Statute Law Information Retrieval and Entailment

Mi-Young Kim
Dept. of Science, Augustana Faculty, University of Alberta
Camrose, AB, Canada
miyoung2@ualberta.ca

Juliano Rabelo
Alberta Machine Intelligence Institute
University of Alberta
Edmonton, AB, Canada
rabelo@ualberta.ca

Randy Goebel
Dept. of Computing Science and Alberta Machine Intelligence Institute
University of Alberta
Edmonton, AB, Canada
rgoebel@ualberta.ca

ABSTRACT

Our Yes/No statute law question answering system combines components for both statute law information retrieval and confirmation of textual entailment between statues and legal questions. We describe a statute law question answering system that exploits TF-IDF and a language model for information retrieval, and inter-paragraph entailment. We have evaluated our system using the data from the competition on legal information extraction/entailment (COLIEE-2019). The competition consists of four tasks: Tasks 1 and 2 are for the case law information extraction/entailment, and Tasks 3 and 4 are for the statute law information extraction/entailment. Here we explain our methods and evaluation results for Tasks 3 and 4. Task 3 requires the identification of civil law articles relevant to Japan legal bar exam query. For this task, we used TF-IDF and language model-based information retrieval approaches. Task 4 requires a decision on yes/no answer for previously unseen queries given relevant civil law articles. Our approach compares the approximate meanings of queries with relevant articles. Because many statute law and queries consist of more than one paragraph, we need an inter-paragraph entailment method. Our inter-paragraph entailment process exploits an analysis of statute law structure, and negation patterns to predict entailments. Using our heuristic selection of attributes, we perform two experiments which provide the basis for making a decision on the yes/no questions. One experiment uses an SVM model, and the other uses a general heuristic rule. Our experimental evaluation demonstrates the value of our method, and the results show that our method was ranked No. 1 in both of the Tasks 3 and 4 in COLIEE 2019.

ICAIL '19, June 17–21, 2019, Montreal, QC, Canada
© 2019 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6754-7/19/06.
https://doi.org/10.1145/3322640.3326742

CCS CONCEPTS

• **Computing methodologies → Information extraction**;

KEYWORDS

Textual entailment, question answering, information retrieval, legal AI

ACM Reference format:

Mi-Young Kim, Juliano Rabelo, Randy Goebel. 2019. Statute Law Information Retrieval and Entailment. In *Proceedings of ICAIL 2019: The 17th ACM International Conference on Artificial Intelligence and Law (ICAIL 2019).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3322640.3326742

1 Task Description

Our approach to statute law yes/no question answering combines information retrieval and textual entailment. We achieve this combination with a number of intermediate steps. For instance, consider the question "Is it true if the adjudication of disappearance is made with respect to *A*, who was aboard a vessel which later sank, *A* is deemed to have died upon elapse of one year after the sinking accident?" Our system must first identify and retrieve relevant legal statutes. Then, it must compare semantic connections between the question and the relevant sentences, and confirm a threshold of evidence about whether an entailment relation holds.

The Competition on Legal Information Extraction/Entailment (COLIEE) 2019 focuses on two aspects of statute law and the information processing related to answering yes/no questions from legal bar exams: first is statute law retrieval (Task 3), and second is whether there is a textual entailment relation between a query and relevant statute laws (Task 4).

In Task 3, the goal is to retrieve relevant Japan civil law statutes that are most closely related to a question in legal bar exams, from which we can confirm a 'yes' or 'no' answer based on deciding if there is an entailment relation between the question and the relevant statutes.

We approach the information retrieval part of Task 3 with two models based on statistical information. One is the TF-IDF model [1], i.e., term frequency-inverse document frequency. The idea is that relevance between a query and a document depends on their intersecting word set. The importance of words is measured with a function of term frequency and document frequency as parameters. We do lemmatization and stop word filtering as a pre-processing step.

Another popular model for text retrieval is language model-based information retrieval [2]. This model uses the idea that a common suggestion for coming up with good queries is to think of words that would likely appear in a relevant document, and to use those words as the query. The language modeling approach to information retrieval directly models that idea: a statute is a good match to a legal bar exam query if the statute model is likely to generate the query, which will in turn happen if the statute contains many of the query words.

