Lexical to Discourse-level Corpus Modeling for Legal Question Answering

Danilo S. Carvalho*, Vu Duc Tran, Khanh Van Tran,
Viet Dac Lai, and Minh-Le Nguyen

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
{danilo, vu.tran, tvkhanh, vietld, nguyenml}@jaist.ac.jp

Abstract. On top of the traditional Question Answering (QA) problems, Legal QA presents its own set of particular challenges. These challenges mainly involve terminology resolution, searching on heterogeneous information and solving complex abstraction-realization mappings. In this work, we propose a three-stage model for answering legal questions, focused on the terminology and abstraction issues, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE). The three stages comprise: (i) Relevance analysis and ranking of legal articles, (ii) relevance re-ranking and (iii) textual entailment recognition. A set of textual features ranging from lexical to discourse-level is used to train Machine Learning models applied to (ii) and (iii). Experimental results on the previous competition data indicate competitive performance with state-of-the-art methods for the same task. Additionally a discussion on the proposed method's strengths and shortcomings is provided.

1 Introduction

Answering questions is one of the fundamental goals in the field of Natural Language Processing (NLP), receiving an unfading stream of improvements, ranging from base NLP techniques to sophisticated new approaches. It is also highly *domain dependent*, as questions are asked in different ways (i.e., syntactically), with different terminology, and expect diverse answer formulations, according to predefined discourse standards. In the legal domain, demand for higher efficiency Question Answering (QA) methods is on the rise, as we experience an explosive growth in legal document availability on the World Wide Web and specialized systems. Such growth is not accompanied by a matching increase in information analysis capabilities, leading to under-utilization of available legal resources and to potential for information quality issues [1]. The under-utilization of legal resources also brings up the matter of professional ethics and liability on law practice, since having relevant and correct information is of vital importance in legal case solving.

* Supported by CNPq – Brazil scholarship grant

Legal QA, as other QA tasks, can take advantage of improvements made in base NLP techniques, e.g. POS-tagging, parsing, and also from Knowledge Engineering, such as ontology construction and analysis methods. Nevertheless, it has its own set of challenges, which can be grouped in the following three aspects: *terminology*, *information heterogeneity* and *abstraction-realization*. The terminology problem relates to the way that words are used in legal text and how the underlying concepts represented will often differ from common language use, making distributional language modeling much less reliable. The information heterogeneity problem is related to the multitude of information types, such as past decisions, laws and facts, involved in answering a legal question, and the difficulty in identifying and composing them appropriately. The abstraction-realization problem regards the fact that laws are written with abstraction in mind, to cover most possible scenarios of any predicted situation, whereas the practice of the law is aimed at realization of the written law, in order to characterize and substantiate its application. A strong semantically motivated NLP framework is necessary to deal with these challenges, by uncovering term relationships in the legal corpora, facilitating information type identification and linking, and both describing and resolving abstraction relations between parts of the discourse.

In this work, we propose a three-stage method for Legal Question Answering, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE), covering the tasks of Information Retrieval (IR) and Recognition of Textual Entailment (RTE). The stages are: 1) Relevance analysis, 2) Relevance re-ranking and 3) Entailment classification. The method is based on a mixed n-gram language model for relevance analysis, and a set of syntactic, semantic and discourse features, used to train separate Machine Learning models for pair-wise ranking and binary classification, for the relevance re-ranking and entailment classification stages, respectively. The terminology aspect of Legal QA is accounted in all three stages, while the abstraction-realization aspect is only accounted for stages 2 and 3.

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 studies in the Legal Information Retrieval task can be divided according to their focus in Knowledge Engineering (KE) or Natural Language Processing (NLP) methods. On the KE front, Kim et. al. [2], presented an ontology-based model for law retrieval centered on a R&D and business perspective, applied to Korean law. A query is regarded as a network of legal terms, which is positioned in the document network according to its semantic relations, using the page-rank algorithm, and then ranked based on a classical TF-IDF weighting

scheme over the nouns found in the query through morpheme filtering. Santos et. al. [3] addresses the information heterogeneity problem through strong ontology conceptualization, on the perspective of consumer disputes in the air transport passenger domain, applied to European law. Both approaches [2] and [3] share solutions for improving law accessibility, but also the downsides of building and maintaining legal ontologies.

