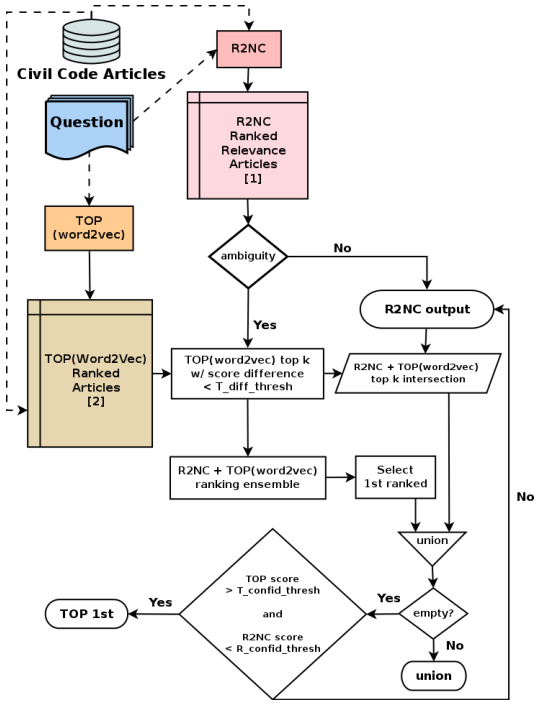

Figure 1: Architecture of the proposed system as a process flow. Legal articles ranked by R_2NC are evaluated regarding the confidence in their score and exchanged or complemented by articles ranked by sentence semantic similarity calculated from TOP embeddings.

4.1 Relevance analysis

The relevance analysis stage was done entirely with R_2NC [4], which can be summarized in the following process:

1. Collect the content for each article;

2. Check references between articles and annotate;
3. Tokenize and POS-tag;
4. Remove stopwords: determiners, conjunctions, prepositions and punctuation;
5. Lemmatize words;
6. Generate n-grams;
7. Expand the n-gram set, by including referenced articles' n-grams;
8. Associate article number and references;
9. Store the model.

Except for step 4, each step adds new information to the model. The information is obtained from the text, references, and morphological analysis, e.g., POS-tags, lemmas. If an article has references, its n-gram set incorporates the references' n-grams. In this way, all the necessary information for interpretation of any single article is self-contained. Besides the n-grams, links between the articles are also stored. The same process is repeated for the questions to include the training data information, and n-gram sets from the trained questions are included in the associated articles' n-gram models.

Tokenization and lemmatization were done using *NLTK*[2] (v. 3.2.1) with the Punkt tokenizer and WordNetLemmatizer modules, respectively. Those modules were used with their unchanged default models and settings, trained with corpora prepared from the English Penn Treebank by Kiss and Strunk [11] and WordNet [3], respectively. POS-tagging was done using *Stanford Tagger*[4] (v. 3.5.2), using the unchanged *english-left3words-distsim* model, which is trained on the part-of-speech tagged WSJ section of the Penn Treebank corpus. Fig. 2 illustrates the R_2NC process flow. Fig. 3 illustrates the n-gram model creation scheme.

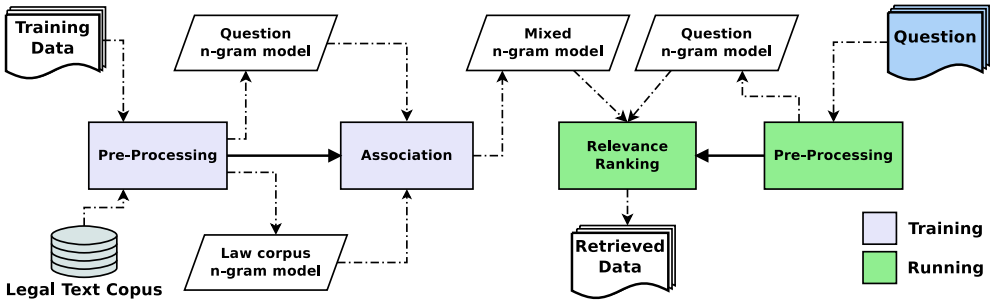

Figure 2: R_2NC process flow. N-gram language models from both the law corpus and training questions are associated into a mixed n-gram model. Article relevance for unseen questions is evaluated using this mixed model.

The relative relevance of an article with regard to the content of a question is scored using the following formula:

[2] www.nltk.org
[3] wordnet.princeton.edu
[4] nlp.stanford.edu/software/tagger.shtml

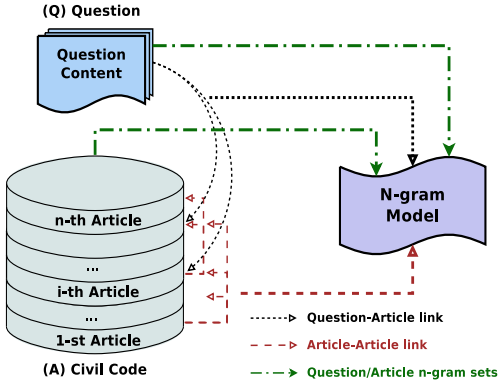

Figure 3: The n-gram model construction scheme. Both article-article and question-article links are stored, and the respective document n-gram sets are associated. A single association index is generated for each article.

$$score = \frac{\sum_{\forall t} idf(t)}{I_q \times |q\_ng\_set| + I_{art} \times |art\_ng\_set|}, \quad t \in (q\_ng\_set \cap art\_ng\_set) \tag{1}$$

where q_ng_set is the set of n-grams for the question, art_ng_set is the set of n-grams for the article in the stored model, I_q is the relative significance of the question n-gram set size and I_{art} is the relative significance of the article n-gram set size. $idf(t)$ is the Inverse Document Frequency for the term t over the articles collection

$$idf(t) = log\frac{N}{df_t} \tag{2}$$

where N is the total number of articles and df_t is the number of articles in which t appears. Both I_q and I_{art} are parameters. The scored articles are then ranked from the highest score to the lowest.

4.2 Term Order Probabilities

The *Term Order Probabilities* (TOP) [3] is an inexpensive method for combining word embeddings into sentence or document embeddings, while keeping word order information and highlighting or attenuating uncommon/common word order combinations, respectively.

It consists in two steps:

1. Calculate $P(t_1, t_2, d)$: the probability of any pair of terms (words, n-grams) t_1 and t_2 appearing in this particular order in the corpus, separated by a maximum of d words. $P(t_1, t_2, d)$ is calculated as:

$$P(t_1, t_2, d) = \frac{\#(t_1, t_2, d)}{\#(t_1, t_2, d) + \#(t_2, t_1, d)} \tag{3}$$

where $\#(t_1, t_2, d)$ is the number of occurrences of t_1 appearing before t_2 in the reference corpus.

2. Combine the embeddings into one, using the formula

$$\frac{\sum_{i=0}^{k} e(t_i) + \sum_{i,j=0:i<j}^{k} (e(t_i) + e(t_j)) * (1 - P(t_i, t_j))}{k + \sum_{i=0}^{k} k - i} \tag{4}$$

where $e(t_i)$ are the term t_i embeddings and k is the length of the sentence. The resulting vector is the sum of the weighted combinations of all embedding pairs in the sentence, and the value j defines a fixed size window of distance d for each term, improving efficiency in longer sentences by limiting the number of calculations. The contribution of each term to the sentence embedding is weighted by an "attention index" $(1 - P(t_i, t_j))$, representing how unlikely the term is to appear in that context. In this way, uncommon patterns have a higher contribution, helping to distinguish even between similar sentences.

The probability table $P^{n \times n}$ is calculated for the entire target corpus, where n is the vocabulary size. It is a sparse matrix that can be efficiently stored and accessed. TOP differs from other sentence embedding methods, such as Paragraph Vectors [12], in that it can work well with a relatively reduced amount of textual data for training. For this reason, it was chosen for obtaining embeddings from the COLIEE questions and articles' sentences, considered a very small corpus by current standards (see Section 5).

Although the TOP method can use a variety of word embeddings, in this work we chose word2vec [14] to obtain the distributional representations of words. Thus, all mentions to TOP henceforth mean the Term Order Probabilities method applied to word2vec embeddings.

4.3 Relevance disambiguation

Having obtained a ranked list of articles from the previous stage, the relevance disambiguation stage is triggered when the R_2NC scores of the first and second ranked articles are close or ambiguous, falling under the following ambiguity condition:

$$R\_score(a_1, q) - R\_score(a_2, q) < R\_ambi\_thresh$$

where $R\_score(a_i, q)$ is the R_2NC score of the article a_i for the question q, and R_ambi_thresh is the parameter representing the specified lower bound of ranking ambiguity by R_2NC. Then a candidate list using TOP cosine similarity ranking is created by selecting top k articles under the condition:

$$T\_cand(q) = \{a_i \mid \forall i \leq k \; and \; j < i \; |T\_score(a_i, q) \; - T\_score(a_j, q)| < T\_diff\_thresh\}$$

where $T\_cand(q)$ is the candidate list from TOP, $T\_score(a_i, q)$ is the TOP score, i.e. similarity, of the article a_i for the question q, and T_diff_thresh represents the upper bound relative to the first ranked article by TOP. Under the prior condition, at most top k articles having close TOP scores are selected into $T\_cand(q)$. If any of the articles retrieved by R_2NC is also in $T\_cand(q)$, it is selected as the relevant article. Additionally, an aggregated ranked list is generated by linear ranking ensemble from R_2NC and TOP for each article in $T\_cand(q)$. The top 1 of the aggregated ranked list is added to the output relevant list (preferably R_2NC in cases of equality).

If previous steps result in no article selected, the system checks if the article ranked first by R_2NC has low score and the other ranked first by TOP has high score under confidence conditions:

- $R\_score(a_1, q) < R\_confid\_thresh$
- $T\_score(a_1, q) > T\_confid\_thresh$

where R_confid_thresh, T_confid_thresh are the confidence thresholds over R_2NC, and TOP scoring respectively. If the prior condition is satisfied, the first ranked article by TOP is selected as the relevant article. Otherwise, R_2NC output is selected as relevant articles.

5 Experiments and Results

5.1 Experimental Setup

The legal question answering dataset was obtained from the published data for the COLIEE shared task [5], consisting in a text file with a fragment of the Japanese Civil Code translated into English and a set of XML files with training data. The training set for the two tasks contains 580 pairs (question, relevant articles). The Japanese Civil Code fragment contains 1057 articles, with a total of approximately 1700 sentences and 71500 words. Experiments are then conducted to evaluate Information Retrieval methods.

Additional data used in the experiments includes the training segment of "1 billion word language model benchmark" corpus [5] and the complete Japanese Civil Code[6], which were used to train the Distributional Semantics model (word2vec). The amount of available data for common vs. legal text was highly unbalanced, so as a balancing measure, the legal text was replicated until it composed a certain fraction (around 25%) of the combined data. The combined size of the corpora after balancing is approximately 1.2 billion words. Pure common text embeddings were also tested, in particular, the Google News dataset pre-trained vectors [7]. However, this resulted in poor retrieval performance, most probably due to the absence of legal vocabulary and corresponding semantics.

We focus on detailed experiments for R_2NC and R_2NC+TOP, but not solely TOP. Preliminary experiments showed that TOP similarity ranking alone is consistently worse than R_2NC.

5.2 Parameter adjustment

R_2NC relative significance parameters were adjusted by leave-one-out validation on the training data. The best setting was $I_q = 0.98, I_{art} = 0.02$.

Variants of TOP models were trained as follows. The data for training word embedding models: lemmatized or non-lemmatized texts. The data for training TOP models: the training questions and provided articles, with or without the whole Japanese Civil Law. The best setting was using non-lemmatized text for training the word embedding model, and training TOP models without the whole Japanese Civil Law.

R_2NC+TOP parameters were selected by conducting leave-one-out experiments with adjusting the parameters over certain ranges (Table 1). The adjustment results in higher performance on certain values between the mean and bounds of each range, then lower on the bounds, while in the middle, we got steady performance, hence average values were selected.

Word2vec parameters were set as the following and not changed: $d = 200$ (dimensionality), $cbow = 0$ (using skip-gram mode), $window = 10$, $negative = 0$.

[5] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2017/
[6] www.japaneselawtranslation.go.jp
[7] code.google.com/archive/p/word2vec

Parameter	Range	Selected
R_ambi_thresh	[0.005, 0.03]	0.012
k	[2, 6]	4
T_diff_thresh	[0.005, 0.03]	0.025
R_confid_thresh	[0.3, 0.5]	0.42
T_confid_thresh	[0.7, 0.9]	0.8

Table 1: R_2NC+TOP parameter adjustment.

5.3 Evaluation Method

For the relevance analysis stage, leave-one-out validation was used to evaluate the potential recall of the model for a limited size ranked list of articles. Performance for phase one was evaluated using precision (P), recall (R) and F-measure (F) as metrics (Eqs. (5), (6) and (7)).

$$P = \frac{Cr}{Rt} \quad (5) \qquad R = \frac{Cr}{Rl} \quad (6) \qquad F = \frac{2(P * R)}{P + R} \quad (7)$$

where Cr counts the correctly retrieved articles for all queries, Rt counts the retrieved articles for all queries, Rl counts the relevant articles for all queries, Cq counts the queries correctly confirmed as true or false and Q counts all the queries.