The goal of Task 4 is to construct yes/no question answering systems for legal queries, by heuristically confirming elements of a query from relevant statutes. The answer to a question is typically determined by measuring some kind of semantic similarity between question and answer. Because the legal bar exam query and relevant statutes are complex and varied, we need to carefully determine what kind of information is needed for confirming textual entailment, and which of that information is easily extracted. Here we exploit the structure analysis on the query and relevant statutes, and investigate the way of inter-paragraph entailment. In addition, we detect negations. We then construct two models: an SVM model using the constructed features, and a heuristic rule-based model.

The remainder of the paper is organized as follows. Section 2 presents previous work on information retrieval and textual entailment. Section 3 explains the details of our method. Section 4 describes the data used for our experiments and presents experimental results which demonstrate the performance of our method. Finally, we provide our conclusions.

2 Related Work

Question answering (QA) with textual corpora is typically modeled as first finding a candidate set of passages (sentences) that may contain an answer to a question, followed by an optional candidate re-ranking stage, and then finally an information extraction step to select the answer string. QA systems normally employ an information retrieval (IR) system to produce the initial set of candidates, usually treated as a black box, bag-of-words process that selects candidate passages best overlapping with the content in the question [3].

Recent efforts in corpus-based QA have been focused heavily on re-ranking, or answer sentence selection: i.e., filtering the candidate set as a supervised classification task to single out those that answer the given question. Extensive research has explored employing syntactic/semantic features [4, 5] and recently using neural networks [6]. Here we analyze the query/document structure as a syntactic feature, and also construct a lexical-semantic feature set using a thesaurus.

The PASCAL RTE challenges [7] have played an important role in developing the understanding of the linguistic entailment problem. Due to the small size of these datasets, most earlier approaches relied on hand-designed features and alignment systems [8]. With the advent of large entailment datasets [9], neural network architectures have been developed for the entailment task. However, these datasets were designed in isolation from any end task and with synthesized sentences. As a result, while they help advance our understanding of entailment, they do not necessarily capture entailment queries that naturally arise in an end task [10].

With regard to using linguistic structure, deep learning entailment models mainly rely on generating a single vector representation for each of the premise and the hypothesis, sometimes using attention features between the sentences [11, 12]. Few models have incorporated sentence structure from both premise and hypothesis, to help improve these representations. Our proposed model explicitly uses the sentence structure of legal statutes and bar exam queries for the linguistic entailment.

3 COLIEE Task 3: Statute Law Retrieval

Our information retrieval models are the term frequency-inverse document frequency (TF-IDF) model and language model. We will describe the two components in the following.

3.1 TF-IDF model

One of our models is a TF-IDF model. The TF-IDF score of a statute *S* related to a query *Q* is:

$$tf-idf(Q,S)=\sum_{t}[\sqrt{(tf(t,S)}\times(1+\log(idf(t)))^{2}]$$

First, for every term *t* in the query *Q*, we compute *tf(t,S)* and *idf(t)*. The score *tf(t,S)* is the term frequency of *t* in the statute *S*, and *idf(t)* is the inverse document frequency of term *t*, which is the number of statutes that contain *t*. The final score is the sum of the scores of terms in both the statutes and the query. The bigger *tf-idf(Q,S)*, the higher the relevance between the query *Q* and the statute *S*.

The choice of terms in documents is as important as choosing the score functions. Instead of using the original words in a text, we do lemmatization and stop word filtering as a pre-processing step.

3.2 Language model-based statute retrieval

The language modeling approach to information retrieval captures the following idea: a statute is a good match to a query if the statute model is likely to generate the query, which will in turn happen if the statute contains the query words often. This approach is an alternative perspective on some of the basic ideas for document ranking. Instead of overtly modeling the probability *P(R=1|q,s)* of relevance of a statute *s* to a query *q*, as in the traditional probabilistic approach to information retrieval, the basic language modeling approach instead builds a probabilistic language model *Ms* from each statute *s*, and ranks documents based on the probability of the model generating the query: *P(q|Ms)*.