On a more traditional IR and NLP front, Liu, Chen and Ho [4] presented a method called three-phase prediction (TPP) for retrieval of relevant statutes in Taiwanese criminal law, given general language queries. The method was a hierarchical ranking approach for law corpora, featuring a combination of several Information Retrieval techniques, as well as Machine Learning and feature selection.

For the Recognition of Textual Entailment task, an application to the legal domain can be found in the work of Tran et al. [11], which addressed legal text QA with an inference method based on requisite-effectuation structures of legal sentences and similarity measures, on Japanese National Pension Law.

The previous COLIEE competition had two works with improvements over the baseline of the Legal Information Retrieval task (winner and runner-up respectively): Kim et. al. [5] presented a ranking method based on the SVM algorithm, using lemmatized words intersection, dependency pairs and TF-IDF scoring as features for training the model. Carvalho et. al. [6] presented 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. The Recognition of Textual Entailment task improvements were led by Kim et. al. [5], with a Convolutional Neural Network (CNN) based method for classification, using *word2vec* [12] word embeddings and encoded dependency tree structures as features, with dropout regularization for over-fitting avoidance.

3 Legal Question Answering – COLIEE

Answering a legal question consists in: (i) finding out the necessary knowledge for understanding a given law related question and (ii) providing the appropriate and correct answer to it. In the context of the *Competition on Legal Information Extraction/Entailment (COLIEE)*[1], the question is a legal statement (broad or situational) and the necessary knowledge is encoded in the law itself, presented as a set of articles that compose a fragment of the Japanese Civil Code. The legal statement can be true or false, according to the interpretation of the relevant civil code articles, hence the appropriate answer is either affirmative or negative. The Japanese Civil Code is composed by a collection of numbered articles, each one containing a set of declarations pertaining to a specific topic of the law, e.g., labor contracts, mortgages.

[1] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

In COLIEE 2016, activities (i) and (ii) are separated in corresponding *phases*, with an additional one combining both:

- Phase One (IR): retrieving relevant articles to a given question from all Japanese Civil Code, given a set of YES/NO questions.
- Phase Two (RTE): evaluating the entailment relationship between the question and retrieved articles.
- Phase Three: combination of Phase One and Phase Two, the system shall retrieve a list of relevant articles given a question, and then decide the entailment relationship between the retrieved articles and the provided question.

Legal text is inherently different from other types of written communication, due to the nature of both its content and intent: it is written to express rules and situations on which they apply. This should be done in an abstract, but at the same time unambiguous way, such that the rules will be applied only to the intended cases and no case is covered by multiple, conflicting rules. Those requirements produce 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 important to distinguish between the corpus specific and common senses of terms, since both of them are used, specially in questions. Another noteworthy characteristic of legal text is its preference for longer sentences, making automatic parsing more difficult.

Knowing the characteristics of legal text makes it possible to focus on a specific set of concerns when applying NLP to question answering, namely the correct identification of corpus specific vs. common sense of terms, and also the breakdown and capture of important syntactic structures, e.g., argument modification (for negation), requisite and effectuation support. For COLIEE, information heterogeneity is a lesser problem, since the answers are guaranteed to be found on the specified fragment of the civil code.

4 Proposed Approach

Once a core set of goals and concerns was established, the question answering problem was divided into three stages: (i) relevance analysis and ranking, (ii) relevance re-ranking, (iii) entailment classification. This division was motivated by the authors previous experience in the competition and to enable both independent retrieval and entailment recognition in phases one and two, as well as sequential processing in phase three.

