Answering Yes/No Questions in Legal Bar Exams

Conference Paper · November 2014

DOI: 10.1007/978-3-319-10061-6_14

4 authors, including:

mi-young Kim
University of Alberta, Augustana Faculty
49 PUBLICATIONS **320** CITATIONS

SEE PROFILE

Randy Goebel
University of Alberta
206 PUBLICATIONS **2,808** CITATIONS

SEE PROFILE

Answering Yes/No Questions in Legal Bar Exams

Mi-Young Kim[1], Ying Xu[1], Randy Goebel[1], and Ken Satoh[2]

[1] Dept. of Computing Science, University of Alberta, Edmonton, Canada
[2] National Institute of Informatics/Sokendai, Tokyo, Japan

{miyoung2,yx2,rgoebel}@ualberta.ca
ksatoh@nii.ac.jp

Abstract. The development of Question Answering (QA) systems has become important because it reveals research issues that require insight from a variety of disciplines, including Artificial Intelligence, Information Extraction, Natural Language Processing, and Psychology. Our goal here is to develop a QA approach to answer yes/no questions relevant to civil laws in legal bar exams. A bar examination is intended to determine whether a candidate is qualified to practice law in a given jurisdiction. We have found that the development of a QA system for this task provides insight into the challenges of formalizing reasoning about legal text, and about how to exploit advances in computational linguistics. We separate our QA approach into two steps. The first step is to identify legal documents relevant to the exam questions; the second step is to answer the questions by analyzing the relevant documents. In our initial approach described here, the first step has been already solved for us: the appropriate articles for each question have been identified by legal experts. So here, we focus on the second task, which can be considered as a form of Recognizing Textual Entailment (RTE), where input to the system is a question sentence and its corresponding civil law article(s), and the output is a binary answer: whether the question sentence is entailed from the article(s). We propose a hybrid method, which combines simple rules and an unsupervised learning model using deep linguistic features. We first construct a knowledge base for negation and antonym words for the legal domain. We then identify potential premise and conclusion components of input questions and documents, based on text patterns and separating commas. We further classify the questions into easy and difficult ones, and develop a two-phase method for answering yes/no questions. We answer easy questions by negation/antonym detection. For more difficult questions, we adapt an unsupervised machine learning method based on morphological, syntactic, and lexical semantic analysis on identified premises and conclusions. This provides the basis to compare the semantic correlation between a question and a legal article. Our experimental results show reasonable performance, which improves the baseline system, and outperforms an SVM-based supervised machine learning model.

Keywords: legal text mining, natural language processing, question answering, recognizing textual entailment

1 Introduction

The last decade has challenged many disciplines with a deluge of written information, typically in digital form. In the legal domain, this situation was anticipated and referred to as the "information crisis" in law, and served as the impetus for the development of legal full-text information extraction systems [1].

Our immediate goal is to automatically answer yes/no questions relevant to civil law in legal bar exams. Legal bar examinations are intended to determine whether a candidate is qualified to practice law in a given jurisdiction. The task can be conceived as the first of evaluating the semantic equivalence between input questions and relevant law articles. This task is related to Recognizing Textual Entailment (RTE), where the task is to confirm whether a question sentence is entailed by a corresponding civil law article; the output is a binary classification decision, "yes" or "no". Since the input questions and articles are all domain-specific, they share the same technical terms, and therefore detecting semantic relationship is easier than for open-domain questions.

Earlier studies have concluded that simple word overlap measures (e.g., bag of words, n-grams) have a surprising degree of utility [3], but they still need to be improved. A common problem identified in these earlier systems is the lack of understanding the semantic relation between words and phrases. Later systems that include more linguistic features extracted from resources such as WordNet showed better performance [4]. Previous studies have also shown that syntactic features from parse trees are also helpful in this task [5]. Even more recent studies gained further leverage from systematic exploration of the syntactic feature space through analysis of parse trees [6]. Our methods also extract some deep linguistic features, such as lexically semantic information from thesauri, and syntactic dependency information.