We estimate $\hat{p}(Q|M_s)$, the probability of the query Q given the language model of statute *s* as follows:

$$\hat{p}(Q \mid Ms) = \prod_{w \in Q} \hat{p}(w \mid Ms) \times \prod_{w \notin Q} 1.0 - \hat{p}(w \mid Ms)$$

The first term is the probability of generating words in the query and the second term is the probability of not generating other terms. The detailed probabilities for $\hat{p}(Q|M_s)$ are defined as follows:

$$\hat{P}(w \mid Ms) = \begin{cases} \hat{p}_{ml}(w,S)^{(1.0-\hat{R}_{w,S})} \times \hat{p}_{avg}(w)^{\hat{R}_{w,S}} & \text{if } wf_{(w,S)} > 0 \\ \dfrac{c_{wt}}{cs} & \text{otherwise} \end{cases},$$

$$\hat{R}_{w,S} = \left(\frac{1.0}{(1.0+\bar{f}_w)}\right) \times \left(\frac{\bar{f}_w}{(1.0+\bar{f}_w)}\right)^{wf(w,S)},$$

$$\hat{p}_{ml}(w,S) = \frac{wf_{(w,S)}}{dls},$$

$$\hat{p}_{avg}(w) = \frac{\left(\sum d_{(w \in S)} \hat{p}_{ml}(w,s)\right)}{df_w}.$$

$\hat{p}_{ml}(w|M_s)$ shows the maximum likelihood estimate of the probability of term *w* under the term distribution of statute *S*, where *wf(w, S)* is the raw term frequency of term *w* in statute *S*, and *dls* is the total number of tokens in statute *S*. c_{wt}/cs is the background probability for a statute that is missing one or more of the query terms, since we do not want to assign 0 for $\hat{p}_{ml}(w|M_s)$ of this statute, where c_{wt} is the raw count of term *w* in the collection and *cs* is the total number of tokens in the statute collection. $\hat{p}_{avg}(w)$ is the estimate of the probability of the word *w* from a larger volume of data. $\hat{R}_{w,S}$ is a risk function based on a geometric distribution, selected to benefit from the robustness of the estimator $\hat{p}_{avg}(w)$ and to minimize the risk of using the estimator. $\bar{f}_w$ is the mean term frequency of word *w* in documents where *w* occurs. For more details on each probability, refer to Ponte and Croft [2].

4 COLIEE Task 4: Textual Entailment in Legal Documents

Previous textual entailment tasks, such as the PASCAL textual entailment challenge, are inter-sentence entailment. In other words, those tasks require confirmation of entailment from one sentence to another. But in the COLIEE Task 4, we have to predict entailment from a statute to a query, and a statute and a query consist of many sentences/paragraphs. Our system uses structure analysis of statutes/queries and negation to predict inter-paragraph textual entailment. We will look at the entailment types and extract features from sentences, as described in detail in the next subsections.

4.1 Textual entailment procedure

The overall description of our procedure for textual entailment is as follows.

1. Analyze structures of paragraphs in both statute and query
2. Capture the structure of a statute which is related to the query
3. Detect negation
4. Perform two kinds of processing: heuristic rule-based entailment, and machine learning-based entailment

In the following subsections, we explain each step in detail.

4.2 Analyze structures of paragraphs in statute and query

Currently, most of the entailment studies are between single sentences. However, in our task, a statute usually consists of many paragraphs, and a query consists of multiple sentences. To perform inter-paragraph entailment, we analyze the structure of statute/query, and filter out part considered unnecessary to confirm entailment.

From the analysis of the structure of statute, we extract components based on the rules in [13]:

$$\begin{aligned} &conclusion := segment_{last}(sentence, keyword), \\ &condition := \sum_{i \neq last} segment_i(sentence, keyword), \\ &exception\_conclusion := segment_{last}(sentence, exception\_keyword), \\ &exception\_condition := \sum_{i \neq last} segment_i(sentence, exception\_keyword) \end{aligned}$$

From keywords of a condition, we segment sentences. The keywords of the condition are as follows: "in case(s)", "if", "unless", "with respect to", "when", "even if" and ",(comma)." After this segmentation, the last segment is considered to be a conclusion, and the rest of the sentence is considered as a condition. We used the symbol Σ to denote the concatenation of the segments. We also distinguish segments that denote exceptional cases. Currently, we take the exception_keyword indication as "this shall not apply, if (unless)", and "this shall not apply to".

