EPiC Series in Computing
Volume 47, 2017, Pages 23–30
COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Multiple Agent Based Entailment System (MABES) for RTE

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

Abstract

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

1 Introduction

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

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

COLIEE (Competition on Legal Information Extraction/Entailment) has been held in order to deal with legal IR and RTE problems. In the IR phase, a legal information processing system needs to find relevant articles for a given query. In the RTE part, the system needs to determine whether the article found in the IR phase entails the query.

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

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

2 Related Works

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

3 Multiple Agent Based Entailment System

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

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

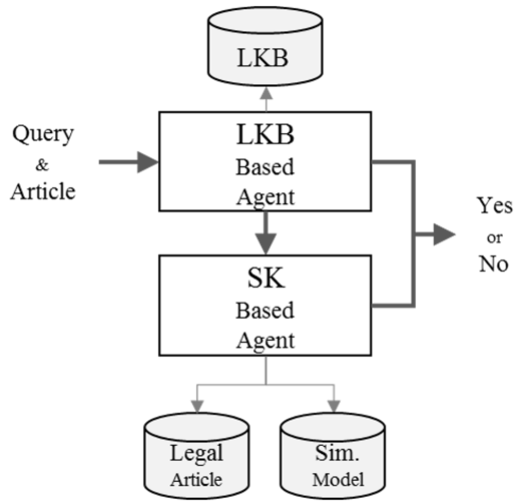

Figure 1: Overall flow diagram of MABES

4 LKB Based Approach

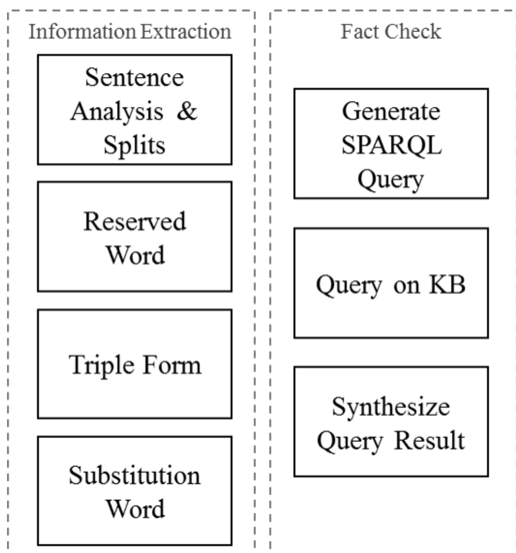

Figure 2: Configuration of LKB based agent

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

4.1 Information Extraction

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

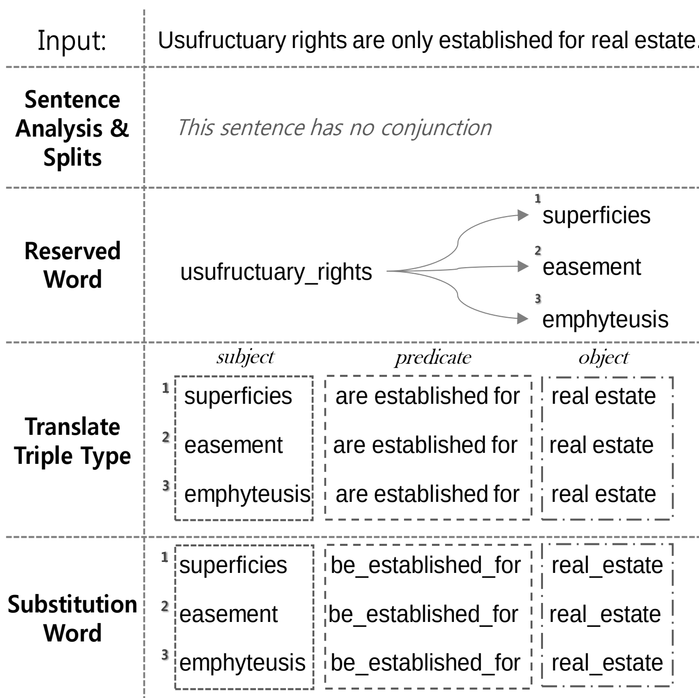

Figure 3: Data processing on the information extraction

4.2 Fact Check

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

5 Syntactic Knowledge Based Agent

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

5.1 Negation Detection

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

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

Figure 4: Equation rule

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

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

Table 1: Negation types

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

5.2 Similarity Analysis

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

6 Experiments and Results

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

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

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

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

6.1 Results

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

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

Correct	Incorrect	Accuracy
334	247	57.5

Table 2: Negation type result

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

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

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

Table 3: Similarity score by models

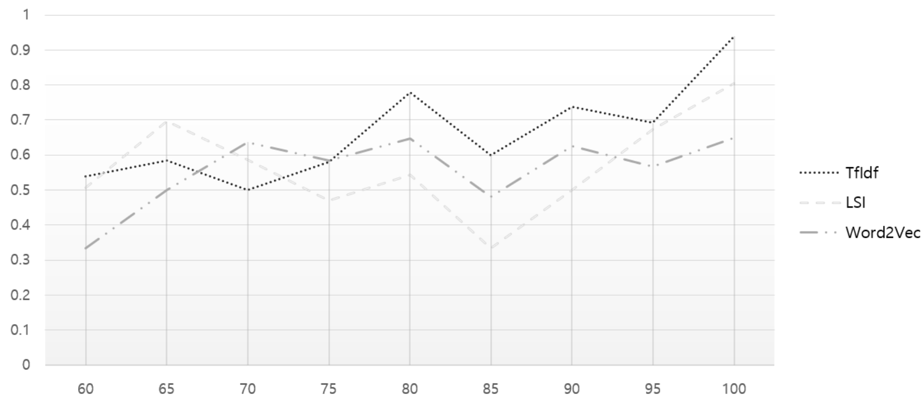

Figure 5: Model similarity and accuracy

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

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

Table 4: Results for training data

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

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

Table 5: Final results

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

7 Conclusion

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

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

References

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

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

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

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

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

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

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

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

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