An interesting recent development in the area of recognizing textual entailment (RTE) has been the application of so-called natural logics [7]. Natural logics provide a form of meaning representations that are essentially phrase-structured natural language sentences; from these one can compute entailments as substitutions for constituents (words or phrases). Any implementation of a natural logic will require the specification of conditions for monotonicity, subsectiveness, subsumption and exclusion properties of the predicates and modifiers identified in the vocabulary of the text, as well as vocabulary-independent meta-axioms that support reasoning with these properties. In addition, the natural logic inference systems need to incorporate a lot of background domain-dependent subsumption facts (e.g., walking and running are both subsumed by some kind of human locomotion). In our study, since the questions and corresponding documents are all in a restricted legal domain, they share the same technical terms. Because it is easy to compare lexical terms in this domain, we do not implement a general logic which needs to supply general, vocabulary-independent meta-axioms, but instead use unsupervised learning by constructing a domain-specific knowledge base and extracting deep linguistic features.

Question answering system comprises the extraction of a relevant paragraph of a source text that somehow aligns with the information need expressed by a natural language question. In order to automatically answer the bar exam yes/no questions, we first have to find the corresponding articles based on the Q/A technologies, and then we have to compare the meaning of the input question and the corresponding

Table 1. 'No' question types in legal bar exam

'No' question types	Proportion	'No' question types	Proportion
Negation	0.32	Constraints in premise	0.2
Using (semi-) anonym word	0.176	Constraints in conclusion	0.04
Paraphrasing of a phrase	0.12	Etc.	0.08
Exceptional case written in article	0.064		

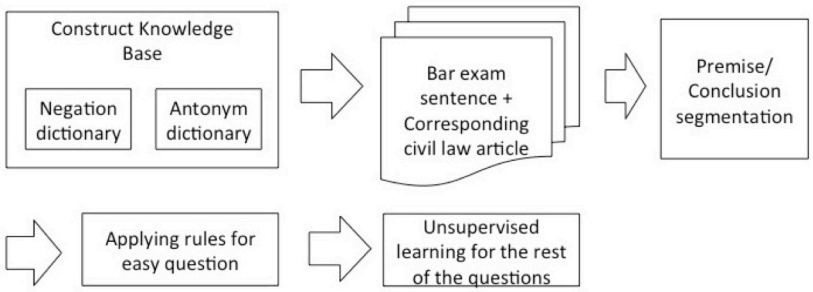

Figure 1. Overall Procedure of our method

article. We can then produce yes/no answers. In our case here, legal experts have already annotated corresponding civil law articles for questions. So for now, we focus only on comparing the meaning of the input questions and corresponding articles, to provide the basis for our yes/no assessment.

The rest of our paper is structured as follows. First, we explain the details of our method in Section 2, and we describe experimental setup, results, and error analysis in Section 3. Section 4 explains related work, and finally our future work and conclusions are described in Section 5.

2 Our method

In order to answer yes/no questions according to the corresponding legal articles, we have to align structures and words embedded in the sentence pairs. These alignments are not given as inputs, and to determine them is a non-trivial task. This alignment-based approach has been shown effective by many RTE, QA, and MTE systems [6,8]. But alignment is not the only approach. Other studies have successfully applied theorem proving and logical induction techniques, translating both sentences to more abstract knowledge representations and then doing inference on these representations [9]. In comparison to previous work that exploits various ad-hoc or heuristic methods, we intend to build on more principled techniques.

Table 2. Examples of negation types

Negation type	Example
Negation affix	not, no, less…
Negation words	unreasonable, block, withdraw, cancel, shrink, forbid, prohibit..
Negation concepts	n457, n444 …

Table 3. Examples of antonym dictionary

Term	(Semi) Antonym	Term	(Semi-) Antonym
principal	interest	creditor	debtor
employer	employee	credit	debt
creditor	third-party	debtor	third-party

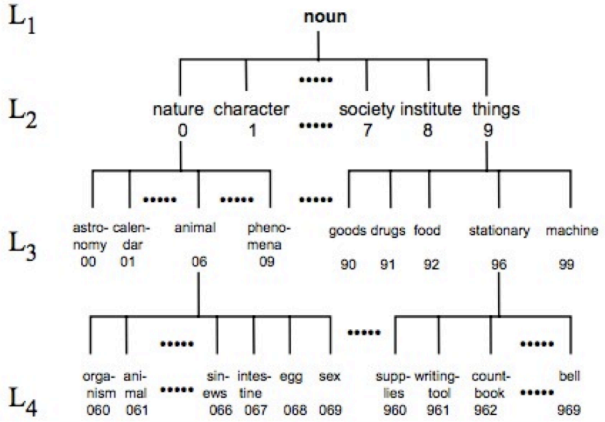

Figure 2. Concept hierarchy of the Kadokawa thesaurus

Part of our method is to classify the yes/no questions into a spectrum from simple to difficult, according to the observations on the data. Table 1 shows "no" question types in our data. "Negation" is the highest proportion with 32%, and the second largest category is when there are different constraints in the premises between question and article. The third largest category arises when a word in the article is replaced with an "antonym" word or "different-meaning"-word. The fourth is a question with paraphrasing that cannot be resolved without expert knowledge, and the fifth are those where there is a difference of constraints in the conclusion between question and corresponding article. We notice that the categories of negation and antonym make up about 50% of the total.