The original bar law examinations in the COLIEE data are provided in Japanese and English, and our system used a Korean translation, provided by the Excite translation tool and Google Translate. The reason that we use the Korean translation is that we have a team member whose native language is Korean who can easily analyze errors, and we can use a high-performance Korean morpho-syntatic analyzer

Table 1. Examples of negation types

Negation type	example
Negation affix	not, no, less…
Negation words	unreasonable, block, withdraw, cancel, shrink, forbid, prohibit..
Negation concepts	457, n444 …

[14] for the feature extraction. The following is an example of condition and conclusion detection:

> *Article 336*
> *General statutory liens may be asserted against obligees without special security, even if the liens are not registered with respect to the relevant immovable property;provided, however, that this shall not apply to registered third parties.*
> → Structure analysis result:
> **Conclusion:** General statutory liens may be asserted against obligees without special security
> **Condition**: even if the liens are not registered with respect to the relevant immovable property
> **Exception_condition**: registered third parties
> **Exception_conclusion**: this shall not apply to

4.3. Capture the structure of a statute which is related to the query

Between the two segments [condition, conclusion] and [exceptional condition, exceptional conclusion], we determine which is related to the query. We choose the segment which has greater value of *overlap* with the query. The overlap is computed as follows:
overlap = (# of overlapped words between condition(exceptional condition) and query)/(# of words in condition(exceptional condition))

For the *overlap* computation to be more accurate, we filter out stop words and do lemmatization.

4.4 Detect negation

The most important feature in determining semantic equivalence or near-equivalence, is accurate attribution of negation. 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 [15] which has a 4-level hierarchy of about 1,100 semantic classes, as shown in Figure 1. Concept nodes in level L1, L2, and L3 are further divided into 10 subclasses. The Korean morpho-syntactic analyzer that we employed provided Kadokawa thesaurus concept numbers for words of the input sentences. Table 1 shows examples of negation types.

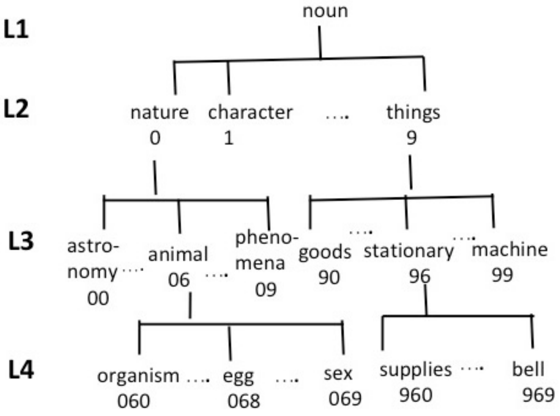

Figure 1. Kadokawa thesaurus hierarchy

4.5 Two approaches for textual entailment

We tried two approaches for textual entailment: one is to use a heuristic rule based on negation and sentence structure, and the other is SVM machine learning using the negation and sentence structure as input features. For the heuristic rule, we used the simple rule of Figure 2.

```
if neg_level(condition of statute)=neg_level(condition of
    query) and neg_level(conclusion of statute)=
    neg_level(conclusion of query),
                Entailment := yes,
otherwise,      Entailment := no,
 where neg_level(segment) := 1 if negation occurs odd
number of times in the segment.
           neg_level(segment) := 0 otherwise
```

Figure 2. Heuristic rule

For the SVM machine learning, we employed a linear kernel and used the following features:

1) Word lemma
2) Kadokawa thesaurus concept number of word
3) Negation feature of each structure segment
4) Structure analysis feature: condition, conclusion

5 Experimental Results

In Tasks 3 and 4, the training data consists of 12 sets of Japan legal bar exam queries from the years 2006 to 2017, with the Japan civil law articles as documents (1044 articles in total). Test data is from the 2018 Japan bar exam queries. The training data has 716 queries and test data has 98 queries. We explain our method's performances in Tasks 3 and 4 in the next subsections.

5.1 Experimental results for Task 3

Here follows one example of the query and a corresponding relevant article from the training data.