The answering process flow is as follows: Firstly, given a single question, a ranked list with a limited number of relevant articles is obtained by using R_2NC [6] (Section 4.3). Next, a set of syntactic, semantic and discourse features is extracted from each retrieved article (Section 4.2) and the relevant article list is re-ranked using SVM^{Rank} [7] with the extracted features. The selection of relevant articles is then decided by taking into account the final re-rank position and the original R_2NC score (Section 4.4). Finally, the same set of features, plus

the R_2NC score is provided to a linear kernel SVM binary classifier, for each selected article, to obtain an entailment relationship between it and the given question. After classification result, a set of bias thresholds is used to decide the relationship. A diagram of the overall process flow is shown in Fig. 1.

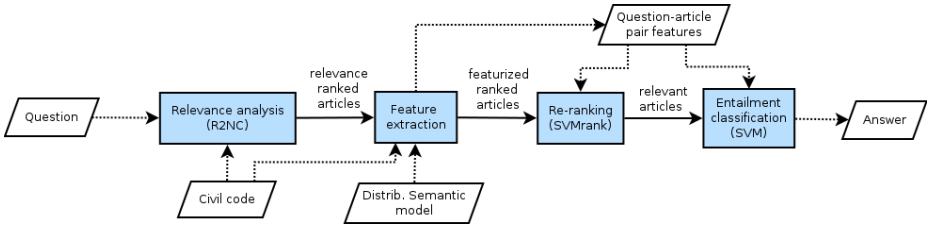

Fig. 1. The process flowchart for the legal question answering method. For COLIEE, the answer is a binary YES/NO value.

Training is done separately for each stage. For relevance analysis, both the civil code articles and training data are analyzed and a mixed n-gram model is built, containing both lemmatized n-gram and link information (Section 4.3). For the re-ranking stage, a leave-one-out session of relevance analysis is run on the training data, generating a R_2NC ranked list of articles for each question. Those lists are used as training data for SVM^{Rank}, after the featurization process described in Section 4.2, producing an article-question re-ranking model. The entailment classification stage is trained by simply featurizing phase two training data the same way as in the re-ranking stage and providing it as training data for a linear kernel SVM classifier, obtaining a question-article entailment classification model.

Sections 4.1 to 4.5 explain in detail the corpus analysis and feature construction processes, as well as each stage of the question answering process.

4.1 Corpus analysis

Analysis of the Japanese Civil Code was conducted in the aspects of lexical and semantic content, syntactic dependency structures and discourse links. As terminology plays an important role in Information Retrieval, the goal of the first was to establish a common lexical-semantic index that would allow efficient question-article term association, while at the same time being able to distinguish between corpus-specific and common language sense use. This was done in two simultaneous ways: (a) the calculation of corpus-specific n-gram statistics and (b) Distributional Semantics modeling on combined *common + legal* text data. The first would allow coarse-grained filtering through lexical relatedness measures, e.g., TF-IDF, while the latter enables more fine-grained semantic distinction.

Applying Distributional Semantics modeling, however, is complicated by the difference in volume of the available data for common vs. legal text uses, with the former being much more available than the latter. To deal with this problem, a dataset balancing technique was employed, in which the legal text was replicated until it composed a certain fraction (around 25%) of the combined data. The resulting data was used to train a *word2vec* [12] embedding model.

On the matter of syntax, dependency structures were chosen due to the interest in capturing clause modifiers, specially negations, e.g. "no", "never", that apply to both effectuation clauses and article links, e.g., "...the Manager shall **not** be liable to compensate for damages...". Those structures have low corpus specificity and can be obtained by using state-of-the-art dependency parsers with no modifications. Structural matching of sentences is also facilitated by comparing dependency tags, and can be combined with lexical and semantic similarity. The stock version of *SyntaxNet* [8] is used for dependency parsing.