5.4 Competition Results

Participant	Precision	Recall	F-measure
iLis7-1	0.734	0.554	0.632
JNLP1-RT⋆	0.689	0.545	**0.609**
JNLP1-R⋄	0.686	0.536	0.602
KID17	0.703	0.518	0.596
iLis7-2	0.654	0.500	0.567

Table 2: Competition results for phase one (IR). The first five ranked participants are shown in order, along with their achieved metrics. ($\star$) denotes the method presented in this paper, while $\diamond$ was a pure R_2NC run.

The approach here presented was ranked at second place in the LIR competition (phase one). The third place was achieved by a pure R_2NC run. The results, shown in Table 2, indicate a small improvement in both precision and recall, meaning that the added relevance disambiguation stage contributed one or more relevant articles to the R_2NC retrieved list without introducing non-relevant ones. Relevance disambiguation changed the final ranking for 6 out of the 78 questions (7.7%) in the test set. From the TOP modified rankings, half (2) received relevant articles and no irrelevant ones, while the other half had no improvement.

Additional experiments were performed after the competition, with the ground truth data being made available by the COLIEE organizers. As shown in Table 3, it was possible to obtain further improvement over the competition results.

This was achieved by changing R_2NC to a more aggressive setting of $I_q = 0.99, I_{art} = 0.01$, with extreme penalization of article length. This setting resulted in marginal gains in the training data (less than 1×10^{-4} in F-score, with a value of 0.519), on a leave-one-out test using the training data section files, i.e., leaving a single file for ranking (e.g., riteval_H18.xml)

Method	Precision	Recall	F-measure
JNLP1-RT	0.701	0.554	**0.619**
JNLP1-R	0.689	0.545	0.609

Table 3: Post-competition results for phase one (IR). Indicated methods are the same as in Table 2

and training with the others. Nevertheless, it improved the overall result in the competition dataset.

5.5 Error Analysis and Discussion

This subsection answers the question: how does TOP help improve on top of R_2NC? The retrieval system with R_2NC supported by TOP shows the improvement on top of R_2NC in the following two examples (Examples 1 and 2). On the way of investigating the contribution of TOP, we further analyze other examples where TOP results better than R_2NC. The analysis suggests the semantic characteristics of TOP over R_2NC which is limited to lexical matching.

Example 1: Question H28-22-4. R_2NC selects non-relevant Article 606. R_2NC supported by TOP selects relevant Article 613.

> **Question H28-22-4.** In cases where a lessee lawfully subleases a leased Thing, if the lessor assumes an obligation to the lessee to effect repairs of the leased Things, the lessor shall also assume a direct obligation to the sublesee to effect repairs of the leased Things.
>
> [✗] [R_2NC]**Article 606.**
> (1) A lessor shall assume an obligation to effect repairs necessary for using and taking the profits of the leased Things.
> (2) The lessee may not refuse if the lessor intends to engage in any act that is necessary for the preservation of the leased Thing.
>
> [✓] [$R_2NC+TOP$]**Article 613.**
> (1) If a lessee lawfully subleases a leased Thing, the sublessee shall assume a direct obligation to the lessor. In such cases, advance payment of rent may not be asserted against the lessor.
> (2) The provisions of the preceding paragraph shall not preclude the lessor from exercising his/her rights against the lessee.

In question H28-22-4, R_2NC selects Article 606 as the relevant article while the combination selects Article 613 which is actually relevant (Example 1). Looking at the two articles, R_2NC results in ambiguity (R_2NC scores are 0.808 versus 0.798 for Article 606 and 613 respectively). It turns out that the two articles are very lexically similar to the question. The difference is in logical structures. On one hand, the effectuation of the first paragraph of Article 606 is very similar with the requisite of the question. On the other hand, the effectuation of the question is matched with the one of the first paragraph of Article 613 in the reversed way. That is "sublessee" and "lessor" change their roles between the question and Article 613.

In another configuration of R_2NC, with adjusting the relative significance ($I_q = 0.99, I_{art} = 0.01$), R_2NC results in ambiguity in the returned ranked articles for question H28-34-4 for which $R_2NC + TOP$ selects Article 975 instead of Article 763 (Example 2). In this question, Article 763 mentions about "husband and wife may ..." but not "make their will on the same certificate". Despite that, the match is so decisive by R_2NC that Article 763 is picked instead of Article 975 because of word scattering in the relevant article. Even then, TOP is able to pick

Example 2: Question H28-34-4. R_2NC selects non-relevant Article 763. R_2NC supported by TOP selects relevant Article 975.

> **Question H28-34-4.** A husband and wife may make their will on the same certificate.
>
> [✗][R_2NC]**Article 763.**
> A husband and wife may divorce by agreement.
>
> [✓][$R_2NC + TOP$]**Article 975.**
> A will may not be made by two or more persons on the same certificate.

up the order of the scattered words, the expressions "will ... made ... on the same certificate" and "make ... will ... on the same certificate" are similar in TOP vector space.

This indicates a TOP advantage on capturing disjoint expressions that make the core of the sentence topic, which may also include inflections and conjugations.

To assess these characteristics of TOP, we observe some cases where TOP draws correct outputs while R_2NC fails. Those are of questions H28-11-2, H28-22-2, and H28-26-5 (Examples 3,4, and 5).

Example 3: Question H28-11-2. R_2NC selects non-relevant Article 305 and 296. TOP selects relevant Article 304 (ranked 2nd by R_2NC).

> **Question H28-11-2.** Extension of security interest to proceeds of collateral may be done with respect to a right of retention, a statutory lien, a pledge and a mortgage.
>
> [✗][R_2NC]**Article 305.**
> The provisions of Article 296 shall apply mutatis mutandis to statutory liens.
>
> [✗][R_2NC]**Article 296.**
> A holder of a right of retention may exercise his/her rights against the whole of the Thing retained until his/her claim is satisfied in its entirety.
>
> [✓][TOP]**Article 304.**
> (1) A statutory lien may also be exercised against Things including monies that the obligor is to receive as a result of the sale, lease or loss of, or damage to, the subject matter of the statutory lien; provided, however, that the holder of the statutory lien must attach the same before the payment or delivery of the monies or other Thing.
> (2) The provisions of the preceding paragraph shall likewise apply to the consideration for real rights established by the obligor on the subject matter of the statutory lien.

In question H28-11-2 (Example 3), TOP shows the advantage of semantical similarity over lexical matching by R_2NC. Article 305 with complementary text from Article 296 has higher lexical matching with the question than Article 304. While Article 305 complemented by Article 296 shares phrases "a right of retention", and "statutory lien", Article 304 only shares phrase "statutory lien" with the question, then, certainly has lower score than Article 305 by R_2NC using lexical matching. On the other side, in the distributed vector space, "collateral" in the question and "payment", and "monies" in Article 304 are similar, which benefits from distributional word similarity capability of TOP.

In question H28-22-2 (Example 4), the relevant Article 572 is selected by TOP, but ranked 8th by R_2NC. The relevant score of Article 572 given the question by R_2NC is heavily penalized

Example 4: Question H28-22-2. R_2NC selects non-relevant Article 635. *TOP* selects relevant Article 572 (ranked 8th by R_2NC).

> **Question H28-22-2.** In cases where there is a special agreement, in a contract of sale, to the effect that the seller will not provide the warranties against defects, if the seller knew but did not disclose that there is any latent defect in the subject matter of a sale, he/she may not be released from the warranties against defects.
>
> [✗][R_2NC]**Article 635.**
> If there is any defect in the subject matter of work performed and the purpose of the contract cannot be achieved because of the defect, the party ordering the work may cancel the contract; provided, however, that this shall not apply to a building or other structure on land.
>
> [✓][*TOP*]**Article 572.**
> Even if the seller makes a special agreement to the effect that the seller will not provide the warranties set forth from Article 560 through to the preceding Article, the seller may not be released from that responsibility with respect to any fact that the seller knew but did not disclose, and with respect to any right that the seller himself/herself created for or assigned to a third party.

by the long length of the article complemented with a considerable number of referred articles. Even after reducing the effect of article length in R_2NC computation, Article 572 is still ranked 3rd. *TOP*, however, has the advantage of only focusing on the most similar sentence in the article.

Example 5: Question H28-26-5. R_2NC selects non-relevant Article 656. *TOP* selects relevant Article 648 (ranked 4th by R_2NC).

> **Question H28-26-5.** In the absence of any special agreements, the mandatary may claim remuneration from the mandator.
>
> [✗][R_2NC]**Article 656.**
> The provisions of this Section shall apply mutatis mutandis to mandates of business that do not constitute juristic acts.
>
> **Question H25-29-E.** (Relevant to Article 656) A mandatary of a quasi-mandate contract may not claim remuneration from a mandator before performance, even with special agreements that a mandatary may claim remuneration before he/she administers the mandated business.
>
> [✓][*TOP*]**Article 648.**
> (1) In the absence of any special agreements, the mandatary may not claim remuneration from the mandator.
> (2) In cases where the mandatary is to receive remuneration, the mandatary may not claim the same until and unless he/she has performed the mandated business; provided, however, that if the remuneration is specified with reference to period, the provisions of Paragraph 2 of Article 624 shall apply mutatis mutandis.
> (3) If the mandate terminates during performance due to reasons not attributable to the mandatary, the mandatary may demand remuneration in proportion to the performance already completed.

In question H28-26-5 (Example 5), while it is obvious that the question is a perfect match of the first paragraph of Article 648, R_2NC and *TOP* selections are different. R_2NC with the penalty from article length, and the inclusion of relevant question gives a low score for Article 648, hence, selects Article 656. Besides, *TOP* only looks at the highest match, then selects Article 648.

The analysis suggests complementary characteristics of *TOP* and R_2NC and shows potential

of exploiting *TOP* to improve the retrieval system.

6 Conclusion

Information Retrieval in the legal domain is a challenging task, due to its unique combination of complex syntax, domain dependency and high abstraction level. Such combination presents a valuable ground for the application of semantically motivated NLP techniques, capable of a limited level of abstraction. Distributional semantic representations fit this category of techniques, but despite promising results for general IR, still lack on performance in the legal domain. Despite this, in this work we propose a method for combining a pure lexical approach, based on n-gram statistics, with distributional sentence representations in the context of Competition on Legal Information Extraction/Entailment (COLIEE). The combination is done by means of disambiguation rules, applied when the lexical approach is deemed insufficient to decide on the set of relevant documents for a given query, also knowing that the distributional approach is weak by itself.

Competition results and further experiments indicate that it is possible to obtain small gains in overall retrieval performance through the proposed approach. Analysis of the errors and improvements observed in the training and competition data revealed complementary characteristics from the lexical and distributional approaches, e.g., sensitivity to document size, which can be exploited in order to cover each other's weaknesses and further improve performance.

Acknowledgements

This work was supported by JSPS KAKENHI Grant number 15K16048, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST.

References

[1] Kevin D. Ashley and Edwina L. Rissland. Law, learning and representation. *Artificial Intelligence*, 150(1):17 – 58, 2003.

[2] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria. Automatic semantics extraction in law documents. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, pages 133–140, New York, NY, USA, 2005. ACM.

[3] Danilo S. Carvalho and Minh-Le Nguyen. Efficient neural-based patent document segmentation with term order probabilities. In *Proceedings of the 25th European Symposium on Artificial Neural Networks, ESANN 2017 (preprint)*, 2017.

[4] Danilo S. Carvalho, Minh-Tien Nguyen, Tran Xuan Chien, and Minh-Le Nguyen. Lexical-morphological modeling for legal text analysis. *New Frontiers in Artificial Intelligence (Lecture Notes in Artificial Intelligence)*, 10091, 2016.

[5] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[6] Michael Curtotti, Eric McCreath, Tom Bruce, Sara Frug, Wayne Weibel, and Nicolas Ceynowa. Machine learning for readability of legislative sentences. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, ICAIL '15, pages 53–62, New York, NY, USA, 2015. ACM.

[7] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, pages 795–798, New York, NY, USA, 2015. ACM.

[8] Teresa Gonçalves and Paulo Quaresma. Is linguistic information relevant for the classification of legal texts? In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, pages 168–176, New York, NY, USA, 2005. ACM.

[9] Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kyhyun Hong, and Young-Yik Rhim. An ensemble based legal information retrieval and entailment system. In *Tenth International Workshop on Juris-informatics (JURISIN)*, 2016.

[10] Mi-Young Kim, Ying Xu, Yao Lu, and Randy Goebel. Legal question answering using paraphrasing and entailment analysis. In *Tenth International Workshop on Juris-informatics (JURISIN)*, 2016.

[11] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[12] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[13] Yi-Hung Liu, Yen-Liang Chen, and Wu-Liang Ho. Predicting associated statutes for legal problems. *Information Processing & Management*, 51(1):194–211, 2015.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[15] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(4):694–707, April 2016.

[16] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, pages 12:1–12:8, New York, NY, USA, 2015. ACM.

EPiC Series in Computing

Volume 47, 2017, Pages 57–67

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Analyzable Legal Yes/No Question Answering System using Linguistic Structures

Yoshinobu Kano[1,*], Reina Hoshino[1] and Ryosuke Taniguchi[1]

[1] Faculty of Informatics, Shizuoka University

kano@inf.shizuoka.ac.jp, rhoshino@kanolab.net, rtaniguchi@kanolab.net

Abstract

A central issue of yes/no question answering is usage of knowledge source given a question. While yes/no question answering has been studied for a long time, legal yes/no question answering largely differs from other domains. The most distinguishing characteristic is that legal issues require precise linguistic analysis such as predicates, case-roles, conditions, etc. We have developed a yes/no question answer-ing system for answering questions in a legal domain. Our system uses linguistic analysis, in order to find correspondences of predicates and arguments given problem sentences and knowledge source sentences. We applied our system to the COLIEE (Competition on Legal Information Extraction/Entailment) 2017 task. Our team shared the second place in this COLIEE 2017 Phase Two task, which asks to answer yes or no given a problem sentence. This result shows that precise linguistic anal-yses are effective even without the big data approach with machine learning, rather better in its analyzable design for future improvements.