<pair id="H25-14-U" label="Y">
<t1>
Article 308
Statutory liens for employer-employee relationships shall exist with respect to salaries and other claims that arose under the employer-employee relationship between the obligor and his/her employee.
</t1>
<t2>
Statutory liens for employer-employee relationships shall exist with respect to salaries and other claims that arose under the employer-employee relationship between the obligor and his/her employee.
</t2>
</pair>

The following shows a query from the test data, for which we need to retrieve the relevant statute number (Task 3).

<pair id='H30-4-A'>
<t2>
In cases an agent is entrusted to perform any specific juristic act, if the agent performs such act in accordance with the instructions of the principal, the principal may not assert that the agent did not know a particular circumstance without negligence which the principal did not know due to his/her negligence.
</t2>
</pair>

The metric for measuring performance of our information retrieval models is the F-measure, with the same weights for both recall and precision. Table 2 presents the results of our two models. The result shows that the TF-IDF achieves better performance than Language model.

Table 2. Performance of our method in Task 3

Method	F-measure	Precision	Recall
TF-IDF	0.5493	0.5918	0.5442
Language model	0.4518	0.4898	0.4473

Table 3 shows the performances of all teams' evaluation results. The performance of our TF-IDF method was ranked No. 1 amongst 13 submitted methods.

Table 3. Performances of all participating teams' methods in Task 3

	F-measure	Precision	Recall
TF-IDF	0.5493	0.5918	0.5442
JNLP-tf	0.5343	0.4592	0.5820
EVORA1	0.5334	0.5714	0.5289
EVORA2	0.5334	0.5714	0.5289
EVORA3	0.5289	0.5714	0.5238
JNLP-tfnv	0.5054	0.4031	0.5616
KIS	0.5032	0.4227	0.6126
KIS_2	0.5030	0.4272	0.6371
DBSE	0.4659	0.4544	0.4932
Language model	0.4518	0.4898	0.4473
iitpBM25	0.4472	0.4898	0.4422
HUKB	0.4140	0.4490	0.4102
iitptfidf	0.4008	0.4388	0.3963

5.2 Experimental results for Task 4

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 are entailed by the corresponding civil law articles.

There is a balanced positive-negative sample distribution in the dataset (52.04% no, 47.96% yes) for the test data of COLIEE 2019, so we consider the baseline for true/false evaluation is the accuracy when always returning 'no', which is 52.04%.

Table 4 shows the experimental results. To translate from Japanese to Korean, we used two translators: Google and Excite. The SVM-based model showed accuracy of 57.14% using Google translate and 66.33% using Excite translation. The heuristic rule showed better performance of 59.18% using Google translate, and 68.37% using Excite translation.

Table 4. Performance of our method in Task 4

Translator	Accuracy with SVM	Accuracy with Heuristic rule
Using Google Translate	0.5714	0.5918
Using Excite Translate	0.6633	0.6837

Table 5 shows the performances of all participating teams' submitted results. Our system using heuristic rule with Excite translation result was ranked No.1 amongst the 15 submitted results.

Table 5. Performances of all participating teams' methods in Task 4

Method	Accuracy
Using heuristic rule with Excite Translate	0.6837
KIS_3module	0.6224
IITP	0.5918
KIS_dic	0.5918
Using heuristic rule with Google Translate	0.5918
KIS_frame	0.5816
DBSE	0.5714
JNLP.t=98	0.5714
TRAttn	0.5612
TRSimFeat	0.5306
JNLP.t=85	0.5204
EVORA1	0.5102
JNLP.t=78	0.4898
EVORA3	0.4796
EVORA2	0.4490

5.3 Discussion

In Task 3, TF-IDF showed better performance than the language model-based information retrieval. In the language model-based information retrieval formulation, for a word *w* that is included in a statute but not included in a query, one assigns a penalty by computing the probability that the language model does not generate *w* based on the term frequency. There is some risk of assigning a penalty based on only term frequency in a document for the words that have different information content, as measured by document frequency. For future work, to assign a small penalty to the words that are common to the legal domain but do not have a large amount of information content (e.g. the words 'cases', 'law', and 'article'), we will try to add a document frequency measure. The document frequency of a word helps to determine its information content: the smaller the document frequency of a word is, the more information content it has.