Furthermore, discourse links are also an important source of information, since they bind together complementary information about a topic and allow contradictions to be found. In an Information Retrieval setting where articles are considered *documents*, the article and paragraph references are defined as the discourse links and are typed into a set of classes ("plain", "case" and "provisioning") and indexed. Finally, to take maximum advantage of the discourse link information, all text processed from the civil code is indexed at paragraph level, allowing efficient reference resolution. Fig. 2 illustrates the corpus structure.

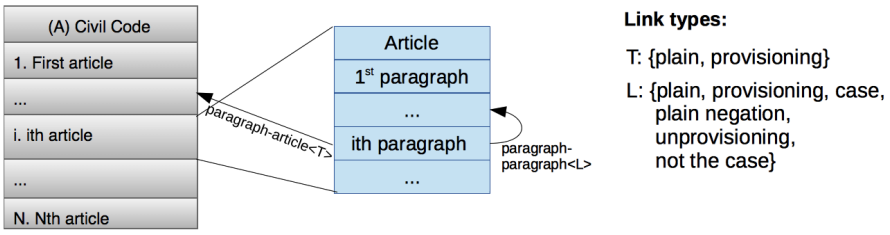

Fig. 2. The structure of the Japanese Civil Code corpus. Link types specify how the arguments in both sides of the reference are related. For example, a "case" link type defines an argument as valid on the particular condition denoted by the referent.

4.2 Feature construction

Once a clear understanding of the corpus structure was reached, it was necessary to find an appropriate way to associate questions with their respective answers. For this reason, a set of features was developed, incorporating information from all the previously mentioned corpus aspects (lexical, semantic, syntactic dependency and discourse links), with the goal of training Machine Learning models to differentiate among several association characteristics.

The features are defined as follows:

- **Dependency node semantic similarity**: *word2vec* embedding cosine similarity, calculated over words matching predefined pairs of dependency tags in the question and article paragraph, respectively. For example, a pair (*nsubj*, *dobj*) would match a question's "nsubj" (a noun subject) and an article paragraph's "dobj" (a direct object), and the similarity would be calculated. Each pair is regarded as a single feature. Dependency pairs considered for semantic similarity were: *(nn,nsubj), (nsubj,nsubjpass), (pobj,dobj), (poss,nn), (pobj,nsubj), (dobj,nsubj)*, plus same element pairs.

- **N-gram intersection**: The size of the n-gram intersection (with n from 1 up to 10) between a question and paragraph, normalized by the paragraph's n-gram set size.
- **Paragraph links**: normalized index of link destinations for each paragraph (up to 6).
- **Paragraph link types**: boolean indicating the pertinence relationship of the respective link to a certain type or its negation. There are 3 possible types: "plain", "case", "provisioning", and their respective negations: "plain negation", "not the case" and "unprovisioning". They define the discourse relations between a paragraph and the content it refers to. For example, an unprovisioning link can be found in Article 295, paragraph 2: "The provisions of the preceding paragraph shall not apply…".
- **Negation similarity**: *word2vec* embedding cosine similarity between negated terms in the question and paragraph, normalized by the number of negated terms. Negated terms are obtained from negation links in the dependency tree. If either one of the question or paragraph does not contain negated terms, similarity is set to zero. If neither contain negated terms, similarity is set to one.

The features are calculated for the association question-paragraph, for each paragraph of each article evaluated. Since articles have different numbers of paragraphs, the paragraph number was fixed at 4, as there were very few cases of articles longer than that. Exceeding paragraph features were filled with placeholder values (0 for similarity and boolean features and -1 for link indexes). Similarity features were designed to facilitate terminology sense distinction and abstraction-realization issues, e.g., "neighbor" $\leftrightarrow$ "agent" $\leftrightarrow$ "person", while the link features were intended to highlight discourse conformity/contradiction, and facilitate entailment classification.

4.3 Relevance analysis

The relevance analysis stage was done entirely with R_2NC [6], which can be summarized in the following process:

1. Collect the entire content for each article, including section title;
2. Check references between articles and annotate accordingly;
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 references n-grams;
8. Associate article number and references;
9. Store the model.