To address these two categories, we first construct two kinds of knowledge bases: a negation dictionary and an antonym dictionary. When an input pair (question, corresponding article) is given, we look for the premise and conclusion parts of the input question and corresponding civil law article. We then use a rule-based method to solve easy questions. Subsequent categories are addressed by exploiting some

machine learning, based on exploiting some deeper linguistic information. We describe each process in detail in the following subsections. The overall workflow of our method is shown in Figure 1.

2.1 Constructing supportive knowledge bases

The most important features in determining semantic equivalence or near-equivalence, are accurate attribution of negation and confirming the use of antonyms. In our approach, we construct a negation knowledge base from the civil law articles. We identify two types of negation expressions: one is to note negation prefixes such as "not", "no", etc. The other is the case where the word itself conveys negative information. To extend our identification of negation words, we also use the Kadokawa thesaurus [10] which has a 4-level hierarchy of about 1,100 semantic classes, as shown in Figure 2. Concept nodes in level L1, L2, and L3 are further divided into 10 subclasses. Table 2 shows examples of negation types.

We also manually collect the legal terms that can be used as antonyms or semi-antonyms having the same named entities from the civil law articles. Table 3 shows examples of the antonym dictionary. In a preprocessing phase, we add features "NEG" for the negation words in the questions and articles, and add features "ANT" for the (semi-) antonyms in the questions by comparing the words in the corresponding articles.

2.2 Premise and conclusion detection

The general idea of determining alignment between question and law article is easier if we first divide the question and articles into premise and conclusion. We compare the premise (conclusion) of a question and that of the corresponding article, and then examine negation or contradiction of intended meaning if it exists. Sentences in the legal law articles are usually long (average 21.25 words/sentence according to our data), and a comma is the most common delimiter between phrases.

Based on commas and keywords of a premise, we segment sentences. The keywords of premise are as follows: “in case(s) ”, “if”, “unless”, “with respect to”, “when”, and comma. After segmentation, the last segment is considered to be a conclusion, and the rest of the sentence is considered as a premise as follows:

$$conclusion := segment_{last}(sentence, keyword),$$
$$premise := \sum_{i \neq last} segment_i(sentence, keyword),$$

In our context, it is typical that a law article consists of two sentences. The first sentence is the main explanation of the law, and the second is for any exceptions. The second sentence only includes specific terms in the exceptional case and conclusion of the case. Consider the following example:

<Civil Law Article 295-1>: If a possessor of a Thing belonging to another person has a claim that has arisen with respect to that Thing, he/she may retain that thing until that claim is satisfied. ***Provided, however, that this shall not apply if such claim has not yet fallen due****.*

Our central task here is to determine if a question belongs to the overall case or to the exceptional case. To do so, we cannot simply use the count of overlapped words as features, because typically the first article sentence, which represents the overall case, has more overlapping words with the question than the second article sentence, which represents the exceptional case. Since the second sentence typically includes only terms specific to describing exceptional cases and the conclusion in the case, we first compare the premise of the second sentence and that of the question: if the content words in the premise of the exceptional sentence are included in the question above a threshold, then we conclude that the question belongs to the exceptional case. Our measure for this is as follows:

$$if\left(\frac{n(w_i(article_{n\_exception}, premise) \in W(q_n, premise))}{n(W(article_{n\_exception}, premise))} >= threshold\right)$$

$$then,\ article_n := article_{n\_exception}$$

$$otherwise,\ article_n := article_{n\_main},$$

where we define $article_{n\_main}$ as the main sentence of the corresponding article of n-th question, and $article_{n\_exception}$ as the second sentence describing exceptional case. We use only the lemma of each word for comparison, and consider only contents words (verb, adjective, and noun). We currently set the threshold to 0.7 based on a 10% of random sampling of all data in our experiments.

