EPiC Series in Computing

Volume 47, 2017, Pages 57–67

COLIEE 2017. Fourth Competition on Legal Information Extraction/Entailment

Analyzable Legal Yes/No Question Answering System using Linguistic Structures

Yoshinobu Kano[1,*], Reina Hoshino[1] and Ryosuke Taniguchi[1]

[1] Faculty of Informatics, Shizuoka University

kano@inf.shizuoka.ac.jp, rhoshino@kanolab.net, rtaniguchi@kanolab.net

Abstract

A central issue of yes/no question answering is usage of knowledge source given a question. While yes/no question answering has been studied for a long time, legal yes/no question answering largely differs from other domains. The most distinguishing characteristic is that legal issues require precise linguistic analysis such as predicates, case-roles, conditions, etc. We have developed a yes/no question answer-ing system for answering questions in a legal domain. Our system uses linguistic analysis, in order to find correspondences of predicates and arguments given problem sentences and knowledge source sentences. We applied our system to the COLIEE (Competition on Legal Information Extraction/Entailment) 2017 task. Our team shared the second place in this COLIEE 2017 Phase Two task, which asks to answer yes or no given a problem sentence. This result shows that precise linguistic anal-yses are effective even without the big data approach with machine learning, rather better in its analyzable design for future improvements.

1 Introduction

Automatic question answering is attracting more interests recently. Due to the increasing expectation to the Artificial Intelligence (AI) technologies, people tend to regard question answering systems as a brand new technology emerged today. However, most successful systems employ rather traditional techniques of question answering which have decades of history (Lin & Pantel, 2001) (Ravichandran & Hovy, 2002) (Yu & Hatzivassiloglou, 2003) (Pinto et al., 2003) (Cui et al., 2005) (Xue et al., 2008) (Bian et al., 2008), including series of shared tasks such as TREC (Voorhees & Harman, 2005), NTCIR (Kando et al., 1999) and CLEF (Braschler, 2001). This paper describes our challenge to the COLIEE 2017 legal bar exam, which asks participants to answer true or not based on the Civil Law Articles, given text drawn from the Japanese legal bar exam.

A variety of algorithms and systems has been proposed for question answering. Typically, these question answering systems used *big data* for answering questions (Kwok et al., 2001) (Etzioni et al.,

K.Satoh, M.-Y.Kim, Y.Kano, R.Goebel and T.Oliveira (eds.), COLIEE 2017 (EPiC Series in Computing, vol. 47), pp. 57–67

t1: （留置権の行使と債権の消滅時効）
第三百条　留置権の行使は、債権の消滅時効の進行を妨げない。
(Exercise of Rights of Retention and Extinctive Prescription of Claims)Article 300
The exercise of a right of retention shall not preclude the running of extinctive prescription of claims.
t2: 留置権者が留置物の占有を継続している間であっても，その被担保債権についての消滅時効は進行する。
Even while the holder of a right to retention continues the possession of the retained property, extinctive prescription runs for its secured claim.

Fig. 1. An example of COLIEE legal bar problem which asks to answer t1 entails t2 or not. The correct answer is “yes” in this example. t1 is not given for the Phase Three before COLIEE 2016 and Phase Two in COLIEE 2017.

2004) (Jeon et al., 2005) (Kanayama et al., 2012). For example, Dumais et al. (Dumais et al., 2002) focused on the redundancy available in large corpora as an important resource. They used this redundancy to simplify their algorithm and to support answer mining from returned snippets. Their system performed quite well given the simplicity of the techniques being utilized.

The now widely known IBM Watson system (D. Ferrucci, 2012) would be considered as a typical example of such a question answering system of the big data approach. The IBM Watson system won in the Jeoperdy! Quiz TV program competing with human quiz winners. The core Watson system employed a couple of open source libraries, including the traditionally well-designed DeepQA system (Ferrucci, 2011) as its skeleton of question answering processing. Because their target domain, the Jeoperdy! Quiz, could ask broad range of questions, they collected a huge amount of knowledge sources from the Internet, etc., extracting relevant knowledge by combining a couple of different natural language processing (NLP) techniques.