In Task 4, we could see that the translation tool affected the entailment result, and the Excite translation tool showed better performance of entailment than the Google translation tool. The Google translation tool's translation had many space errors, so the system using the Google translation results had more errors in identifying the structures of the statutes. In addition, our heuristic rule outperformed SVM-based machine learning method. The negation factor was the most important feature in the SVM machine learning, but other information, such as semantic feature using thesaurus concept number and lexical features of words, did not positively contribute to the performance. However, the heuristic rule using only negation feature and query/statute structure showed good performance. While analyzing the errors of the results using our heuristic rule, we found out that the rule did not deal with the cases of specific example queries of statutes. Because there were no common words between statutes and example queries, our system could not detect the matching conditions and conclusions. In future work, we will investigate the semantic feature which can identify some analogical relationship between query and statute even though there are no common words.

6 Conclusion

We have described our most recent implementation for the Computation on Legal Information Extraction/Entailment (COLIEE)-2019. In Task 3 of statute law information retrieval, we implemented a TF-IDF and Language model-based algorithm. TF-IDF model was ranked No. 1 in the COLIEE 2019 competition amongst 13 submitted results.

In Task 4, we showed an inter-paragraph entailment method using structure identification of query/statute and negation detection. Our heuristic rule using structure and negation information achieved the best performance in the COLIEE 2019 competition amongst 15 submitted results.

As future work, we will consider document information in the penalty calculation of the language model-based information retrieval in Task 3, and semantic feature investigation which can analyze the analogy case between query and statute in Task 4.

ACKNOWLEDGEMENTS

This research was supported by the Alberta Machine Intelligence Institute (AMII), and would not be possible without the significant support of Colin Lachance from vLex and Compass, and the teams from Ross Intelligence and Intellicon.

REFERENCES

[1] K. S. Jones, A statistical interpretation of term specificity and its application in retrieval. In: Willett, P. (ed.) Document Retrieval Systems, pp. 132-142. Taylor Graham Publishing, London, UK, UK, 1988.

[2] J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval", Proc. of ACM SIGIR conference on Research and development in information retrieval, pp.206-214, 1998.

[3] T. Chen, and B. V. Durme, 2017, April. Discriminative information retrieval for question answering sentence selection. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers (pp. 719-725).

[4] W. Yih, M. Chang, C. Meek, and A. Pastusiak. 2013. Question answering using enhanced lexical semantic models. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1744–1753. Association for Computational Linguistics

[5] X. Yao, B. V. Durme, C. CallisonBurch, and P. Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 858–867. Association for Computational Linguistics.

[6] W. Yin, H. Schutze, B. Xiang, and B. Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Transactions of the Association of Computational Linguistics, 4:259–272.

[7] I. Dagan, O. Glickman, and B. Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. In MLCW. de Marneffe, M.-C., and Manning, C. D. 2008. The Stanford typed dependencies representation.

[8] I. Androutsopoulos, and P. Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. J. Artif. Intell. Res. 38:135–187.

[9] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In EMNLP.

[10] T. Khot, A. Sabharwal, P. Clark. Scitail: A textual entailment dataset from science question answering. In Thirty-Second AAAI Conference on Artificial Intelligence 2018 Apr 27.

[11] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen. 2017. Enhanced LSTM for natural language inference. In ACL.
[12] A. P. Parikh, O. Tackstrom, D. Das, and J. Uszkoreit. 2016. A decomposable attention model for natural language inference. In EMNLP
[13] M.-Y. Kim, Y. Xu, Y. Lu, and R. Goebel. Legal question answering using paraphrasing and entailment analysis. In Tenth International Workshop on Juris-informatics (JURISIN) 2016.
[14] M-Y. Kim, S-J. Kang and J-H. Lee, Resolving Ambiguity in Inter-chunk Dependency Parsing, Proc. of 6th Natural Language Processing Pacific Rim Symposium, pp. 263-270, 2001
[15] S. Ohno and M. Hamanishi, New Synonym Dictionary. Kadokawa Shoten, Tokyo, 1981