2.3 Applying rules for easy questions

if (neg_level(article$_n$, premise)+neg_level(article$_n$, premise)
= neg_level(q_n, conclusion)+neg_level(q_n, conclusion)),
Answer$_n$:= yes,
otherwise, Answer$_n$:= no,
where neg_level() := 1 if negation and antonym occur odd number of times.
neg_level() := 0 otherwise.

Figure 3. Answering rule for easy questions

Because our language domain is restricted for both the input questions and law articles, there are some questions that can be answered easily using only negation and antonym information. If the question and article share the same word as the root in each syntactic tree, we consider the question as easy, which means it can be answered using only negation/antonym detection. Here is an example:

Question : If person A sells owned land X to person B, but soon after, sells the same land X to person C then if the registration title is transferred to B, ***then person B can assert against C in the acquisition of ownership of land X.***

Article 177 : **Acquisitions of, losses of and changes in real rights concerning immovable properties may not be asserted against third parties**, *unless the same are registered pursuant to the applicable provisions of the Real Estate Registration Act and other laws regarding registration.*

➔ *Conclusion of the question : then person B can assert against C in the acquisition of ownership of land X.*
Premise of the question : : If person A sells owned land X to person B, but soon after, sells the same land X to person C then if the registration title is transferred to B,
Conclusion of the article : Acquisitions of, losses of and changes in real rights concerning immovable properties may ***not*** *be asserted against third parties,*
Premise of the article : ***unless*** *the same are registered pursuant to the applicable provisions of the Real Estate Registration Act and other laws regarding registration.*

In the above example, the conclusions in both the question and the article use the root word “assert” of the syntactic tree. So, this example can be answered using only the confirming negation and antonym information. If the sum of the negation levels of a question is the same with that of the corresponding article, then we determine the answer is "yes", and otherwise "no".

The negation level is computed as following: if [negation + antonym] occurs an odd number of times in a premise (conclusion), its negation level is “1”. Otherwise if the [negation + antonym] occurs an even number of times, including zero, its negation level is “0”. In the above example, the negation level of the premise of the question is zero, and that of the conclusion of the question is also zero. The negation level of a premise of the article is one, and that of a conclusion of the article is also one. Since the sum of the negation levels of the question is the same with that of the corresponding article, we determine the answer of the question is "yes”.

Our precise description for this rule is shown in Figure 3. The output of our rule-based system is also used below in an unsupervised learning model for assigning labels of premise (conclusion) clusters for non-easy questions.

2.4 Unsupervised learning for the non-easy questions

For the questions not confirmed as easy, we need to construct deeper representations. Fully general solutions are extremely difficult, if not impossible; for our first approximation to the non-easy cases, we have developed a method using unsupervised learning with more detailed linguistic information. Since we do not know the impact each linguistic attribute has on our task, we run a machine learning algorithm that ‘learns’ what information is relevant in the text to achieve our goal.

The types of features we use are as follows:

Word matching Having the same lemma.
Tree structure features Considering only the dependents of a root.

Lexical semantic features Having the same Kadokawa thesaurus concept code.

We use our learning method on linguistic features to confirm the following semantic entailment features:

Feature 1 : $if\ w_{root}(q_n,\ premise) = w_{root}(article_n,\ premise)$

Feature 2 : $if\ w_{root}(q_n, conclusion) = w_{root}(article_n,\ conclusion)$

Feature 3 : $if\ w_{dep\_i}(q_n,\ conclusion) \in W_{dep}(article_n, conclusion)$

Feature 4 : $if\ c_{root}(q_n, premise) = c_{root}(article_n, premise)$

Feature 5 : $if\ c_{root}(q_n, conclusion) = c_{root}(article_n, conclusion)$

Feature 6: $if\ neg\_level(q_n, premise) = neg\_level(article_n, premise)$

Feature 7: $if\ neg\_level(q_n, conclusion) = neg\_level(article_n, conclusion)$

Features 1, 2, 3 consider both lexical and syntactic information, and Features 4 and 5 consider semantic information. Features 6 and 7 incorporate negation and antonym information. Features 1 and 2 are used to check if premises (conclusions) of a question and corresponding article share the same root word in the syntactic tree. Feature 3 is to determine if each dependent of a root in the conclusion of a question appears in the article. We heuristically limit the number of dependents as those three nearest to the root. Features 4 and 5 confirm if the root words of premises (conclusions) of the question and corresponding article share the same concept code. We use some morphological and syntactic analysis to extract lemma and dependency information. Details of the morphological and syntactic analyzer are given in Section 3.