Except for step 4, each step is responsible for adding new information to the model. The information is obtained either from the text, e.g., section title, references, or from morphological analysis, e.g., POS-tags, lemmas. If an article

have references, its n-gram set is expanded with the references' n-grams. This is done so that all the necessary information for interpretation of any single article is self-contained. Besides the n-grams, links between the articles are also stored. To include the training data information, the same process is repeated for the questions, and n-gram sets from the trained questions are used to expand the associated articles' n-gram models. Tokenization and lemmatization were done using *NLTK*[2] (v. 3.0.2) with the Punkt tokenizer and WordNetLemmatizer modules, respectively. Those modules were used with their unchanged default models and settings, trained with the Punk corpus and WordNet, respectively. POS-tagging was done using *Stanford Tagger*[3] (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. 3 illustrates the n-gram model creation scheme.

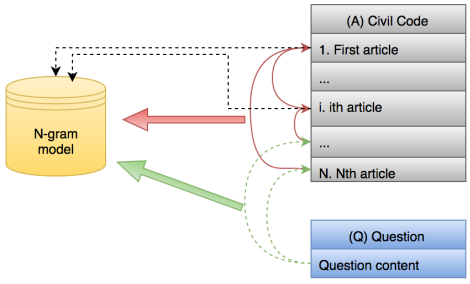

Fig. 3. The n-gram model construction scheme. Both article-article and question-article links are stored, and the respective document n-gram sets are associated. A single association index is generated for each article.

The relative relevance of an article with regard to the content of a question is ranked by applying the following scoring formula:

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

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

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

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.

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

4.4 Relevance re-ranking

Preliminary analysis of the previous competition data showed that R_2NC could achieve a 0.8 recall when using only the first 20 ranked articles (out of 1106). This meant that most unrelated content was already being filtered out and a directed re-ranking approach was appropriate to improve precision on the reduced relevant set. Further observation of the ranked lists revealed that the abstraction issue was responsible for a considerable part of the loss in precision.

The features described in Section 4.2 were designed taking this into consideration and were used to train a question-article re-ranking model using SVM^{Rank} [7]. Training data is obtained by running a leave-one-out session of R_2NC on COLIEE's training questions. This results in a ranked list of relevant articles for each question (limited to 20), for which the features are calculated. The article lists are then presented as training inputs to SVM^{Rank}. If an article list does not contain the correct relevant ones, those are added to the end of the list, with their corresponding ranks (1^{st}, 2^{nd}, ...). The re-ranking stage is done by featurizing R_2NC outputs and using them as inputs for the trained SVM^{Rank} model. Final selection of the relevant articles is done by taking the first ranked from the list, and optionally the following ones for which the distance $d = score(q, a[1]) - score(q, a[i]) < extend\_thresh$, where q is the given question, $a[i]$ is the i^{th} ranked article, $score(q, a)$ is the R_2NC score and $extend\_thresh$ is a threshold parameter. Fig. 4 illustrates the re-ranking process.

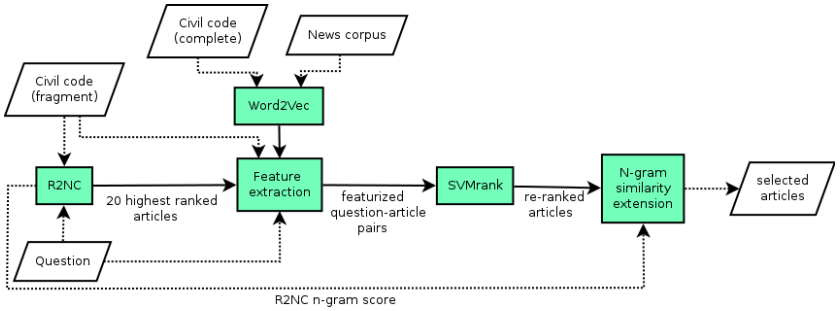

Fig. 4. The re-ranking process flowchart. The n-gram similarity extension step is decided upon a user-defined threshold $extend\_thresh$. This way, articles arbitrarily close to the 1st ranked one may be also selected.

