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/m̃iyoung2/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.

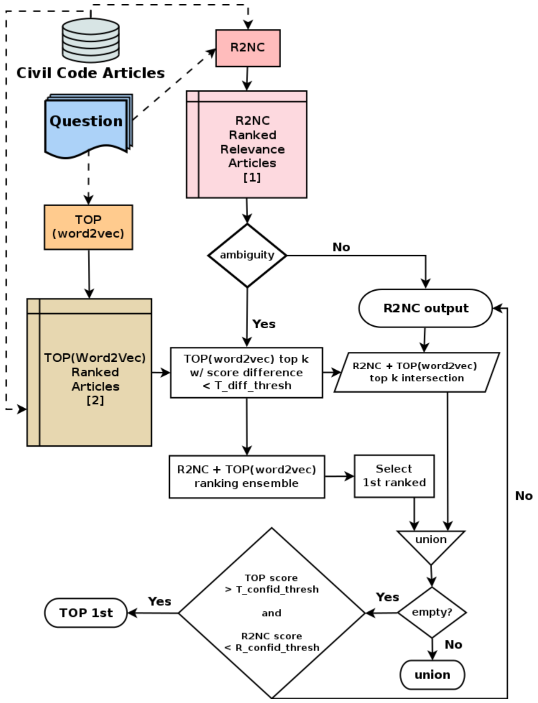
Civil Code Articles
R2NC
Question
R2NC Ranked Relevance Articles [1]
TOP (word2vec)
ambiguity
No
R2NC output
Yes
TOP(Word2Vec) Ranked Articles [2]
TOP(word2vec) top k w/ score difference < T_diff_thresh
R2NC + TOP(word2vec) top k intersection
R2NC + TOP(word2vec) ranking ensemble
Select 1st ranked
No
union
TOP score > T_confid_thresh and R2NC score < R_confid_thresh
Yes
TOP 1st
Yes
empty?
No
union

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.

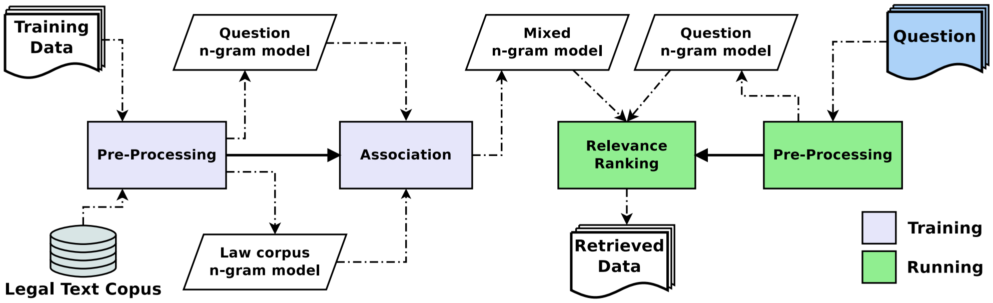

Figure 2: R_2NC process flow. N-gram language models from both the law corpus and training questions are associated into a mixed n-gram model. Article relevance for unseen questions is evaluated using this mixed model.

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

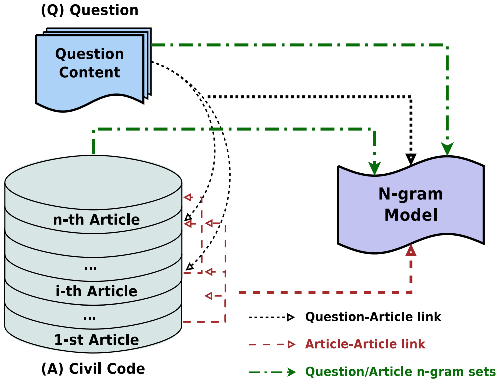

Figure 3: The n-gram model construction scheme. Both article-article and question-article links are stored, and the respective document n-gram sets are associated. A single association index is generated for each article.

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

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/m̃iyoung2/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. (⋆) denotes the method presented in this paper, while ⋄ 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.