The inputs for our unsupervised learning model are all the questions and corresponding articles. The outputs are two clusters of the questions. The yes/no outputs of easy questions which have been already obtained are used as a key for assigning yes/no label of each cluster. The cluster which includes higher portion of "yes" of the easy questions is assigned the label "yes", and the other cluster is assigned "no". For the non-easy questions, we determine their yes/no answers following their clustering labels. For the easy questions, we use results of the rule of Figure 3, regardless of the clustering labels of the questions, because the rule produces more accurate answers for easy questions than the clustering output.

3 Experiments

3.1 Experimental setup

In the general formulation of the textual entailment problem, given an input text sentence and a hypothesis sentence, the task is to make predictions about whether or not the hypothesis is entailed by the input sentence. We report the accuracy of our

method in answering yes/no questions of legal bar exams by predicting whether the questions can be entailed by the corresponding civil law articles.

There is a balanced positive-negative sample distribution in the dataset (49.8% yes, and 50.2% no), so we consider the baseline for true/false evaluation is the accuracy when returning always "no", which is 50.2%. Note that other systems that give state-of-the-art performance on RTE use non-comparable techniques such as theorem-proving and logical induction, and often involve significant manual engineering specifically for RTE. It is thus difficult to make meaningful comparisons with the methods employed in our model.

Therefore the basis for our calibration is with the yes/no questions of legal bar exam sentences related to civil laws. The experts (law school students) annotated corresponding articles for each question. The correspondence type of (question, article) can be divided into three categories: The first is (one question, one article), the second is (one question, multiple articles), and the last is (one question, precedence which is not an article). The proportion of (one question, one article) is 25.63% of overall questions, and we target only the first case, which is a one-to-one correspondence between question and article. Our data has 247 questions, with total 1044 civil law articles.

The original examinations are provided in Japanese, and our initial implementation used a Korean translation, provided by the Excite translation tool (http://excite.translation.jp/world/). Because most of our study team members are not proficient in Japanese, we translated the Japanese data into Korean. The reason that we chose Korean is that the characteristics of Korean and Japanese language are similar, and the translation quality between two languages ensures relatively stable performance. In addition, because our study team includes a Korean researcher, we can easily analyze the errors and intermediate rules in Korean. We used a Korean morphological analyzer and dependency parser [11], which extracts enriched information including the use of the Kadokawa thesaurus for lexical semantic information. We use a simple unsupervised learning method, since the data size is not big enough to separate it into training and test data.

We compare our method with SVM, a supervised learning model. Using the SVM tool included in the Weka [27] software, we performed cross-validation for the 247 questions using 7 features explained in Section 2.4. We used a linear kernel SVM because it is popular for real-time applications as they enjoy both faster training and classification speeds, with significantly less memory requirements than non-linear kernels because of the compact representation of the decision function.

Table 4. Performance of our system

Our method	Accuracy (%)
Baseline	50.20
Rule-based model for easy questions	68.36
Rule-based model for all questions	60.02
Unsupervised learning for difficult questions (K-means)	54.62
Unsupervised learning (K-means) for all questions	56.73
Rule for easy questions + unsupervised learning for difficult questions	61.13
Supervised learning (SVM) for all questions	58.01
Supervised learning (SVM) for difficult questions	55.78

Table 5. Error types

Error type	Accuracy (%)	Error type	Accu.(%)
Specific example case	7.45	Paraphrasing	42.55
Exceptional case	8.09	Constraints in premise	28.09
Condition, conclusion mismatch	3.19	Reference to another article	3.19
Etc.	7.45		

3.2. Experimental results

Evaluation of question answering systems is in general almost as complex as question-answering itself. So one must make the choice to consider several features of QA systems in the evaluation process, e.g., query language difficulty, content language difficulty, question difficulty, usability, accuracy, confidence, speed and breadth of domain [12].