4.5 Recognition of Textual Entailment

For the final stage, observation of the previous competition data pointed to the interpretation of argument negation as a key factor in textual entailment recognition performance. To deal with this, discourse link type and negation information where included in the feature set, under the hypothesis that they carry enough information to induce entailment discrimination on the appropriate cases, when used with a Machine Learning classifier. In this stage, the classifier of choice was the linear kernel SVM, with the SVM^{perf} [9] implementation.

Training the classifier was done by calculating the same features as in the previous stage, but directly on COLIEE's training data, with R_2NC score as an

additional feature. All pairs question-article were featurized and the labels set to the corresponding question label.

Entailment recognition was done in the following way:

1. Featurize the relevant/selected articles for a given question.
2. Classify each pair question-article.
3. For each pair question-article:
   (a) If maximum n-gram intersection feature > *strong_intersection_thresh*, then question answer is "Y".
   (b) If class = "Y" and the maximum n-gram intersection > *weak_intersection_thresh*, then question answer is "Y".
   (c) Else, class = "N", then question answer is "N".

where *strong_intersection_thresh* and *weak_intersection_thresh* are threshold parameters, intended to bias the entailment classification towards the answers found for articles with highly intersecting question-paragraph pairs.

5 Experiments and Results

5.1 Experimental Setup

The legal question answering dataset was obtained from the published data for the COLIEE shared task [4], 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 three tasks contains 412 pairs (question, relevant articles). Experiments where divided in phases one and two only, dealing with Information Retrieval and Textual Entailment methods respectively.

Additional data used in the experiments include the training segment of "1 billion word language model benchmark" corpus [10] and the complete Japanese Civil Code[5], which were used to train the Distributional Semantics model. The combined size of the corpora after balancing is approximately 1.2 billion words.

Experiment data was divided into training and validation, taking advantage of the fact that the previous competition test data was distributed as part of the current one's training data. By using the previous competition as validation data, a direct performance comparison was possible, facilitating the evaluation process. Competition files *riteval_H{18..23}.xml* were used for training and *riteval_H{24, 25}.xml* for validation.

5.2 Parameter adjustment

Parameter adjustment was done separately for each stage. R_2NC parameter k (the maximum n-gram size) was kept as in [6] ($k = 3$), since changing it did not offer performance improvements on a leave-one-out test over COLIEE 2016 training data. However, increasing I_q to 0.98 and thus decreasing I_{art} to 0.02 incurred in a 0.04 F-score increase. For re-ranking, SVM^{rank} was run with parameter $C = 2000$ and $extend\_thresh = 0.5 \times 10^{-7}$. C was adjusted by starting

[4] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

[5] www.japaneselawtranslation.go.jp

from 200 and increasing its value by 200 until performance dropped or model convergence could not be reached in the validation set. *extend_thresh* was initially set to 0.5 and then divided by 10 until no improvements could be found. For entailment classification, SVM^{perf} was run with parameter $C = 400$ and thresholds *strong_intersection_thresh* and *weak_intersection_thresh* were set to 0.9 and 0.4 respectively. Adjustment was made in the same way as in the re-ranking stage, but with C increasing by 100 and *strong_intersection_thresh* and *weak_intersection_thresh* starting at 1.0 and decreasing by steps of 0.1. *word2vec* parameters were set as: $d = 200$, $cbow = 0$, $window = 0$, $negative = 0$. SyntaxNet was run with default parameters.

5.3 Baselines

As a COLIEE directed effort, the single baseline used in this work was the previous competition results, favored by the possibility of direct performance comparison given by the release of the corresponding test data with ground truth labels. The results are compared for both the winner [5] and the runner up [6] in phase one, and only for the winner [5] in phase two. Phase three results were considered as consequence of the performance in phases one and two, so they were not evaluated.