1 Introduction

Automatic question answering is attracting more interests recently. Due to the increasing expectation to the Artificial Intelligence (AI) technologies, people tend to regard question answering systems as a brand new technology emerged today. However, most successful systems employ rather traditional techniques of question answering which have decades of history (Lin & Pantel, 2001) (Ravichandran & Hovy, 2002) (Yu & Hatzivassiloglou, 2003) (Pinto et al., 2003) (Cui et al., 2005) (Xue et al., 2008) (Bian et al., 2008), including series of shared tasks such as TREC (Voorhees & Harman, 2005), NTCIR (Kando et al., 1999) and CLEF (Braschler, 2001). This paper describes our challenge to the COLIEE 2017 legal bar exam, which asks participants to answer true or not based on the Civil Law Articles, given text drawn from the Japanese legal bar exam.

A variety of algorithms and systems has been proposed for question answering. Typically, these question answering systems used *big data* for answering questions (Kwok et al., 2001) (Etzioni et al.,

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 57–67

t1: （留置権の行使と債権の消滅時効）
第三百条　留置権の行使は、債権の消滅時効の進行を妨げない。
(Exercise of Rights of Retention and Extinctive Prescription of Claims)Article 300
The exercise of a right of retention shall not preclude the running of extinctive prescription of claims.
t2: 留置権者が留置物の占有を継続している間であっても，その被担保債権についての消滅時効は進行する。
Even while the holder of a right to retention continues the possession of the retained property, extinctive prescription runs for its secured claim.

Fig. 1. An example of COLIEE legal bar problem which asks to answer t1 entails t2 or not. The correct answer is “yes” in this example. t1 is not given for the Phase Three before COLIEE 2016 and Phase Two in COLIEE 2017.

2004) (Jeon et al., 2005) (Kanayama et al., 2012). For example, Dumais et al. (Dumais et al., 2002) focused on the redundancy available in large corpora as an important resource. They used this redundancy to simplify their algorithm and to support answer mining from returned snippets. Their system performed quite well given the simplicity of the techniques being utilized.

The now widely known IBM Watson system (D. Ferrucci, 2012) would be considered as a typical example of such a question answering system of the big data approach. The IBM Watson system won in the Jeoperdy! Quiz TV program competing with human quiz winners. The core Watson system employed a couple of open source libraries, including the traditionally well-designed DeepQA system (Ferrucci, 2011) as its skeleton of question answering processing. Because their target domain, the Jeoperdy! Quiz, could ask broad range of questions, they collected a huge amount of knowledge sources from the Internet, etc., extracting relevant knowledge by combining a couple of different natural language processing (NLP) techniques.

Answering university examinations is another example. The Todai Robot project (Arai, 2015) is a challenge to solve Japanese university examinations, focusing towards attaining a high score in the National Center Test for University Admissions, and passing the entrance exam of the University of Tokyo (Todai). Although the Todai Robot project tries to achieve higher scores, their aim is rather to reveal the current performance and limitation of the existing AI technologies, using the examinations as its benchmark, similar to the COLIEE’s legal bar exam task. In contrast to the COLIEE task, the challenge of Todai Robot project includes variety of subjects including Mathematics, English, Japanese, Physics, History, etc. all written in Japanese language. While solving any problem of these subjects could be considered as question answering, some problems require special technologies. For example, Mathematics and Physics require to process formula; Japanese requires to infer emotions of story characters. Solving the History subjects might be considered as rather an extension of the existing question answering issues. The Todai Robot project achieved better scores than the average of the real human applicants in their Mock Exam challenges.

Recognition of textual entailments (RTE or RITE) is another related issue. RTE has been intensively studied for recent days, including shared tasks such as RTE tasks of PASCAL (Dagan et al., 2006)(Giampiccolo et al., 2007), SemEval-2012 Cross-lingual Textual Entailment (CLTE) (Negri et al., 2012), NTCIR RITE tasks (Shima et al., 2011)(Watanabe et al., 2013)(Matsuyoshi et al., 2014), etc. In the third PASCAL RTE-3 task, contradiction relations are included in addition to entailment relations (Giampiccolo et al., 2007). In the RTE-6 task, given a corpus and a set of candidate sentences retrieved by a search engine from that corpus, systems are required to identify all the sentences from among the candidate sentences that entail a given hypothesis. NTCIR-9 RITE, NTCIR-10 RITE2, and NTCIR-11 RITEVal Exam Search tasks (Matsuyoshi et al., 2014) required participants to find an evidence in source documents and to answer a given proposition by yes or no. Research of RTE normally tries to employ logical processing.

As described above, question answering techniques could include logic, reasoning, syntactic and semantic analysis. Many previous related works tried to employ such deeper analyses. However, required techniques more or less differ depending on a target domain.

Another issue is whether the knowledge source needs to be "big data" or not. Regarding the COLIEE's legal problems, required knowledge source can be limited. In this paper, we suggest to use small data in a precise way, rather than to use enormous amount of data as knowledge source. Due to this small data issue, supervised machine learning methods would suffer from insufficient training data. In addition, there are no "similar" problems exist for most of the legal bar exam problems. Therefore, a solver needs to "comprehend" the contents of the knowledge sources. Moreover, it is difficult to analyze why the approaches using machine learning answer so, due to their black box architecture. Rule-based methods would make analyses less difficult, and are especially effective in a limited domain like legal documents.

Based on these thoughts, we built our yes/no question answering system. We aim to create an analyzable and human language processing bases system. We implemented a prototype of this system for the previous COLIEE 2016 task (Taniguchi & Kano, 2016). Although we shared the best score in Phase Two (Kim et al., 2016) (yes/no question answering given an evidence sentence), our system lacked precise analyses which human would do.

Our system does not employ any machine learning as its core. The main method of our system is clause-based analysis where predicates, case-roles, conditions, and negations are considered. We prepared a precise match, a loose match and a rough match. We layered these match modules in order to cover insufficiency of the current natural language technology/databases. We focused on Phase Two, yes/no answering question task of COLIEE 2017. Our system achieved score of the second best team in Phase Two.

We describe related works including datasets of NTCIR RITE challenge, and datasets of previous and this COLIEE tasks in Section 2. These datasets use the same format. Section 3 describes our design of the yes/no question answering system. Section 4 shows our experimental results for this COLIEE 2017 task. We discuss our achievements and limitations in Section 5, mentioning possible future works. We conclude our paper in Section 6.

2 Related Works

2.1 Exam Search Subtask in NTCIR RITEVal

While there were a couple of subtasks in the NTCIR RITE series, we describe the exam search subtask of NTCIR-11 RITEVal because the COLIEE dataset adopted the same format as the RITEVal dataset. RITEVal is an evaluation-based workshop held in 2013, aiming to recognize entailment, paraphrase, and contradiction between sentences, which is a common problem shared widely among researchers of NLP and information access (Dagan et al., 2005) (Giampiccolo et al., 2007).

The entrance exam subtask attempts to emulate human's process of answering entrance exam questions. A system solves multiple-choice questions of real university entrance exams by referring to textual knowledge such as Wikipedia and textbooks. The Entrance Exam subtask provides two types of evaluation challenges. In this paper, we treat the RITE-2 Search Style evaluation, whose explanation is given below. This style of subtask was called FV (Fact Validation) subtask in the RITEVal task. We refer to this RITE-2 Entrance Exam Search Style (ExamSearch) subtask simply as RITEVal in this paper. We only regard Japanese version of the subtask, while there were English and Chinese subtasks.

RITEVal's dataset was developed from the past Japanese National Center Test questions for University Admissions (Center Test). The Center Test asks students multiple-choice style questions.

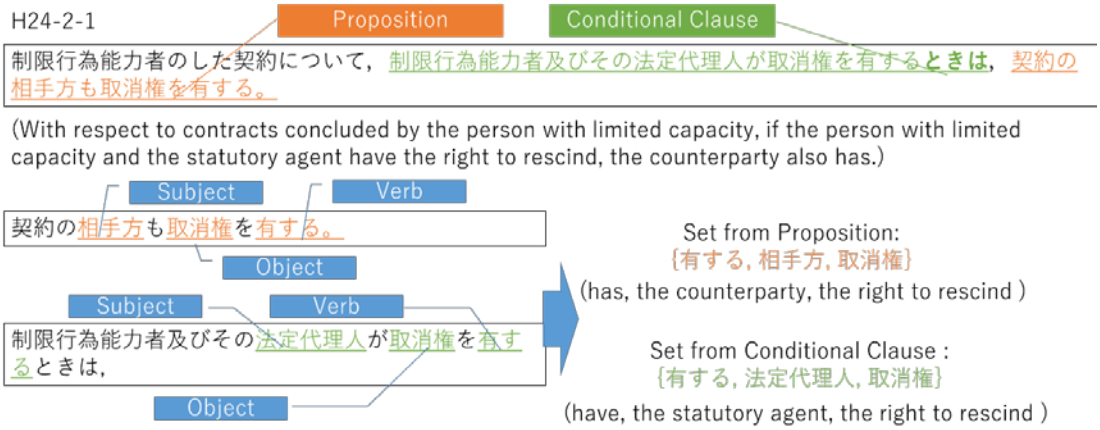

Fig. 2. A conceptual figure of our basic design with an example, showing condition clause, proposition clause, subject, object and predicate.

The RITEVal dataset consists of three types of questions, “select the correct choice” type, “select the wrong choice” type, and “combination” type.

In the RITEVal task, the original multiple-choices were not given as a whole, but given one by one. In “select the correct choice” type questions, given a choice, RITEVal participant systems are asked to return a confidence value for that choice. Evaluation is performed by comparing confidence values for each original multiple-choices, regarding the largest value as the participant system’s answer (smallest in case of “select wrong choice” type questions). In the “combination” type questions, the system is required to label Y or N for each choice and evaluated by a combination of these Y/N w.r.t the original multiple-choice question. In this paper, we focus on the “select correct/wrong choice” type questions.

2.2 JURISIN COLIEE datasets

The COLIEE (Competition on Legal Information Extraction and Entailment) shared task series were held in association with the JURISIN (Juris-informatics) workshops. The first one was the COLIEE 2014 shared task, and the second one was the COLIEE 2015 shared task (Kim et al., 2015). The previous one was the COLIEE 2016 shared task (Kim et al., 2016). This paper mainly describes our participation to the latest COLIEE 2017 shared task.

The COLIEE shared task consists of three phases until COLIEE 2016.

Phase One of this legal question answering task involves reading a legal bar exam question, and extracting a subset of Japanese Civil Code Articles.

Phase Two of the legal question answering task involves the identification of an entailment relationship. Given a question (t2) and a relevant article (t1), a participant’s system has to determine if the relevant articles entail the question or not by answering yes or no.

Phase Three is combination of Phase One and Phase Two. Phase Three requires both of the legal information retrieval system and textual entailment system. Given a set of legal yes/no questions, a participant’s system will retrieve relevant Civil Law articles. Then answer yes/no entailment relationship between input yes/no question and the retrieved articles.

The corpus of legal questions is drawn from Japanese Legal Bar exams, and the relevant Japanese Civil Law articles were also provided.

While there was an English translation version of the dataset provided, we only used the original Japanese version.

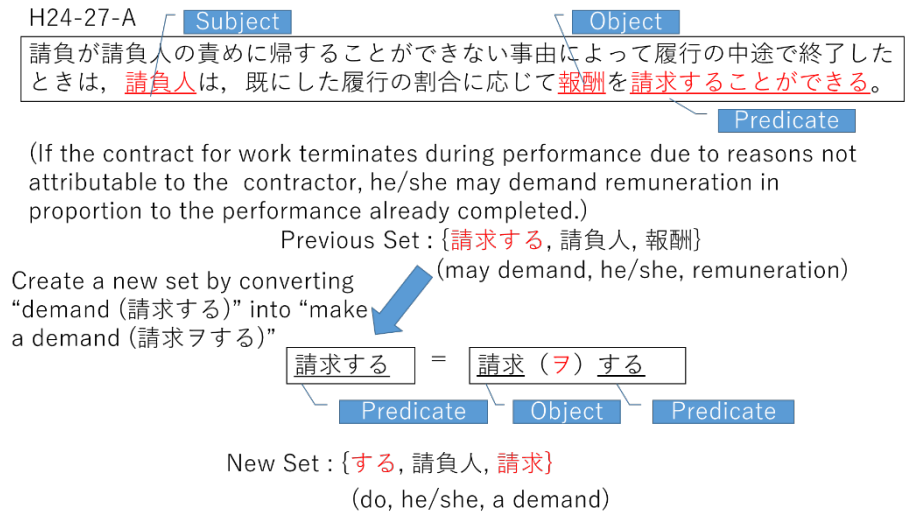

Fig. 3. An example converting an implicit predicate in a noun into an explicit predicate in a verb.

In COLIEE 2017, the original Phase Two was omitted and the original Phase Three is called Phase Two. We use the COLIEE 2017 numbering in this paper.

Fig. 1 is an example of the COLIEE dataset but can also be regarded as an example set of choices in the RITEVal dataset. This example shows the original Phase Two. Therefore, t1, an Civil Law article to be compared, is given. This t1 is not given in the original Phase Three and COLIEE 2017 Phase Two.