Table 4 shows our results. A rule-based model for easy questions showed accuracy of 68.36%, and it covered 117 questions, which is 47.18% of all questions. When we applied the rule-based method for all questions, the accuracy was decreased into 60.02%. We use a K-means clustering algorithm with K=2 for unsupervised learning for the rest of the questions, and it showed accuracy of 54.62%. The overall performance when combining the use of rules and unsupervised learning showed 61.13% of accuracy which outperformed unsupervised learning for all questions, and even SVM, the supervised learning model we use with a linear kernel. According to p-value measures between the baseline and each model in the true/false determination,

all models significantly outperformed the baseline. Since previous methods use supervised learning with syntactic and lexical information, we consider the supervised learning experiment with SVM in Table 4 approximately represents the performance of previous methods.

3.3 Error analysis

From unsuccessful instances, we classified the error types as shown in Table 5. The biggest error arises, of course, from the paraphrasing problem, which should be solved by expert knowledge and much larger corpora. The second biggest error is because of complex constraints in conditions. As with the other error types, there are cases where a question is an example case of the corresponding article, and the corresponding article embeds another article. In further work, we will need to complement our knowledge base with some kind of paraphrasing dictionary, perhaps with the help of experts. We also found cases that indicate the need to do more extensive temporal analysis.

It will be interesting if we compare our performance using Korean-translated sentences with that using original Japanese sentences. We would expect the system using original sentences will show better performance than ours, because there exist no translation errors. As future work, we will construct a Japanese system using paraphrase/synonym/antonym dictionaries for Japanese, and then analyze how the translation affects performance.

3.4 Using PROLEG

Of course the capture of legal concepts and their relationships is central to the improvement of systems such as ours, but the automatic construction of this kind of knowledge is equivalent to the general problem of open information extraction. However, in the legal domain, there are examples of legal representation systems that have already been used to capture some of this knowledge [24-26].

The one we know best is PROLEG [2], which is a PROLOG-based legal reasoning system. A PROLEG program is a general description of a legal reasoning case, which outputs a trace of derivation, and this trace is represented in the form of an argument between plaintiff and defendant. The main function of PROLEG is to capture and simulate the judge's decision process, and a derivation trace is a by-product of legal reasoning performed by a judge in the form of argument. We have constructed PROLEG logics for civil laws, and we intend to use PROLEG in our base system to improve performance by exploiting the deeper legal knowledge captured in PROLEG.

Here follows an example of PROLEG usage. We have a question <18-16-B> and the corresponding civil law article No.333 as follows:

<Question 18-16-B>

In cases where movable X was delivered from person A to person B, and then from person B to person C based on a sale, the transfer of movable, person A can deter movable X as exercise of statutory liens for sale of movables.

<Civil law article 333>

Statutory liens may not be exercised with respect to the movables that are the subject matter of the same after the obligors have delivered those movables to third-party acquirers.

We have the following PROLEG rules and exceptions related with the article 333:

<PROLEG>

```
1. 'effect of statutory lien'(Obligee,Obligor,Third_Party,Object)<=
       'statutory lien over movables'(Obligee,Obligor,Cause).
2.'statutory lien over movables' (Obligee, Obligor,contract ('Sales',Obligee,
   Obligor, Object, T_contract))<=
     contract(Obligee,Obligor,contract('Sales',Obligee,Obligor,Object,T_contract)).
3. exception('effect of statutory lien'(Obligee, Obligor,Third_Party, Object),
   'exception of third party acquirers'(Obligor,Third_Party,Obligee,Object)).
4. 'exception of third party acquirers'(Obligor,Third_Party,Obligee,Object)<=
   contract(Obligor,Third_Party,contract('Sales',Obligor,Third_Party,Object,
   T_contract)),
   delivery(Obligor,Third_Party,contract('Sales',Obligor,Third_Party,Object,T_contract),
   T_delivery).
```

For readability, we express the above PROLEG rules and exceptions using the letters A, B, C, D, E and F.

1) A <= B.
2) B <= C.
3) exception (A, D)
4) D <= E, F.

In rule 3), we have the "exception" meta-predicate which takes two arguments. The former of the arguments is the head of default rule, and the latter is the head of exceptional rule. Then, "exception(A, D)" means "if D, then not A".

We can represent the above question into the following PROLEG:

```
'effect of statutory lien'(personA,personB,personC,movableX) <=
  contract(personA,personB, contract('Sales', personA,personB, movable,t_contract1)),
  contract(personB, personC, contract('Sales', personB, personC, movableX, t_contract2)),
  delivery(personB, personC, contract('Sales', personB, personC, movableX, t_contract2),
     t_delivery),
```

which means A <= C, E, F.