5.4 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. (3), (4) and (5)). In phase two, accuracy (A) measurement is used (Eq. (6)).

$$P = \frac{Cr}{Rt} \quad (3) \qquad R = \frac{Cr}{Rl} \quad (4) \qquad F = \frac{2(P * R)}{P + R} \quad (5) \qquad A = \frac{Cq}{Q} \quad (6)$$

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.

5.5 Pre-competition Results

Experiment results on the validation set are presented in Tables 1 and 2.

Table 1. Experiment results for phase one (IR). All systems were tested on the file *riteval_H24.xml* from the COLIEE dataset.

Method	Precision	Recall	F-measure
Kim et. al. [5]	0.6329	0.4902	0.5525
Carvalho et. al. [6]	0.5663	0.4608	0.5081
This method	0.6707	0.5392	**0.5978**

Table 2. Experiment results for phase one (RTE). All systems were tested on the file *riteval_H25.xml* from the COLIEE dataset.

Method	Accuracy
Kim et. al. [5]	0.6667
This method	**0.6969**

The results indicate a noticeable improvement in both tasks, specially in phase one. Phase one results also indicate a recall consistent with state-of-the-art methods for similar legal corpora, slightly surpassing TPP's [4] mark of 0.52 for the top 3 ranked law articles, while using considerably less training data: 267 documents for R_2NC against 1518 documents for *TPP*.

Phase two results also indicate that the feature set developed in this work can help deciding on textual entailment by including relevant discourse information, in the form of reference typing and argument negation matching.

5.6 Error Analysis and Discussion

Observation of the misranked and misclassified cases helped understanding the improvements obtained over R_2NC, as well as the limitations of the proposed method.

The following example shows a case of R_2NC misrank, while the proposed method succeeds:

Table 3. Example question for which the re-ranking approach improves the result. R_2NC rank shows the position before the re-ranking.

ID	Question	Rel. article	R_2NC rank	re-rank
H24-19-1	In cases where the obligee (C) exercises the credit of sale value vested in the obligor (A) against (B), and filing the action regarding to the credit, if the upholding judgment of the action is established, the credit is deemed to extinguish by the performance	Article 423 (1) An obligee may exercise the right vested in the obligor in order to preserve his/her own claim;provided, however, that, this shall not apply to rights which are exclusive and personal to the obligor. (2) Until exercised by way of subrogation admitted in a judicial proceeding, the obligee may not exercise the right set forth in the preceding paragraph unless and until his/her claim has become due;provided, however, that, this shall not apply to any act of preservation.	19th	1st

In this case, the use of structural similarity features allowed abstraction to be applied (e.g., "right" → "credit of sale value") and the lexical aspect to be overcome in the ranking. In this case, 18 other articles had a greater share of term occurrences such as "obligee", "right" and "credit" as keywords.

In the example shown in Table 4, however, the proposed method is still unable to correctly rank the articles. In that case, a very high lexical match (from Article 21, not included due to space constraints) overweights all the remaining features. A completely different semantic approach would be needed to address such case.

Table 4. Example question for which the re-ranking approach is still unable to improve the position of the relevant article. R_2NC rank shows the position before the re-ranking.

ID	Question	Rel. article	R_2NC rank	re-rank
H24-2-4	In cases where a person with limited capacity manipulates any fraudulent means to induce others to believe that he/she is a person with capacity, his/her juristic act has effect even if there is a mistake in any element of the juristic act in question.	Article 95 Manifestation of intention has no effect when there is a mistake in any element of the juristic act in question; provided, however, that the person who made the manifestation of intention may not assert such nullity by himself/herself if he/she was grossly negligent.	2nd	2nd

For phase two, an intriguing development was found, as no relevant article in phase two validation data contains explicit clause negations, although some questions do. Despite that, questions were misclassified when the negated link features were removed, suggesting an indirect (implicit) link between question clause modifiers and the articles link chain. However, a more detailed investigation is needed to expand such hypothesis.