3 System Design and Method

We prepared three modules where each module individually outputs Yes/No. Our final output integrates outputs of these modules. We describe details of each module and their integration in this section.

3.1 Common Design Concept

We defined our own *clause* unit (“節”) in order to recognize *condition clauses* and main clauses precisely, which are included in a single sentence. After recognizing conditional clauses and main clauses in a sentence, we compare corresponding clauses between a given question and civil law articles.

A clause should include a predicate as a core element of that clause. We apply a dependency parser that makes chunks (“文節”) of a couple of morphemes. Starting form a chunk that includes a predicate, we aggregate neighboring chunks when a neighboring chunk does not include any predicate.

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

Fig.2 illustrates our common design described above.

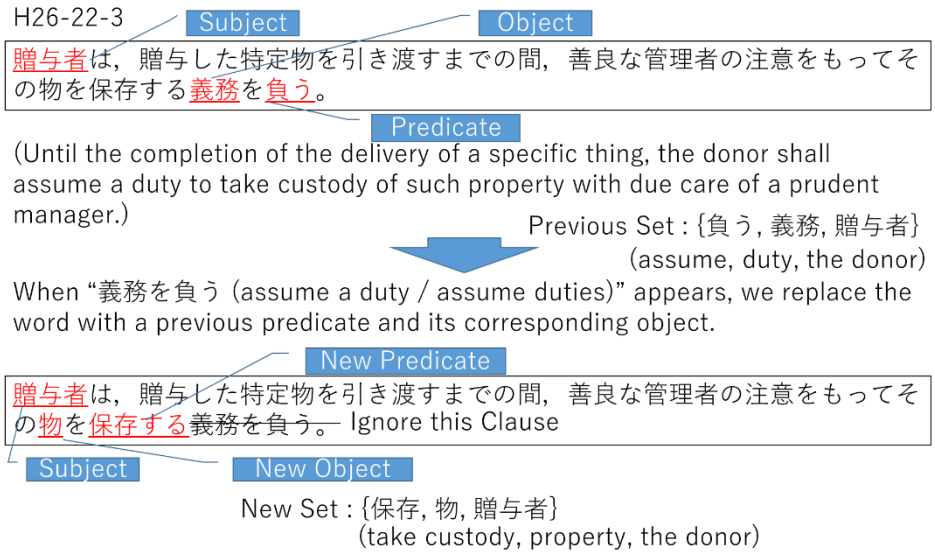

Fig. 4. An example ignoring ineffective expressions.

There are sometimes an implicit predicate which appear as a noun. We convert such an implicit predicate into a normal verb predicate. Fig.3 illustrates this conversion.

Another issue comparing predicates is that there are often ineffective predicates appear in legal documents. In the example of Fig. 4, “… shall assume a duty to …” is such an ineffective expression. We ignore such an expression, and take inner predicate for comparisons.

We define two types of special clauses: proposition clauses and condition clauses. We regard a clause as a condition clause when that clause includes specific patterns e.g. “when …”, “in case of …”, etc. In addition, we used a *force condition clause option* when deciding clause border. If this option is on, a chunk including the specific condition pattern or a previous chunk including a predicate will be forced to be a center of a clause.

3.2 Precise Match

In our precise match module, we extract civil law articles which proposition clauses match with a given problem’s proposition clauses. For each predicate, we find a subject and an object by their case markers. We perform exact matches for predicate itself, its subject and its object. When we could not find any subject nor any object, we skip that sentence. Predicates are compared in their base forms.

When the proposition clauses match in these criteria, we compare condition clauses if any exists. We use the same method for matching condition clauses, i.e. a pair of predicates, subjects and objects.

Finally, we determine a *yes/no* output for each problem; we output *yes* if everything matched, else output *no*. When there is any negation either in problem clause or in article clause, we reverse the *yes/no* output.

3.3 Loose Match

Loose match is a looser version of the precise match. When comparing proposition clauses and condition clauses, we output yes if either subjects or objects are match in addition to matching predicates.

3.4 Rough Match

Rough match is the loosest match in our modules. We only compare predicates of proposition clauses.

3.5 Module Integration

The precise match is the ideal method for us because the precise match tries to imitate human language processing to some extent. However, performance of current natural language processing tools and related databases are not enough to cover required document texts sufficiently, e.g. text structure, hierarchy of lexicon, etc. For this reason, we implemented the three matches as individual modules above, from strict to loose. When a stricter module cannot cover a given specific text, a looser module could augment the stricter module.

We prepared two ways for this integration.

In our first way, filtered integration, we try applying the precise match module first. When the precise match module cannot be applied, we apply the loose match module. If the loose match module cannot be applied as well, we try applying the rough match module.

In our second way, SVM integration, we used SVM (Support Vector Machine) to integrate the modules. Each module outputs a confidence value based on their match result. We trained SVM using these confidence values as training features. We dare designed to loosely integrate our modules rather than to put low level features directly, because we aim to construct an analyzable and human process based system.

4 Experiment and Result

Team ID	Accuracy	Language
iLis7	0.564103	English
iLis9-1	0.576923	English
iLis9-2	0.538462	Japanese
JAISTNLP2-2a-1a-norerank	0.512821	English
JAISTNLP2-2a-1b-rerank	0.474359	English
JAISTNLP2-2b-1a-norerank	0.487179	English
JAISTNLP2-2b-1b-rerank	0.500000	English
JNLP1-R	0.435897	English
JNLP1-RT	0.487179	English
KIS-YN-A	**0.538462**	Japanese
KIS-YN-CM	**0.538462**	Japanese
KIS-YN-CS	**0.589744**	Japanese
KIS-YN-M	**0.576923**	Japanese
KIS-YN-S	**0.653846**	Japanese
NAIST1	0.615385	Japanese

NAIST2	0.653846	Japanese
NAIST3	0.474359	Japanese
NOR17	0.538462	English
UA-LM	**0.717949**	Japanese
UA-TFIDF	0.692308	Japanese

Table 1. Results of COLIEE 2017 Phase Two formal run. Team IDs with prefix "KIS" are our results. Our results and the best team result are highlighted.

COLIEE 2017 provided a problem set of six years (504 problems) for training, one year for the formal run testing. In the *SVM integration*, we determined parameters of SVM by cross-fold validations of the given training data. Table 1 shows the results of COLIEE 2017 formal run for all teams, where prefix of KIS means our team results.

KIS-YN-CM and KIS-YN-CS uses the *force condition clause option.* KIS-YN-CM and KIS-YN-M use the *filtered integration*, while KIS-YN-S and KIS-YN-CS use the *SVM integration*. Among these options, KIS-YN-S obtained the best score.

As far as we observe in this result, the *force condition clause option* was not effective, showing several points decrease. The *SVM integration* was effective, showing 5-8 points increase.

5 Discussion and Future Work

As the number of formal run submissions are limited, we need analysis on training data as follows.

The precise match module could be applied to around 30% of the problems, while the loose match module could be applied to around 50% of the problems. These observations are just same as we assumed beforehand. We suffer from many missing elements when perform-ing linguistic analyses.

For example, there are many omissions e.g. subjects or objects, especially in case of Japanese expressions. Predicate-argument structures are sometimes implicit and difficult to extract.

Semantic hierarchy of predicates is another important issue unresolved. We need to regard different expressions as same meaning from the yes/no answering point of view. However, it depends on broad context whether a different expressions could be regarded as same mean-ing or not.

We observed in the training data that the confidence values contributed to the evaluation scores. More fine-tuned confidence values might increase the scores. However, humans can normally decide their answers in deterministic ways for such legal domain problems. As we suggested in our system design, current NLP technology performance and database coverage are still far insufficient for a system to work like humans. Although this is true, our result is competitive with other teams who used machine learning as their cores. More precise approach would be meaningful as an extension of our system.

We need to grasp distribution of the problems in order to interpret the results objectively.

We observed several points of fluctuations of evaluation scores between years of problems. Very roughly speaking, the COLIEE dataset includes two types of problems: very easy and very difficult.

The very easy problems have almost same text strings in the Civil Law articles. Such a problem can be solved by superficial word-level methods. For example, we found 6 very easy problems in 77 problems (H24 dataset) by our manual verification. Because 6/77 equals 7.8 points and the random baseline is around 50 points, it would be easily available to achieve around 60 points by any relevant method using string/word based features.

The very difficult problems are really hard to solve, they may include logic, abstraction, complex syntactic structure, etc. We do not believe that any current system could handle such issues in

sufficient performance. Rather it might have captured tendency of problem writers. Our aim was to solve problems in the middle of very difficult and very easy ones. While this aim was achieved to some extent, there are still many important issues remaining unresolved. We would need structured lexical database with the semantic hierarchy, at least for this legal domain.

Because our system is designed in an analyzable way, further analyses are available to find what sort of methods/features were effective. Such a deeper and detailed analyses are required future work.

6 Conclusion

We proposed an analyzable and human language process based legal yes/no question answering system. Our team was second best (0.6538) in the COLIEE 2017 yes/no question task (Phase 2). We aimed to solve problems of middle level difficulty by linguistic analyses. This aim was achieved to some extent, while leaving several important issues, such as lexical semantic ontology, unresolved for future work.

Acknowledgements

This research was partially supported by MEXT Kakenhi and JST CREST.

References

Arai, N. H. (2015). The impact of AI—can a robot get into the University of Tokyo? *National Science Review*, *2*(2), 135–136. doi:10.1093/nsr/nwv011

Bian, J., Liu, Y., Agichtein, E., & Zha, H. (2008). Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 467–476). inproceedings, New York, NY, USA: ACM. doi:10.1145/1367497.1367561

Braschler, M. (2001). CLEF 2000 - Overview of Results. In *Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation* (pp. 89–101). inproceedings, London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=648263.753369

Cui, H., Sun, R., Li, K., Kan, M.-Y., & Chua, T.-S. (2005). Question Answering Passage Retrieval Using Dependency Relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 400–407). inproceedings, New York, NY, USA: ACM. doi:10.1145/1076034.1076103

Dagan, I., Glickman, O., & Magnini, B. (2005). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Dagan, I., Glickman, O., & Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment* (pp. 177–190). inproceedings, Berlin, Heidelberg: Springer-Verlag. doi:10.1007/11736790_9

Dumais, S., Banko, M., Brill, E., Lin, J., & Ng, A. (2002). Web Question Answering: Is More Always Better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research*

and Development in Information Retrieval (pp. 291–298). inproceedings, New York, NY, USA: ACM. doi:10.1145/564376.564428

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., … Yates, A. (2004). Web-scale Information Extraction in Knowitall: (Preliminary Results). In *Proceedings of the 13th International Conference on World Wide Web* (pp. 100–110). inproceedings, New York, NY, USA: ACM. doi:10.1145/988672.988687

Ferrucci, D. (2012). Introduction to "This is Watson." *IBM Journal of Research and Development*, *56*(3.4), 1:1-1:15. doi:10.1147/JRD.2012.2184356

Ferrucci, D. A. (2011). IBM's Watson/DeepQA. *SIGARCH Comput. Archit. News*, *39*(3). Journal Article. doi:10.1145/2024723.2019525

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (pp. 1–9). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1654536.1654538

Jeon, J., Croft, W. B., & Lee, J. H. (2005). Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (pp. 84–90). inproceedings, New York, NY, USA: ACM. doi:10.1145/1099554.1099572

Kanayama, H., Miyao, Y., & Prager, J. (2012). Answering Yes/No Questions via Question Inversion. In *the 24th International Conference on Computational Linguistics (COLING 2012)* (pp. 1377–1391). Mumbai, India.

Kando, N., Kuriyama, K., & Nozue, T. (1999). NACSIS Test Collection Workshop (NTCIR-1) (Poster Abstract). In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 299–300). inproceedings, New York, NY, USA: ACM. doi:10.1145/312624.312730

Kim, M.-Y., Goebel, R., Kano, Y., & Satoh, K. (2016). COLIEE-2016: Evaluation of the Competition on Legal Information Extraction/Entailment. In *Tenth International Workshop on Juris-informatics (JURISIN 2016)*.

Kim, M.-Y., Goebel, R., & Ken, S. (2015). COLIEE-2015: Evaluation of Legal Question Answering. In *Ninth International Workshop on Juris-informatics (JURISIN 2015)*. Keio University, Yokohama, Japan.

Kwok, C. C. T., Etzioni, O., & Weld, D. S. (2001). Scaling Question Answering to the Web. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 150–161). inproceedings, New York, NY, USA: ACM. doi:10.1145/371920.371973

Lin, D., & Pantel, P. (2001). Discovery of Inference Rules for Question-answering. *Nat. Lang. Eng.*, *7*(4), 343–360. article. doi:10.1017/S1351324901002765

Matsuyoshi, S., Miyao, Y., Shibata, T., Lin, C.-J., Shih, C.-W., Watanabe, Y., & Mitamura, T. (2014). Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task. In *the 11th NTCIR (NII Testbeds and Community for information access Research) workshop* (pp. 223–232). inproceedings.

Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., & Giampiccolo, D. (2012). Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation* (pp. 399–407). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2387636.2387700

Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and*

development in informaion retrieval (pp. 235–242). New York, NY, USA: ACM. doi:10.1145/860435.860479

Ravichandran, D., & Hovy, E. (2002). Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 41–47). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1073083.1073092

Shima, H., Kanayama, H., Lee, C., Lin, C., Mitamura, T., Miyao, Y., … Takeda, K. (2011). Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In *NTCIR-9 Workshop* (pp. 291–301). inproceedings.

