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

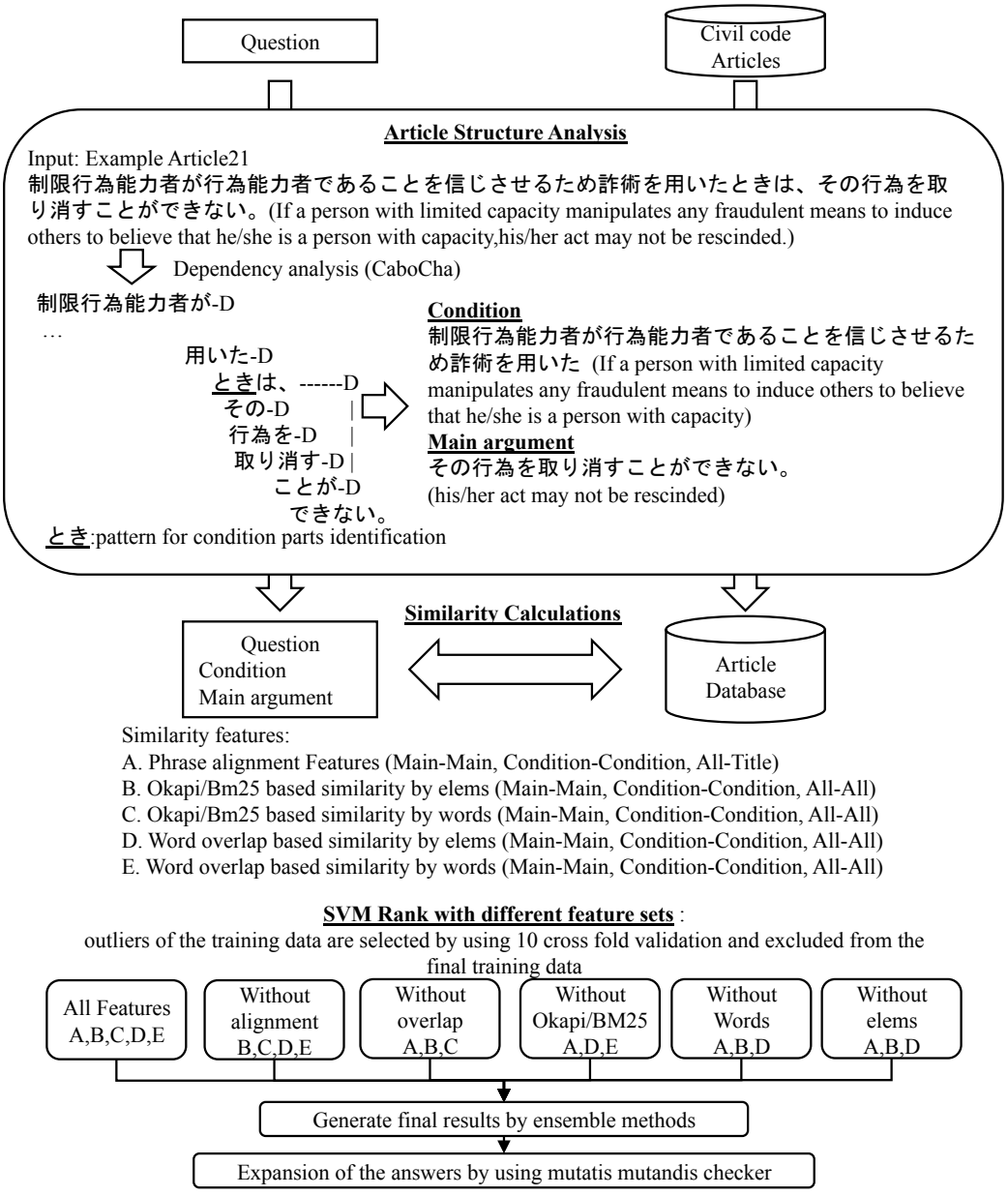

Figure 1: Workflow of the Proposed System

- 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.