6 Conclusion

Legal Question Answering presents a set of particular challenges, on top of the traditional QA problems. These challenges mainly revolve around terminology resolution, searching on heterogeneous information and solving complex abstraction-realization mappings. In the context of the Competition on Legal Information Extraction/Entailment (COLIEE), we propose a three-stage model for answering legal questions, focused on the terminology and abstraction issues.

Starting with relevance analysis, a mixed n-gram model is built from the Japanese Civil Code corpus and training pairs of questions and their relevant articles. The model is then used to rank civil code articles according to their relevance to the question. Next, a limited number of articles is taken from the previously ranked list and then re-ranked, by extracting a set of lexico-syntactic, semantic and discourse link features from the ranked question-article pairs, and using then as inputs for a Learning-to-Rank method. Agglutination of close matches by first stage score threshold is also performed, to increase the recall. The re-ranking system acts as a fine grained relevance analyzer, working on a limited sample but with more detailed information. In the final stage, the same features used in the second stage is combined with the first stage relevance score to serve as input for a binary classifier on the entailment relationship of question-article pairs. Posterior biasing is applied to decide the entailment for questions with multiple relevant articles.

Experimental results with the previous competition data indicate that the proposed method improves over the winning results by 0.04 F-score on phase one (Information Retrieval) and by 3% accuracy on phase two (Recognition of Textual Entailment). The results are also consistent with state-of-the-art results in similar legal corpora. Analysis of the results provided important information about limitations of the proposed approach. Those shall be addressed in future work.

As future work, a secondary level of semantic processing, aimed at argument reasoning, can be developed for addressing some difficult questions. An alternative Distributional Semantic approach with a broader scope shall also be considered.

Acknowledgements

This work is supported partly by the grant of NII Research Cooperation and JAIST's Research grant.

References

1. Robert C. Berring: "The heart of legal information: The crumbling infrastructure of legal research". *Legal information and the development of American law*. St. Paul, MN: Thomson/West, 2008.
2. Wooju Kim, Youna Lee, Donghe Kim, Minjae Won, and HaeMin Jung: "Ontology-based model of law retrieval system for R&D projects." *In Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World*, p. 26. ACM, 2016.
3. Cristiana Santos, Vctor Rodriguez-Doncel, Pompeu Casanovas, and Leon van der Torre: "Modeling Relevant Legal Information for Consumer Disputes." *In International Conference on Electronic Government and the Information Systems Perspective*, pp. 150-165. Springer International Publishing, 2016.
4. Yi-Hung Liu, Yen-Liang Chen and Wu-Liang Ho: "Predicting associated statutes for legal problems." *Information Processing & Management* 51.1: 194-211, 2015.
5. Mi-Young Kim, Ying Xu, Randy Goebel: "A Convolutional Neural Network in Legal Question Answering." *In Proceedings of the 9th International Workshop on Juris-informatics (JURISIN 2015)*, pp. 211-222, 2016.
6. Danilo S. Carvalho, Minh-Tien Nguyen, Chien-Xuan Tran, and Minh-Le Nguyen: "Lexical-Morphological Modeling for Legal Text Analysis." *Lecture Notes in Computer Science: New Frontiers in Artificial Intelligence*, 2016.
7. Thorsten Joachims: "Training Linear SVMs in Linear Time." *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.
8. Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins: "Globally normalized transition-based neural networks." *arXiv preprint arXiv:1603.06042*, 2016.
9. Thorsten Joachims: "A Support Vector Method for Multivariate Performance Measures." *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.
10. 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.
11. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen and Akira Shimazu: "Answering Legal Questions by Mining Reference Information". New Frontiers in Artificial Intelligence. Springer International Publishing: 214-229, 2014.
12. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean: "Distributed Representations of Words and Phrases and their Compositionality". In Proceedings of NIPS, 2013.