Taniguchi, R., & Kano, Y. (2016). Legal Yes/No Question Answering System using Case-Role Analysis. In *Competition on Legal Information Extraction/Entailment (COLIEE 2016), Tenth International Workshop on Juris-informatics (JURISIN 2016)*. Yokohama, Japan.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press.

Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., … Takeda, K. (2013). Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In *the NTCIR-10 Workshop* (pp. 385–404). Tokyo, Japan.

Xue, X., Jeon, J., & Croft, W. B. (2008). Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 475–482). New York, NY, USA: ACM. doi:10.1145/1390334.1390416

Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 129–136). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1119355.1119372

EPiC Series in Computing

Volume 47, 2017, Pages 68–78

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Legal Information Retrieval Using Topic Clustering and Neural Networks

Rohan Nanda[1], Adebayo Kolawole John[1], Luigi Di Caro[1], Guido Boella[1], and Livio Robaldo[2]

[1] University of Turin, Italy
nanda@di.unito.it, kolawolejohn.adebayo@unibo.it, {dicaro,guido}@di.unito.it
[2] University of Luxembourg
livio.robaldo@uni.lu

Abstract

This paper presents a description about our adopted approach for the information retrieval and textual entailment tasks of the COLIEE 2017 competition. We address the information retrieval task by implementing a partial string matching and a topic clustering method. For the textual entailment task, we propose a Long Short-Term Memory (LSTM) - Convolutional Neural Network (CNN) model which utilizes word embeddings trained on the Google News vectors. We evaluated our approach for both tasks on the COLIEE 2017 dataset. The results demonstrate that the topic clustering method outperformed the partial string matching method in the information retrieval task. The performance of LSTM-CNN model was competitive with other textual entailment systems.

1 Introduction

The COLIEE (Competition on Legal Information Extraction/Entailment) 2017 comprises two tasks : information retrieval and textual entailment. The information retrieval task deals with identifying a set of Civil Code articles for a given question or query. The information retrieval system takes a query Q as input and retrieves a set of articles (from the entire Civil Code) relevant to the query. The textual entailment task deals with recognizing textual entailment between the query and the retrieved articles. The objective is to determine if the article confirms "yes" or "no" as an answer to the query.

In this paper[1], we address both tasks of information retrieval and textual entailment. In the next section, we discuss our approach to the information retrieval task. Section 3 describes our approach for the textual entailment task. Section 4 presents the results and analysis. The paper concludes in Section 5.

2 Task 1: Information Retrieval

The objective of information retrieval task is to retrieve a set of articles which are relevant for the input query. Text similarity techniques are an integral part of information retrieval system. Lexical

[1]The first and second author contributed equally

similarity techniques use string-based algorithms for measuring the similarity between two text strings [9]. However, they don't take into account the ambiguities of natural language. Semantic similarity methods comprise knowledge-based and corpus-based techniques. Corpus-based techniques measure the degree of similarity between texts by learning information from a large corpora. Latent semantic analysis (LSA) is one such method which is based on the assumption that semantically similar words occur in similar pieces of text [19]. Knowledge-based techniques compute the degree of similarity between texts by utilizing the knowledge from semantic networks. Topic-based models assign topics to each document where each topic is represented by a term distribution [1]. Semantically similar documents are assigned similar topics. Recently, new corpus-based techniques like word embeddings have been utilized to compute semantic similarity for information retrieval systems [16]. They utilize a neural network language model to predict a word from its surrounding words (continuous bag-of-words model) or predict several surrounding words from an input word (Skip-gram model). These models have shown promising results when trained on a large corpus. However, the COLIEE corpus is quite small as it contains short text articles instead of documents. The total number of articles in the corpus are around 1000. Therefore we tailor our approach for the COLIEE dataset.

The manual analysis of the corpus suggested that in many cases, the length of queries was much shorter than the length of the corresponding articles. The query in some cases only partially expressed the semantics of the article. Also the query and corresponding article had few similar words or phrases in common. Therefore, as our first approach for Task 1 we decided to use a partial string matching algorithm. In this approach, each query is compared with all the articles in the corpus to retrieve the most similar article. The texts of the query Q and article A are first tokenized. Each group of tokens in Q and A is considered as a set [24]. Then the intersection set, I of sorted tokens in set Q and set A is derived as follows:

$$I = Q \cap A \tag{1}$$

The Query set Q is represented as the union of the tokens in the intersection set I and the remaining tokens in the remainder query set R_Q.

$$Q = I \cup R_Q \tag{2}$$

The article set A can be represented as the union of the intersection set I and the remainder article set R_A.

$$A = I \cup R_A \tag{3}$$

Further, we compute three similarity measures: (I,Q), (I,A) and (Q,A). The similarity measure PS between two sets is computed as $2.0 * M/T$, where T is the total number of elements in both sets and M is the number of matches [24]. The maximum similarity value of the three is considered as the partial string similarity. The similarity is in the range of [0,1]. The major significance of this method is the intersection set I. The query set Q and article set A have a high similarity value when set I is the larger part of either set A or Q.

The first approach makes a naive assumption that *relevance* is a measure of the number of common tokens between an article and a query. This assumption is not totally valid, considering that synonyms and polysemous words appear frequently in natural language texts. Our second approach takes care of the naivety by tackling relevance as a measure of semantic similarity between an article and the query. Usually, instead of computing this similarity between each query and all the articles present, we group articles into clusters of topic such that each article has a topic vector which is compared to that of the query. For each query, we only focus on the articles with similar topic vectors. The top-n most similar articles to the query are then passed through the semantic similarity module. This overcomes the limitation of the first approach in two ways. First, by adopting an initial *topic-based* clustering solution, we overcome the bias of tokens that have less importance in the articles as well as language variability issues like synonymy and polysemy. Also, we limit the relevancy search to a subset of 'supposedly' relevant articles. We call this set of articles the *seed* set. Secondly, by computing the semantic similarity between the query and articles in its seed set, we limit the over-dependence on language units like words and their occurrence counts, while considering the interplay between the words in both the query and article.

2.1 Clustering With Topic Models

The first step in our second approach is to cluster articles into their respective topics, thus yielding a set of low dimensional topic vectors for each article. This can easily be done with topic modeling of some documents. Topic Models (TM) are a suite of algorithm that unearth the latent topic distribution in a set of documents, based on the assumption that a document is a mixture of topics. Popular topic models are the Latent Semantic analysis (LSA) [7] and the Latent Dirichlet Allocation (LDA) [5]. We used LDA for this part of our work. LDA is a generative probabilistic model with the intuition that a document is a random distribution over latent topics, where each topic is characterized by a distribution over words in the vocabulary. The model has shown capability to capture semantic information from documents in a way similar to probabilistic latent semantic analysis [11] such that a low dimensionality representation of texts is produced in the semantic space while preserving their latent statistical features.

Formally, given a document $\mathbf{w}$ of N words such that $\mathbf{w} = (\mathrm{w}_1,\mathrm{w}_2,\mathrm{w}_3,...\mathrm{w}_N)$ and a corpus D of M documents denoted by $D = (\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3.....\mathbf{w}_M)$. For each of the words w_n in the document, a topic z_n is drawn from the topic distribution θ, and a word w_n is randomly chosen from $\mathrm{P}(\mathrm{w}_n \mid \mathrm{z}_n, \beta)$ conditioned on z_n. Given α, a k-vector with components with $\alpha_i > 0$ and the Gamma function $\Gamma(\mathrm{x})$. The probability density of the Dirichlet is given as

$$P(\Theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\Pi_{i=1}^{k}\Gamma(\alpha_i)}\Theta_1^{\alpha_1-1}....\Theta_k^{\alpha_k-1} \tag{4}$$

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is thus given by

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(\theta|\alpha)\Pi_{n=1}^{N}P(z_n|\theta)P(w_n|z_n, \beta) \tag{5}$$

Integrating over θ and summing of z, the set of topic assignments, the distribution of a document can be obtained as shown in equation (6) below

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) \left(\Pi_{n=1}^{N}\sum_{z_n} P(z_n|\theta)P(w_n|z_n, \beta)\right) \mathrm{d}\theta \tag{6}$$

where $\mathrm{P}(\mathrm{z}_n \mid \theta)$ is θ_i for the unique i such that $z_n^i = 1$ The probability of a corpus is obtained through the product of marginal probability above for each $\mathbf{w}_n$ in D as given in the equation (7) below:

$$P(\mathbf{w}|\alpha, \beta) = \{\Pi_{d=1}^{M} \int P(\theta_d|\alpha)\Big(\Pi_{n=1}^{N_d} \sum_{z_d n} P(z_d n|\theta_d)P(w_d n|z_d n, \beta)\Big) \mathrm{d}\theta_d\} \tag{7}$$

Training LDA model on a corpus requires feeding the model with sets of tokens from the document. The model statistically estimates the topic distribution θ_d for each document as well as the word distribution in each topic. A model can also be used to predict topic classes for a previously unseen document. More information about LDA can be found in [5, 4]. In order to train LDA algorithm, we used all the articles in the training set as well the queries, taking pair of article-query as a document. We used known relevant pairings of query-article where available and for every other case, we feed each of the query or article as independent document. Furthermore, we include extra articles which formed parts of the documents released for COLIEE 2017 training set. We use the Gensim[2] implementation of LDA with Gibbs sampling, with T = 25 topics and the number of inference also being 20. All the documents

[2]Gensim is a python library offering a suite of algorithms for text processing. Available at https://radimrehurek.com/gensim/

were stemmed and stopwords removed. Each document was weighted with term frequency-inverse document frequency (tfidf) values in order to overcome the bias against unimportant but frequent words. Even though the number of topics was intuitively chosen, we observed that setting the number of topics to be smaller makes the topic distribution to be dense such that most articles have similar topical distribution. Also, when we increased the topic number, we identified a lot of arbitrariness and imprecision in topic distribution. We assumed that this observation is probably due to the tiny size of the training data used.

We use the resulting LDA model to generate topic distribution for each training articles such that the set of topic IDs are transformed into a dense topic vector of 25-dimension. First, we obtain a matrix $G = L \times T$ where $l \in L$ is a sparse vector of length T, the chosen number of topics. Each vector l contains the distribution of the topic IDs assigned by LDA to each article, where by topic ID, we denote the topic group or cluster that an article belongs, i.e., a number in the range $[0,\ T-1]$. The procedure is also repeated for the queries. The resulting outputs are two matrices transformed to dense, i.e. article and query, all of the same dimension. We used Faiss[3] to index these matrices. Faiss implements an array of search algorithms for search and clustering of dense vectors of documents and queries. These algorithms include exact match, L2, dot product vector comparison, nearest neighour, k-means, cosine similarity etc.

For each query, we pick the top 5 matching articles. Now, these articles are not assumed to be the exact relevant ones but rather selected as the seed set and input to the semantic similarity module. The semantic similarity module computes a similarity score between the text of each of the documents in the seed set and the query. Specifically, this module takes account of semantic information and word order information in the text of both the query and the article. Furthermore, it uses WordNet as a lexical taxonomy for looking up synonyms of words while calculating similarity. The similarity function between two words *w1* and *w2* is defined as the product of the depth function, *f1(h)* and length function, *f2(l)* as follows [13]:

$$S(w1, w2) = f1(h).f2(l) \tag{8}$$

The length function *f2(l)* is a monotonically decreasing function of path length l. It includes a constant alpha.

$$f2(l) = e^{-\alpha.l} \tag{9}$$

The depth function is a monotonically increasing function of depth h in concept hierarchy. It is represented as follows:

$$f1(h) = \frac{e^{\beta.h} - e^{-\beta.h}}{e^{\beta.h} + e^{-\beta.h}} \tag{10}$$

The values of alpha and beta are set to 0.2 and 0.45 respectively as per Li et al. [20].

Once the similarity scores have been computed, we sort the articles according to the similarity scores from highest to lowest. We pick the article with the highest similarity score along with any other that has a distance of not more than 15% to the topmost similar article, i.e., the one with the highest similarity score from the semantic similarity module. We did not evaluate, how the choice of the chosen parameters (threshold and top-5) impacts the result and such analysis is planned for future work. The distance score, *Dist*, is computed according to the formula in equation (11).

$$Dist = \Big(\frac{Hsim - aSim}{Hsim}\Big) \times 100 \tag{11}$$

where *Hsim* is the similarity score (highest of the top-5) of the most similar articles according to the similarity module and *aSim* is a similarity score of any of the other articles in the top-5.

3 Task2: Textual Entailment

The second task of COLIEE 2017 involves determining whether the retrieved articles in the first task entail the corresponding queries or not. This task is referred to as Textual Entailment (TE) and its goal

[3]Faiss is available at https://github.com/facebookresearch/faiss.

is to identify entailment relationship between a text and an hypothesis [3]. A premise or text entails an hypothesis (which is another text) if the hypothesis can be inferred from the premise. TE is a typical classification task, either as a binary (YES/NO) or multi-class classification (YES/NO/NEUTRAL etc.). The organizers of COLIEE2017 proposed a binary classification task.

Early TE systems rely on the use of hand crafted features [8]. In addition, a lot of knowledge resources, e.g., Wordnet are often employed in developing these systems [22]. The features often extracted from the training text include simple ones like n-gram overlap, string matching, presence of negation words e.g., never, shouldn't etc. Also, information retrieval (IR)-inspired features such as the TFIDF and a measure of similarity between the text and the hypothesis obtained with metrics such as the cosine similarity etc, are often used [12].