Since we have E and F in the premise of the question, we also have D according to rule 4). Therefore "not A" is derived according to rule 3). Since "not A" contradicts the conclusion of the question, which is "A", the answer of this question is "no". This kind of logical reasoning will likely improve performance.

However, to do this, we need to confirm correspondence between words in a question and predicate names and arguments in PROLEG. To find the corresponding

PROLEG rules and fill the argument variables correctly, we need more extensive natural language processing techniques, including some general information extraction processes like co-reference resolution, query expansion, paraphrasing, synonym dictionary construction, and syntactic graph matching. As we augment our NLP tools, the PROLEG-based text entailment will provide a deeper level understanding of the questions/articles, and improve performance.

4 Related work

W. Bdour et al. [13] developed a Yes/No Arabic Question Answering System. They used a kind of logical representation, which bridges the distinct representations of the functional structure obtained for questions and passages. This method is not appropriate for our task. If a false question sentence is constructed by replacing named entities with terms of different meaning in the legal article, a logic representation can be helpful. However, false questions are not simply constructed by substituting specific named entities, and any logical representation can make the problem more complex. Kouylekov and Magnini [14] experimented with various cost functions and found a combination scheme to work the best for RTE. Vanderwende et al. [15] used syntactic heuristic matching rules with a lexical-similarity back-off model. Nielsen et al. [16] extracted features from dependency paths, and combined them with word-alignment features in a mixture of experts classifier. Zanzotto et al. [17] proposed a syntactic cross-pair similarity measure for RTE. Harmeling [18] took a similar classification-based approach with transformation sequence features. Marsi et al. [19] described a system using dependency-based paraphrasing techniques. All previous systems uniformly conclude that syntactic information is helpful in RTE, and we also use syntactic information combined with lexical semantic information.

There are also many QA studies in the legal field. The first one is ResPubliQA 2009 [20]. It describes the first round of ResPubliQA, a Question Answering (QA) evaluation task over European legislation, proposed at the Cross Language Evaluation Forum (CLEF) 2009. The ResPubliQA 2009 exercise is aimed at retrieving answers to a set of 500 questions. The answer of a question is a paragraph of the test collection. The hypothetical user considered for this exercise is a person interested in making inquiries in the law domain, specifically on the European legislation. There is another system for QA of legal documents reported by Monroy et al. [21]. They experiment by using natural language techniques such as lemmatizing and using manual and automatic thesauri for improving question based document retrieval. In addition, there was a method based on syntactic tree matching [22], and knowledge-based method using a variety of thesaurus and dictionaries [23]. As further research, we can enrich our knowledge base with deeper analysis of data, and add paraphrasing dictionary getting help from experts.

5 Conclusion

We have proposed a method to answer yes/no questions from legal bar exams related to civil law. We construct our own knowledge base by analyzing negation patterns and antonyms in the civil law articles. To make the alignment easy, we first segment questions and articles into premise and conclusion. We then extract deep linguistic features with lexical, syntactic information based on morphological analysis and dependency trees, and lexical semantic information using the Kadokawa thesaurus. Our method consists of two phases. First, we apply our own simple rules for easy questions, and then adopt unsupervised learning for other questions. This achieved quite encouraging results in both true and false determination. To improve our approach in future work, we need to create deeper representations (e.g., to deal with embedded articles and paraphrase), and analyze the temporal aspects of legal sentences. In addition, we will complement our knowledge base with paraphrasing dictionary with the help of experts. We also have access to a logic-based reconstruction of legal rules in a system called PROLEG [2], which we believe can augment our unsupervised learning process with more precise legal information.

Acknowledgements

This research was supported by the Alberta Innovates Centre for Machine Learning (AICML) and the iCORE division of Alberta Innovates Technology Futures.