Even though a number of the systems which rely on feature engineering have achieved relative success, the efforts required in engineering features and feature selection pales this success. An evaluation of systems using different techniques for solving TE tasks is detailed in [3].

Since Neural Networks (NNs) have continued to give excellent performance in language modeling tasks [15], researchers have employed some NNs models for TE tasks as well. Specifically, the listed papers [6, 23, 2] show that deep Neural architecture can match and outperform lexical classifiers which use hand-crafted features for TE task. While these systems are very promising, they enjoy the benefit of a reasonable large amount of training data. Therefore, as we will show later in the results section, the performance reported in this paper may not be competitive compared to these works, given the very tiny training set available in this particular COLIEE challenge.

The authors in [17] utilized a Convolutional Neural Networks (CNN) to model entailment relationship on a previous COLIEE competition dataset. Also, the work described in [14] which was also based on a past COLIEE competition employed the use of Long Short-Term Memory (LSTM) for modeling entailment relationship.

We utilize NNs to autonomously learn latent enatailment features from a pair of text and hypothesis. Our NNs model combines LSTM [10] with CNN [18]. LSTM Networks are a powerful type of recurrent neural networks (RNNs) and are robust to the vanishing gradient problem [10]. Also, LSTMs generally have enhanced memory since they are able to retain information over many time steps. CNNs on the other hand can capture multiple features between adjacent input, owing to the many convolution it uses to filter the input sequences. Each filter transforms a local patch of lower-level features into higher-level representation [18, 15]. The LSTM state transition is given in equation (12) below.

$$
\begin{aligned}
i_t &= \sigma\Big(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\Big),\\
f_t &= \sigma\Big(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\Big),\\
o_t &= \sigma\Big(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\Big),\\
u_t &= \tanh\Big(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\Big),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh c_t
\end{aligned}
\tag{12}
$$

Figure 1 gives a pictorial illustration of the model used in this work. Our model is quite simple, and it is based on sentence encoding approach. This was intentional since we have a very small training set. Also, even when we tried more complex model architectures, we keep getting similar performances.

First, at every input point, the sequences of both the article and the query sentences are each embedded into a 300 dimensional vectors. Instead of learning embeddings from scratch, we used the *GoogleNews* vectors for this part of our work. The *GoogleNews* vectors is a result of training the word2vec algorithm [21] on over 3 billion words. Also, we keep the weights of the embeddings frozen throughout all the layers. Note that the encoded articles and queries have the same sequence length, as they have been zero-padded in order to equal the maximum sequence length (*774*) in the input set. For each data sample, the input sequences are tokens of the corresponding articles and queries. At each time step, the vectors of a word from an input sequence is obtained. We associate each token t

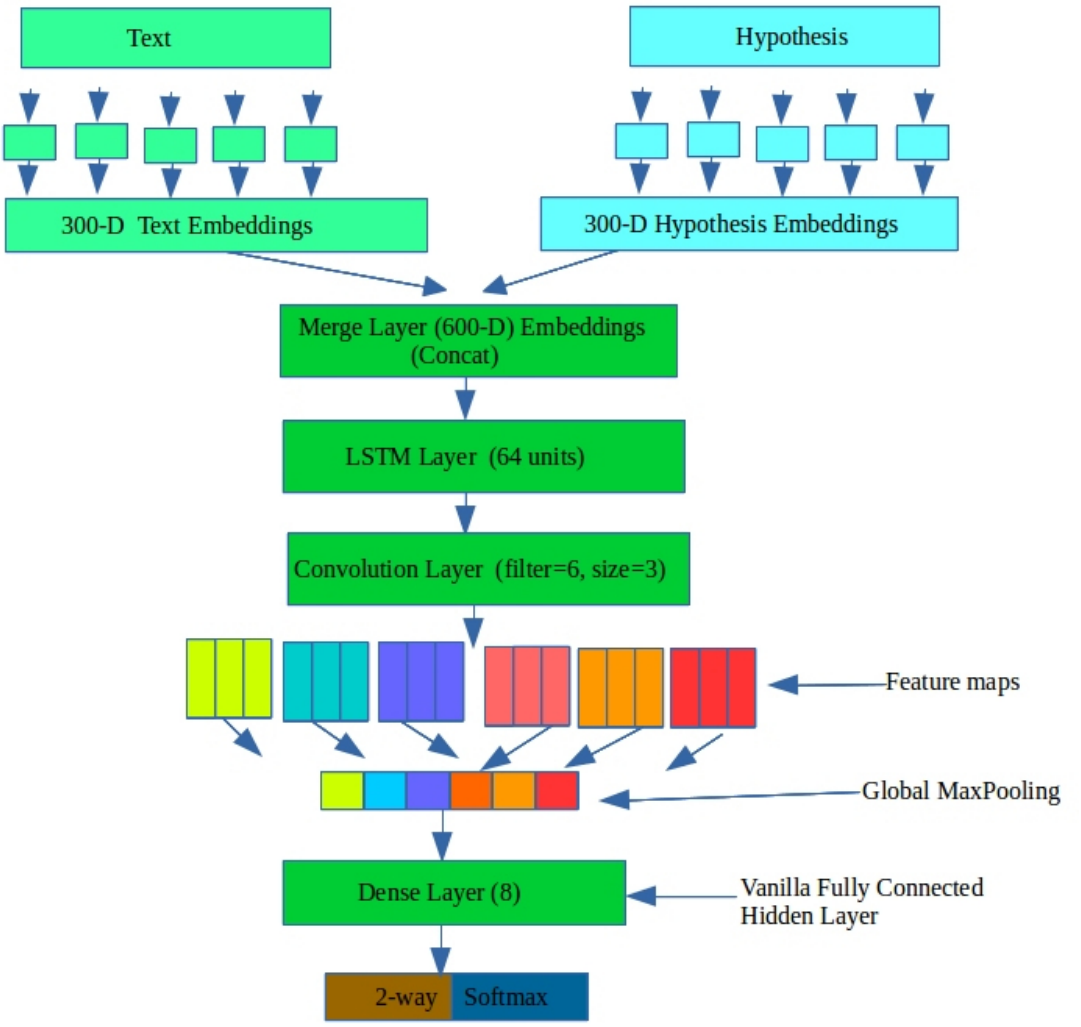

Figure 1: A pictorial illustration of the LSTM-CNN model

in our vocabulary with a vector representation $x_t \in \mathbb{R}^{d=300}$. We generate a sentential representation by performing an element wise sum of each x_t in order to get the sentence embedding for each input sequence, i.e., both for the articles and the queries. We normalize the resulting vector sum by the length of the sequence such that

$$s_i = \frac{1}{|n|} \sum_{t=1}^{|n|} x_t, \quad s_i \in \mathbb{R}^{d=300} \tag{13}$$

Where s_i in equation (13) denotes the embedding representation of each data point.

The encoded sequences are then passed into a Merge layer where the encoded article and query are concatenated. This layer yields a 600 dimensional vector which we pass to the LSTM. The activation function for this layer, like in every other hidden layers of our network is the rectified linear unit (RELU). The LSTM has 64 units, and the representation from the hidden state of the last cell is propagated to a convolution layer. Our convolution layer makes use of 6 filters, each with a region size

= 3. A global maxpooling layer selects the prominent features from the feature maps. We further add a fully connected multilayer perceptron networks (mlp) with 8 hidden layers. We use the softmax to distribute the probability distribution between the +1 or -1 class, where +1 signifies entailment and -1 means Non-entailment. For each input s_j, given the hidden state h_j from the last fully-connected mlp layers, the softmax classifier predicts the label $\hat{y}_j$ as shown in the equation below:

$$\widehat{p}\theta(y|s_j) = softmax\Big(W^{(p)}h_j + b^{(p)}\Big),$$

$$\widehat{y_j} = \arg\max_y \widehat{p}\theta(y|s_j) \tag{14}$$

The cost function is given in equation (15) below:

$$J(\theta) = -\frac{1}{m}\sum_{k=1}^{m}(y^{(k)}|s^{(k)}) + \frac{\lambda}{2}||\theta||_2^2 \tag{15}$$

where m is the number of data samples, in this case the batch size for each iteration and λ is an L2 norm regularization hyperparameter.

For implementation, we used Keras[4] Deep Learning library. Since the training set is rather tiny, we used a batch size of 16. We used *adam*, a stochastic optimizer with learning rate set at 0.01 and a decay value of 1e-4. Throughout the network, we applied a uniform *Dropout* value of 0.2 after each of the LSTM, convolution and the fully connected layers. Initially, the model was trained on the input data for *10* epochs but this was subsequently reduced to *5* since we discovered that training keep exiting at epoch *8* when we used *earlystopping* with patience for unimproved validation loss set to *3*.

4 Results and Analysis

In this section, we discuss the results of both task1 and task2 using the techniques discussed in the previous sections. We evaluate the results on the test set provided by the organizers. We observed from the dataset that there were many cases when multiple articles are retrieved for a single query. Therefore, there was a need to consider the cases where the articles retrieved by our system are a subset of the articles in the gold standard. Hence, we carried out two evaluations for the information retrieval system : strict and lenient. In strict evaluation, only exact matches of the articles retrieved by our system with the gold standard were considered as true positive. In lenient evaluation, a partial match of the retrieved articles with the gold standard set of articles is also considered as a true positive. In other words, if the system retrieves a subset of the articles present in the gold standard then it is considered a match. Table 1 shows the results of task1 for both partial string matching and topic clustering method.

	Partial String Matching (strict evaluation)	Partial String Matching (lenient evaluation)	Topic Clustering (strict evaluation)	Topic Clustering (lenient evaluation)
Precision	0.413	0.645	0.441	0.69
Recall	0.445	0.628	0.49	0.692
F-score	0.428	0.636	0.464	0.691

Table 1: Results for Task 1 (information retrieval)

The results indicate that topic clustering method achieved a higher F-score than the partial string matching method for both strict and lenient evaluation. This is because topic clustering not only relies

[4]Keras is a modular but powerful Deep Learning Library built on top of Theano and Tensorflow. Available at https://github.com/fchollet/keras

on the occurrence of common tokens between article and query but also takes into account the semantics by incorporating topics and a knowledge-based resource (WordNet). We present an example in Figures 2 and 3 to illustrate this. The article retrieved by the partial string matching method captures only a part of the meaning of the query. The query explicitly talks about the conditions in which the contract shall be formed. However, the article retrieved by the partial string matching method talks about some conditions (necessary to form the contract) mentioned in the query but does not mention anything about forming the contract. It captures only the partial meaning of the query. On the other hand, the article retrieved by the topic clustering method mentions both the conditions and forming of the contract. Thus, topic clustering method was successful to capture the semantics of the text to a greater extent as compared to the partial string matching method. The topic clustering method also achieves a higher precision and recall than the partial string matching method. However, it is also important to mention some limitations while using topic clusters for short text similarity. LDA considers the text as a mixture of latent topics. Each topic is a mixture of words. The topics generated in query and article are quite similar when query and article comprise similar words. However, in cases when most words are different in query and article, the topics of the articles (grouped in the cluster) are sometimes not relevant to the topics of the query. We tried to address this limitation to some extent by incorporating synonyms from WordNet. However, even minor variation in the usage of wordings in short texts can lead to different topic distributions. Generally, topic clustering methods have been successful in information retrieval for large corpora which comprise documents of significant length. However, the COLIEE dataset consists of civil codes which are short legal texts.

Query	Article retrieved by Topic Clustering
In the case where (A) loses his/her capacity to act after the dispatch of the notice of acceptance of a contract to (B), who is a person at a distance, even if (B) knows such fact, the contract shall be formed.	(1) A contract between persons at a distance shall be formed upon dispatch of the notice of acceptance. (2) In cases where no notice of acceptance is required due to the offeror's manifestation of intention or usage of trade, the contract shall be formed upon the occurrence of any fact which ought to be regarded as a manifestation of intention of acceptance.

Figure 2: Article retrieved by topic clustering method for a given query

Query	Article retrieved by Partial String Matching
In the case where (A) loses his/her capacity to act after the dispatch of the notice of acceptance of a contract to (B), who is a person at a distance, even if (B) knows such fact, the contract shall be formed.	(1) Manifestation of intention to a person at a distance shall become effective at the time of the arrival of the notice to the other party. (2) The validity of manifestation of intention to a person at a distance shall not be impaired even if the person who made the manifestation dies or loses his/her capacity to act after the dispatch of the notice.

Figure 3: Article retrieved by partial string matching for a given query

Table 2 shows the results of the textual entailment task. We compare our proposed LSTM-CNN model with the best runs of the other teams for the English task. The results show that our model achieves comparable performance with other textual entailment methods. The slightly lower accuracy

Method	Accuracy
JAISTNLP	0.512
JNLP	0.487
iLis	0.576
LSTM - CNN Model	0.538

Table 2: Accuracy for Textual Entailment Task

of our model is probably due to the trained word embeddings on the Google News dataset (which is not tailored for legal text). The results of task 2 indicate that our textual entailment system did not generalize well on the test data. One possible reason for this is the small size of training data in our case. Neural networks have been known to perform well with a large amount of training data. The motivation to use neural networks was to avoid handcrafting features given the legislative nature of the text. Another possible reason for the mediocre performance of the textual entailment system is error propagation due to the evaluation style. This is because the performance of the second system (task 2) is dependent on the results of the first system (task 1). In task 2, we are checking for entailment relationship between each retrieved article and the corresponding query. In the case when the system retrieves wrong articles, the entailment relationship is judged wrongly.

5 Conclusion

This paper presented our team's approach for the COLIEE 2017 competition. For the information retrieval task, we utilized a partial string matching and a topic clustering method. The lexical approach of partial string matching was complemented by a semantic similarity approach of topic clustering while also incorporating the knowledge from WordNet. The results indicate that the topic clustering approach outperformed the partial string matching method. Further, we proposed a LSTM-CNN model for the textual entailment task. We utilized word embeddings from the Google News corpus. The LSTM-CNN model had a comparable performance with other textual entailment systems.

6 Acknowledgments

Research presented in this paper is conducted as a PhD research at the University of Turin, within the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science and Technology. Luigi Di Caro and Livio Robaldo have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

[1] Leif Azzopardi, Mark Girolami, and CJ Van Rijsbergen. Topic based language models for ad hoc information retrieval. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 3281–3286. IEEE, 2004.

[2] Petr Baudiš and Jan Šedivỳ. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*, 2016.

[3] Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. The seventh pascal recognizing textual entailment challenge. *Proceedings of TAC*, 2011, 2011.

[4] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[6] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[7] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[8] Miguel Angel Ríos Gaona, Alexander Gelbukh, and Sivaji Bandyopadhyay. Recognizing textual entailment using a machine learning approach. In *Mexican International Conference on Artificial Intelligence*, pages 177–185. Springer, 2010.

[9] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13), 2013.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[12] Sergio Jimenez, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal. Unal-nlp: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, 2014.

[13] Adebayo Kolawole John, Luigi Di Caro, and Guido Boella. Normas at semeval-2016 task 1: Semsim: A multi-feature approach to semantic text similarity. *Proceedings of SemEval*, pages 718–725, 2016.

[14] Adebayo Kolawole John, Luigi Di Caro, Guido Boella, and Cesare Bartolini. An approach to information retrieval and question answering in the legal domain.

[15] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[16] Tom Kenter and Maarten de Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM, 2015.

[17] Mi-Young Kim, Ying Xu, and Randy Goebel. A convolutional neural network in legal question answering.

[18] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[19] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[20] Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1138–1150, 2006.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] Shachar Mirkin, Ido Dagan, and Eyal Shnarch. Evaluating the inferential utility of lexical-semantic resources. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 558–566. Association for Computational Linguistics, 2009.

[23] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

[24] Jiannan Wang, Guoliang Li, and Jianhua Fe. Fast-join: An efficient method for fuzzy token matching based string similarity join. In *Data Engineering (ICDE), 2011 IEEE 27th International*

Conference on, pages 458–469. IEEE, 2011.

EPiC Series in Computing

Volume 47, 2017, Pages 79–89

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Legal Question Answering System using Neural Attention

Ayaka Morimoto[1], Daiki Kubo[2], Motoki Sato[3], Hiroyuki Shindo[4], and Yuji Matsumoto[5]

[1] Graduate School of Information Science Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
morimoto.ayaka.lw1@is.naist.jp

[2] Graduate School of Information Science Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
kubo.daiki.kz7@is.naist.jp

[3] Graduate School of Information Science Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
sato.motoki.sa7.lw1@is.naist.jp

[4] Graduate School of Information Science Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
shindo@is.naist.jp

[5] Graduate School of Information Science Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
matsu@is.naist.jp

Abstract

This year's COLIEE has two tasks called phases 1 and 2. The phase 1 needs to find the relevant article given a query t2, and the phase 2 needs to answer whether the given query t2 is yes or no according to Japan civil law articles.

This paper presents our proposals for the phase 2 task. Two methods are presented. The first goes along the standard method taken by many authors, such that the relevant article t1 is selected by the similarity to the query t2 at the requirement (condition) and the effect (conclusion) descriptions of the articles. The second is our new proposal, in which Neural Networks with attention mechanism are applied to all the civil law articles in deciding the truthness of the query t2. This method takes into account all the articles by properly calculating their weighted sum.

1 Introduction

COLIEE is an annual competition for legal Information Extraction (IE) and Retrieval. COLIEE 2017 forcuses on two tasks, legal ad-hoc Information Extraction (IE) (Phase 1) and Textual Entailment (TE) (Phase 2).

While the Phase 1 can be regarded as the preprocessing for Phase 2, selecting inappropriate article(s) in Phase 1 may cause a bad effect on the Phase 2 task. We present two approaches to Phase 2, one is a standard method that selects the relevant article t1 for a legal bar exam query t2 and decides if t2 is true or not based on the selected article. In this method we tried

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 79–89

several methods to define word representations to estimate the similarity between articles. The second method, which is our new proposal, does not select the relevant article(s) for t2 but to makes use of Neural Networks with attention mechanism to utilize the information of all civil law articles to decide the correctness of the query t2.

2 Related Work

The major approaches taken by previous research were based on a two step method, to select the most relevant article t1 from the civil law articles as the evidence, then to apply a textual entailment method to decide if the query t2 is entailed by t1. The ensemble method proposed by [2] makes optimization of several weight parameters such as word overlap and tf-idf for selecting the relevant article t1 and applies a voting method with multiple classifiers. A heuristic method is proposed by [8], in which t1 is not explicitly selected. First, they make a one-to-one pair of the subject and the sentence end predicate using the result of a morphological analyzer and a case structure analyzer for each sentence. If the target phrase has no subject to be extracted, they only extract the sentence end predicate. Second, they simplify the extracted information to obtain better abstraction that helps to decide Yes/No. The method proposed by [3] uses tf-idf and Ranking SVM to select t1 and estimates the similarity between t1 and t2 based on the features of paraphrases and word embeddings coupled with condition/conclusion/exception analysis.

3 Preliminaries

This section introduces the base methods used in our two approaches, which are described in the following sections.

3.1 Word Representation

We used the idea of word2vec [7] for measuring the similarity between words and expressions. We tested several definitions for word segmentation for generating several word2vec models. As for the data for learning word embeddings, we used the judgment documents put on the web site of the Japanese Supreme Judicial Court[1], which contains 58,808 judgments (4M sentences).

We tested two sizes of word vectors, 50 and 200, when we use word2vec.

Definition of Words

Since Japanese is non-segmented language where there are no explicit word boundaries in written texts, we need to define proper units for word segmentation. The simplest method is to apply an existing word segmentation and POS tagging tool. For this approach, we used MeCab[2][4] and applied word2vec to the segmented documents. Two other segmentation criteria are also applied, one to attach suffixes to their preceding matrix phrases and the other to further attach functional expressions based on and extended from the dictionary of Japanese functional expressions[6]. Functional expressions in our case mainly mean multi-word expressions that work as postpositions or auxiliary verbs as a whole. Table 3.1 shows some examples of functional expressions. Those three segmentation criteria are simultaneously used to learn the embeddings of all possible word segmentations.

[1] http://www.courts.go.jp/app/hanrei_jp/search1
[2] http://taku910.github.io/mecab/

Functional expressions	English translation
(*ni-tui-te*)	about
(*to-ha-ie*)	however
(*koto-ga-dekiru*)	can
(*to-iu*)	that(complementizer)
(*sai-ni*)	when

Table 1: Examples of Functional Expressions

4 Similarity-based Method

The first method we applied is based on similarity between civil law articles and the given query at the requirement and the effect descriptions. The requirement and effect mean the condition and conclusion parts of Japan civil law articles and the exam queries. We use the surface clue expressions defined by [9] for identifying the requirement and effect parts of legal sentences.

4.1 Overview of the Method

The list of clue expressions that separate the requirement and effect parts of sentences in the law articles and the exam queries is shown in Table 4.2. In the current experiments we did not use the exceptional descriptions[3] often found in law articles.

The overall process is performed as the following. The details of each process is described in the subsequent subsections.

1. Extraction of the requirement and effect parts in law articles and query t2.
   - Clue expressions are used to identify requirement and effect parts.
2. Calculation of the similrity between law articles and t2 at both requirement and effect parts.
   - Previously learned word embeddings (using judgment documents) are used.
     - Attachment of suffixes and functional expressions is also considered.
   - Definition of similarity
     - word mover's distance is used.
   - Negative expressions are taken into consideration for selected articles, which flips the truth-value.

4.2 Extraction of requirement and effect parts

Japanese law adopts the Pandekten system. Therefore, the legal provisions contain the legal effect to be stipulated together with the legal requirements that constitute that condition. We used the method proposed by [9] to extract the legal requirement and effect parts from an article. First, we divide the sentence into parts separated by commas. Next, we extract the requirement part by the clue expressions shown in Table 4.2. The part matching a clue expression is regarded

[3]Typical law article sentences consist of condition and conclusion parts, possibly followed by one or more exceptional descriptions. We disregarded all the exceptional description in the current experiments.

Semantic classification	Normality	Surface patterns
Condition provision	strong	tokiha, tokiniha, saisiteha, atatteha, baaiha, baainiha, baainioiteha, baainohokaha, kagiriha, oiteha, ueha, nosaiha, unitsuiteha, runitsuiteha, reba, naraba
	weak	toki, tokini, saisite, atatte, baai, baaini, baainioite, baainohoka, kagiri, saisi, oite, ue, nosai, unitsuite, runitsuite
Status addition	weak	tokimo, tokinimo, saisitemo, atattemo, atattemo, baaimo, baainimo, baainioitemo, baaideattemo, baainohokanimo, oitemo, uemo, nosaimo, unitsuitemo, runitsuitemo
Separate judgment	strong	desadamerutokoroniyori, "number"niyori
Situation dependent	weak	jijouniyotte, jijouniyori
Means provision	weak	niyotte, niyori, motte, yotte
Objective provision	weak	you, youni, mokutekide, "lemma"utame
Applicable only	strong	taisiteha, tuiteha, kanshiteha
	weak	gendonioite, gendotoshite
Application addition	weak	taishitemo, tsuitemo, kanshitemo
Contents provision	weak	tsuite, kanshi, kanshite
Time provision	strong	madeha, madeniha, inainiha, inaiha, maeha, maeniha, atoha, atoniha, madenoaidaha, kanniha, kanha, chuuha, kikannaiha, tokiha, tokiniha, atonioiteha, maenioiteha, chuunioiteha
	weak	made, madeni, inaini, mae, maeni, ato, atoni, madenoaidani, aidani, kan, chu, kikannai, toki, tokini, atonioite, maenioite, chunioite, chitainaku
Time addition	weak	mademo, madenimo, inainimo, inaimo, atomo, atonimo, madenoaidamo, aidanimo, chumo, kikannaimo
Location regulation	strong	nainioiteha, shonioiteha
	weak	nainioite, shonioite
Location addition	strong	nainioitemo, shonioitemo
Application judgment	strong	mune
Cause specification		node
Applicable limit	weak	nouchi
Heterogeneous application	weak	monotoshite
Principle provision	weak	nishitagatte, nishitagai, nimotoduki, nimotoduite, niouji, nioujite

Table 2: Surface clue expressions for detecting requirement parts

as the requirement part, and the remaining part is regarded as the effect part. When a sentence includes two or more requirement parts, the effect part positioned right after the requirement parts is associated to each of the requirement parts to construct requirement-effect pairs.

4.3 Distance calculation

We calculate the distance between t2 and each article and extract the article t1 with the smallest distance from t2. We define the distance between sentences as the sum of "the distance between the requirement part of t2 and the requirement part of the article(t1)" and "the distance between the effect part of t2 and the effect part of the article(t1)". If the distance between t2

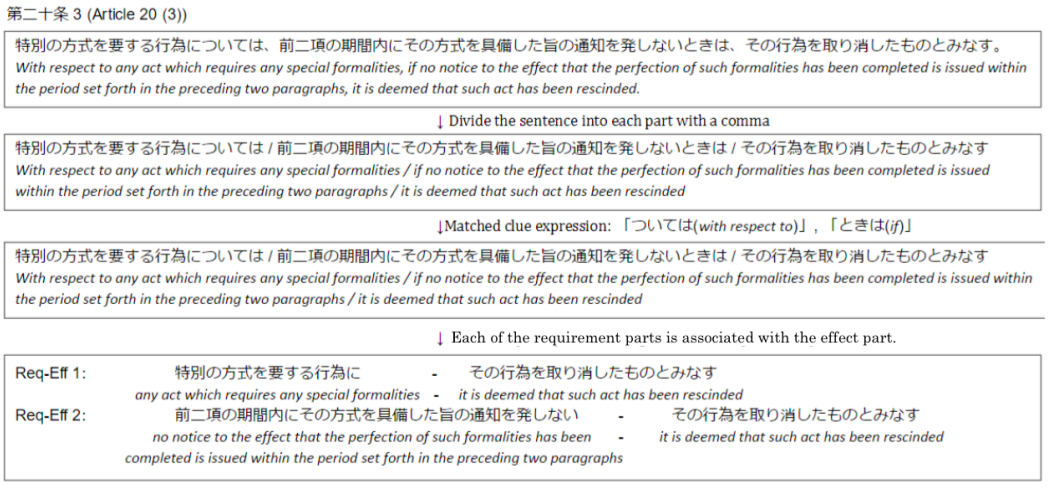

Figure 1: Process of extracting requirement-effect pairs

and the article extracted as t1 is equal to or less than a predefined threshold value, the system output is Y, and if it exceeds the threshold value, the output is N. We tested for threshold values, 25, 30, 35 and 40, and selected 35 for the experiments. The decision is made by a preliminary experiment using the H27 data (the previous year's data) as the development data. We used Word Mover's Distance as the distance measure. Word Mover's Distance is a method to calculate the distance between documents proposed by [5]. The calculation of the distance between documents is defined by the correspondence (association) or words in the compared documents. This method is applied to the calculation of the distance between sentences in our framework. The cost of corresponding words between the sentences is calculated by the distance of the distributed representation vectors, and the distance between sentences is calculated by the sum of the costs of corresponding words between the sentences.