References

1.D. Merkl, and E. Schweighofer, En Route to Data Mining in Legal Text Corpora: Clustering, Neural Computation, and International Treaties, Proc. of International Workshop on Database and Expert Systems Applications, pp. 465-470, 1997
2. K. Satoh, K. Asai, T. Kogawa, M. Kubota, M. Nakamura, Y. Nishigai, K. Shirakawa, and C. Takano, PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology, LNAI 6797, pp. 153-164, 2011
3.V. Jikoun and M. de Rijke. Recognizing textual entailment using lexical similarity. In Proceedings of the PASCAL Challenges Workshop on RTE, 2005
4.B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In Proceedings of HLT-NAACL, 2006
5.R. Sno, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In Proceedings of HLT-NAACL, 2006
6.D. Das, and N. A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In Proceedings of ACL-IJCNLP, 2009
7.L. K. Schubert, B. V. Durme, and M. Bazrafshan, Entailment Inference in a Natural Logic-like General Reasoner, Proc. of the AAAI 2010 Fall Symposium on Commonsense Knowledge, 2010
8.B. MacCartney, M. Galley, and C. D. Manning, A phrase-based alignment model for natural language inference. In Proceedings of EMNLP, 2008
9.B. MacCartney, and Christopher D. Manning. Natural logic for textual inference. In Proceedings of Workshop on Textual Entailment and Paraphrasing at ACL 2007 .

10.S. Ohno and M. Hamanishi, New Synonym Dictionary. Kadokawa Shoten, Tokyo, 1981
11.Mi-Young Kim, Sin-Jae Kang and Jong-Hyeok Lee, Resolving Ambiguity in Inter-chunk Dependency Parsing, Proc. of 6th Natural Language Processing Pacific Rim Symposium, pp. 263-270, 2001
12.M. Walas, How to answer yes/no spatial questions using qualitative reasoning?, Proc. of the International Conference on Computational Linguistics and Intelligent Text Processing, pp. 330-341, 2012
13.W. N. Bdour, and N.K. Gharaibeh, Development of Yes/No Arabic Question Answering System, International Journal of Artificial Intelligence and Applications, Vol.4, No.1 (51-63), 2013
14.Kouylekov, M. and B. Magnini. Tree edit distance for recognizing textual entailment: Estimating the cost of insertion. In Proceedings of the second PASCAL Challenges Workshop on RTE, 2006
15.Vanderwende, L., A. Menezes, and R. Snow. Microsoft research at rte-2: Syntactic contributions in the entailment task: an implementation. In Proceedings of the second PASCAL Challenges Workshop on RTE .2006
16. Nielsen, R. D., W. Ward, and J. H. Martin. Toward dependency path based entailment. In Proceedings of the second PASCAL Challenges Workshop on RTE, 2006
17.Zanzotto, F. M., A. Moschitti, M. Pennacchiotti, and M.T. Pazienza. Learning textual entailment from examples. In Proceedings of the second PASCAL Challenges Workshop on RTE, 2006
18.Harmeling, S. An extensible probabilistic transformation-based approach to the third recognizing textual entailment challenge. In Proceedings of ACL PASCAL Workshop on Textual Entailment and Paraphrasing, 2007
19.Marsi, E., E. Krahmer, and W. Bosma. Dependency-based paraphrasing for recognizing textual entailment. In Proceedings of ACL PASCAL Workshop on Textual Entailment and Paraphrasing, 2007
20. A. Penas, P. Forner, R. Sutcliffe, A. Rodrigo, C. Forascu, I. Alegria, D. Giampiccolo, N. Moreau, P. Osenova, Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation, Proc. of the cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments, pp. 174-196, 2009
21.A. Monroy, H. Calvo, and A. Gelbukh, NLP for shallow question answering of legal documents using graphs, Computational Linguistics and Intelligent Text Processing, Vo.5449, pp. 498-508, 2009
22. Z. Mai, Y. Zhang, and D. Ji, Recognizing text entailment via Syntactic Tree Matchng, Proc. of NTCIR-9 Workshop meeting, 2011
23.D. A. Arya, V. Yaligar, R. D. Prabhu, R. Reddy, and R. Acharaya, A knowledge based approach for recognizing textual entailment for natural language inference using data mining, International Journal on Computer Science and Engineering, Vol.02, No.06, pp.2133-2140, 2010
24. T. Bench-capon, What Makes a System a Legal Expert?, Legal Knowledge and Information Systems:JURIX 2012, pp. 11-20, 2012
25. M. Alberti, A. S. Gomes, R. Goncalves, J. Leite, and M. Slota, Normative Systems Represented as Hybrid Knowledge Bases, LNAI 6814, pp. 330-346, 2011
26. J. E. Lundstrom, G. Aceto, A. Hamfelt, "Towards a Dynamic Metalogic Implementation of Legal Argumentation" Proc. of ICAIL, pp.91-95, 2011
27. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. 2009