4.3.1 Consideration of negation at sentence end

If the output of the system is Y, we consider whether the sentence end predicate[4] of either the requirement or effect part has a negative expression or not. If a negative expression is found, the answer is switched. If the article or t2 cannot be divided into the requirement and effect parts, the end predicate of the whole sentence is checked. Figure 2 shows how a negative expression is processed.

5 Attention Model

This section explains our attention model, which is based on Neural Networks with the attention mechanism[1]. We made an approach based on the idea of Memory Networks[10]. We learn vector representations for words in advance, and for given Japan civil law articles and the bar exam query t2, we calculate vector representations of articles and t2 as follows.

[4]Since Japanese is a head-final language, the main verb of a sentence always comes at the end of the sentence. Negative sentence usually have a negative auxiliary verb at the end.

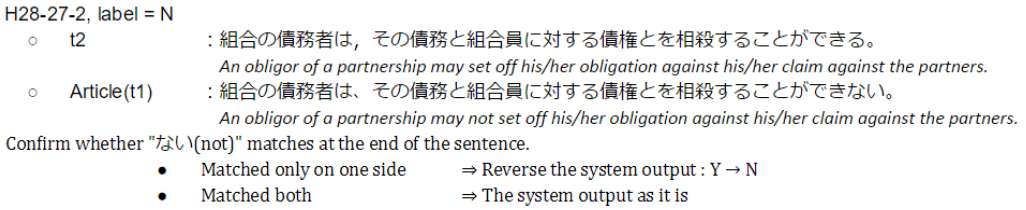

Figure 2: Process of consideration of negation

$$\vec{civilvec}_j = \sum_i civil\_idf_i \times \vec{x_i}$$

where $civilvec_j$ is the vector representation of the j-th article, which is a weighted sum of word representations $\vec{x_i}$ in the article. $civilvec$, representations of civil law articles, is a matrix with $civilvec_j$ as the j-th column. $civil\_idf$ is the idf value of the word x_i in the article[5], and $\vec{x}$ is the vector representation by word2vec learned with 60,000 jugement sentences. Those are the same representations used in the method in the previous section.

The vector for t2 is calculated in the same way as follows:

$$\vec{t2vec} = \sum_i t2\_idf_i \times \vec{x_i}$$

where $t2vec$ is the sum of word vectors in t2 weighted by their idf value. $t2\_idf$ is the idf value of the word x_i in t2, and $\vec{x}$ is the vector learned by word2vec, the same as those used for $civilvec$.

The calculation of attention is done by the inner product of the transpose of $civilvec$ and $t2vec$ defined as follows:

$$attention_j = \vec{civilvec_j}^{\,T} \cdot \vec{t2vec}$$

The size of the obtained attention vector is the number of articles. By taking the element-wise product of this attention and $civilvec$, we obtain the importance of articles, which is realized as the weighted sum of the articles, $attentioncivil$.

We then concatenate $attentioncivil$ and $t2vec$, and make it as the input of a multi-layer perceptron (MLP). We use three layer MLP, in which the number of units in the hidden layer is 50, and the size of the output unit for Y/N is 2. We used H18 to H26 for training and H27 for the development set.

Figure 3 shows the overall configuration of our Attention model.

In the MLP model, we tested several combinations of the parameters, such as the size of word2vec vectors, word2vec model themselves, activation functions and optimizers. Table 5 summarizes those options.

6 Experiments

The settings and the results of experiments are shown in this section.

[5]The idf values are calculated with all the civil law articles.

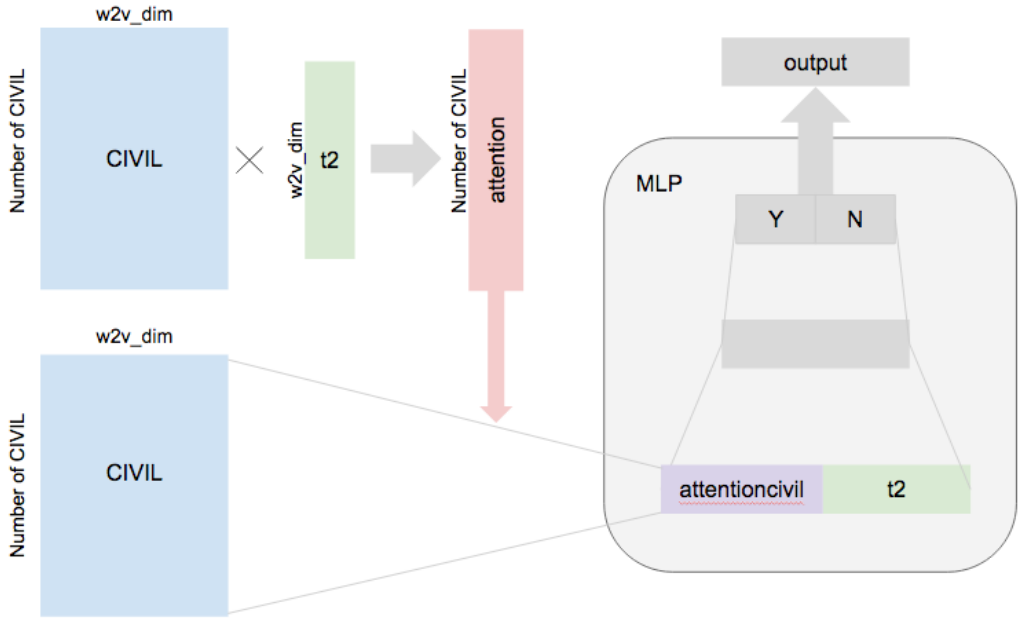

Figure 3: Image of Attention Model

word2vec(dimensions)	50, 200
word2vec(models)	traditional word units, Suffix attached, Suffix and functional expression attached
activation functions	relu, sigmoid, tanh
optimisers	AdaDelta, Adam, MomentumSGD(lr=0.01), SGD

Table 3: Parameter settings

6.1 Experimental setup

For the attention models, We will present the two submitted systems, NAIST1 and NAIST2, and the system that gave the best performance on the test data. We tested two sizes for word2vec, 50 and 200 dimensions. We hereafter report only the results with 200 dimensions since this setting achieved better results.

- NAIST1: sigmoid function for the activation function, and AdaDelta for the optimizer. Suffix and funcntional expressions are not considered.
- NAIST2: tanh for the activation function, and Adam for the optimizer. Suffix is considered, not functional expressions.
- sigMoMWE: sigmoid for the activation function, and MomentumSGD for the optimizer. Both suffix and functional expressions are considered.

	H18	H19	H20	H21	H22	H23
Separation of Requirement/Effect	0.37	0.50	0.49	0.57	0.55	0.63
+ negation	0.43	0.62	0.53	0.73	0.66	0.76
+ negation, functional expression	0.45	0.66	0.56	0.71	0.64	0.71
	H24	H25	H26	H27	H28	average
Separation of Requirement/Effect	0.58	0.58	0.50	0.49	0.40	0.5145
+ negation	0.69	0.68	0.57	0.57	0.46	0.6090
+ negation, functional expression	0.43	0.68	0.59	0.62	0.47	0.6109

Table 4: Results of Similarity-based Model

6.2 Experimental Results

Similarity-based Model

The results of the similarity-based model are shown in Table 6.2. Taking negative expressions and also taking both negative and functional expressions into consideration improve the accuracy.

Attention Model

Table 6.2 shows the results of the three Attention models. The first column shows the name of models, and the remaining columns show activation function, optimizer, word segmentation criteria, the results on the development set, and the results on the test set at the setting in which the dev set achieves the maximum value.

We tested the same experiments without using word2vec, constructing the vectors only with tf-idf, which revealed almost 10 point decrease in performance for all settings. These experiments show the effectiveness of vector representations by word2vec.

	function	optimizer	suffix/func exp	dev	test
NAIST1	sigmoid	AdaDelta	no attachment	0.6351	0.6154
NAIST2	tanh	Adam	suffix attached	0.6351	0.6538
sigMoMWE	sigmoid	MomentumSGD	suffix/func exp attached	0.6351	0.6667

Table 5: Results of Attention Models

Table 6.2 shows that the setting using tanh for activation function, Adam for optimizer and considering both suffixes and functional expressions achieves the best result.

All the results of both models are summarized in Table 6.2.

7 Discussion

7.1 Similarity-based model

Quite a few queries are answered correctly when negative expressions are taken into consideration. The following is an example:

	Accuracy
Similarity-based models	
Separation of Requirement/Effect	0.5145
+ negation	0.6090
+ negation, functional expression (NAIST3)	0.6109
Attention models	
sigAdaDelta (NAIST1)	0.6154
tanhAdamSufix (NAIST2)	0.6538
sigMoMWE	0.6667

Table 6: Results of all Models

Problem: H21-13-U, label = N

t2:

(Pledges can be created over a Thing that cannot be assigned to others.)

Most similar article:

:

(Article 343: Pledges cannot be created over a Thing that cannot be assigned to others.)

Those sentences are similar within the threshold and the system incorrectly answers Yes if the negative expression is not considered. However, if the negative expression that appears at the end of the sentence is considered, "*(can)(cannot)*", the system could answer this problem correctly by switching the truth-value of the selected article. Our strategy to consider the existence of negative expressions in similar sentences produced positive effect.

Furthermore, using word embeddings that consider attachment of suffixes and functional expressions makes similar articles to t2 more similar than the standard word segmentation criterion. In the following example, the similarity values of those sentences become closer when both suffixes and functional expressions are attached in learning word representations. Table 7.1 shows the similarity values (the lower the better).

Problem: H25-11-O, label = Y

t2:

(If a superficiary fails to pay the rent for two or more consecutive years, the landowner may demand the extinction of the superficies.)

Most similar article

:

(Article 276: If an emphyteuta fails to pay the rent for two or more consecutive years, the landowner may demand the extinction of the emphyteusis.)

- The pair of requirement parts:
  –

    (A superficiary fails to pay the rent for two or more consecutive years)

	without	with
Requirement parts	16.2	13.8
Effect parts	4.2	3.7
Total	20.4	17.5

Table 7: Similarity with/without suffix/func exp attachment

–
(An emphyteuta fails to pay the rent for two or more consecutive years)

- The pair of effect parts:

–
(The landowner may demand the extinction of the superficies)

–
(The landowner may demand the extinction of the emphyteusis)

These results show that better vector representations are learned by taking suffixes and functional expressions attached to the matrix phrases. Furthermore, better similarity measure seems to give better effect of consideration of negative expressions.

7.2 Breakdown Analysis of Attention Model

We checked precision and recall for each of Yes/No cases of the queries in the attention models. The results are shown in Table 7.2. This shows that the Attention models give balanced outputs both for Yes/No cases.

	recall	precision
	Y recall	Y precision
NAIST1 sigAdaDelta	0.5667	0.5000
NAIST2 tanhAdamSufix	0.7000	0.5385
	N recall	N precision
NAIST1 sigAdaDelta	0.6458	0.7045
NAIST2 tanhAdamSufix	0.6250	0.7692

Table 8: Recall and precision for Y/N cases

We tested the use of dropout by setting the ratio at 0.2 and 0.5 in combinations of all previous settings. Those tests show that dropout does not contribute better performance.

8 Conclusion

In this paper, we presented two types of models. One uses clue expressions to separate the requirement and effect parts of both civil law articles and queries, and measure the similarities of those parts to make decision. Considering negative expressions and suffix and functional expressions improved the accuracy.

The other model we presented, attention model, gave better performance than the similarity-based model. It uses the attention mechanism as the relevance factors of the articles. Also in this model, the word representation learning by word2vec that take suffixes and functional expressions into consideration gave better results.

8.1 Future Work

The current systems do not use any information of the exceptional parts of the law articles. Exceptional parts usually start with explicit clue expressions and are not so difficult to identify. However, proper understanding and usage of exceptional expressions are not trivial and need be further investigated.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR1513, Japan.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[2] Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kihyun Hong, and Young-Yik Rhim. An ensemble based legal information retrieval and entailment system. *Tenth International Workshop on Juris-informatics (JURISIN)*, 2016.

[3] Mi-Young Kim, Ying Xu, Yao Lu, and Randy Goebel. Legal question answering using paraphrasing and entailment analysis. *Tenth International Workshop on Juris-informatics (JURISIN)*, 2016.

[4] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. *Proceedings of Empirical Method for Natural Language Processing 2004*, 230–237, 2004.

[5] M. J. Kusner, Kolkin Sun, Y., N. I., and K. Q Weinberger. From word embeddings to document distances. *ICML*, 15:957–966, 2015.

[6] Suguru Matsuyoshi, Satoshi Sato, and Takehiko Utsuro. A dictionary of japanese functional expressions with hierarchical organization (in japanese). *Journal of natural language processing*, 14(5):123–146, 2007.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems 2013*, 3111–3119, 2013.

[8] Ryosuke Taniguchi and Yoshinobu Kano. Legal yes/no question answering system using caserole analysis. *Tenth International Workshop on Juris-informatics (JURISIN)*, 2016.

[9] Tatsuhiko Tsunoda, Hitoshi Shimizu, and Makoto Nagao. A method of extracting conditional - part and effect - part from legal sentences on a compendium of laws using surface clues (in japanese). *IPSJ Natural Language Processing*, 1997(4):129–136, 1997.

[10] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.