Answering university examinations is another example. The Todai Robot project (Arai, 2015) is a challenge to solve Japanese university examinations, focusing towards attaining a high score in the National Center Test for University Admissions, and passing the entrance exam of the University of Tokyo (Todai). Although the Todai Robot project tries to achieve higher scores, their aim is rather to reveal the current performance and limitation of the existing AI technologies, using the examinations as its benchmark, similar to the COLIEE’s legal bar exam task. In contrast to the COLIEE task, the challenge of Todai Robot project includes variety of subjects including Mathematics, English, Japanese, Physics, History, etc. all written in Japanese language. While solving any problem of these subjects could be considered as question answering, some problems require special technologies. For example, Mathematics and Physics require to process formula; Japanese requires to infer emotions of story characters. Solving the History subjects might be considered as rather an extension of the existing question answering issues. The Todai Robot project achieved better scores than the average of the real human applicants in their Mock Exam challenges.

Recognition of textual entailments (RTE or RITE) is another related issue. RTE has been intensively studied for recent days, including shared tasks such as RTE tasks of PASCAL (Dagan et al., 2006)(Giampiccolo et al., 2007), SemEval-2012 Cross-lingual Textual Entailment (CLTE) (Negri et al., 2012), NTCIR RITE tasks (Shima et al., 2011)(Watanabe et al., 2013)(Matsuyoshi et al., 2014), etc. In the third PASCAL RTE-3 task, contradiction relations are included in addition to entailment relations (Giampiccolo et al., 2007). In the RTE-6 task, given a corpus and a set of candidate sentences retrieved by a search engine from that corpus, systems are required to identify all the sentences from among the candidate sentences that entail a given hypothesis. NTCIR-9 RITE, NTCIR-10 RITE2, and NTCIR-11 RITEVal Exam Search tasks (Matsuyoshi et al., 2014) required participants to find an evidence in source documents and to answer a given proposition by yes or no. Research of RTE normally tries to employ logical processing.

As described above, question answering techniques could include logic, reasoning, syntactic and semantic analysis. Many previous related works tried to employ such deeper analyses. However, required techniques more or less differ depending on a target domain.

Another issue is whether the knowledge source needs to be "big data" or not. Regarding the COLIEE's legal problems, required knowledge source can be limited. In this paper, we suggest to use small data in a precise way, rather than to use enormous amount of data as knowledge source. Due to this small data issue, supervised machine learning methods would suffer from insufficient training data. In addition, there are no "similar" problems exist for most of the legal bar exam problems. Therefore, a solver needs to "comprehend" the contents of the knowledge sources. Moreover, it is difficult to analyze why the approaches using machine learning answer so, due to their black box architecture. Rule-based methods would make analyses less difficult, and are especially effective in a limited domain like legal documents.

Based on these thoughts, we built our yes/no question answering system. We aim to create an analyzable and human language processing bases system. We implemented a prototype of this system for the previous COLIEE 2016 task (Taniguchi & Kano, 2016). Although we shared the best score in Phase Two (Kim et al., 2016) (yes/no question answering given an evidence sentence), our system lacked precise analyses which human would do.

Our system does not employ any machine learning as its core. The main method of our system is clause-based analysis where predicates, case-roles, conditions, and negations are considered. We prepared a precise match, a loose match and a rough match. We layered these match modules in order to cover insufficiency of the current natural language technology/databases. We focused on Phase Two, yes/no answering question task of COLIEE 2017. Our system achieved score of the second best team in Phase Two.

We describe related works including datasets of NTCIR RITE challenge, and datasets of previous and this COLIEE tasks in Section 2. These datasets use the same format. Section 3 describes our design of the yes/no question answering system. Section 4 shows our experimental results for this COLIEE 2017 task. We discuss our achievements and limitations in Section 5, mentioning possible future works. We conclude our paper in Section 6.

2 Related Works

2.1 Exam Search Subtask in NTCIR RITEVal

While there were a couple of subtasks in the NTCIR RITE series, we describe the exam search subtask of NTCIR-11 RITEVal because the COLIEE dataset adopted the same format as the RITEVal dataset. RITEVal is an evaluation-based workshop held in 2013, aiming to recognize entailment, paraphrase, and contradiction between sentences, which is a common problem shared widely among researchers of NLP and information access (Dagan et al., 2005) (Giampiccolo et al., 2007).

The entrance exam subtask attempts to emulate human's process of answering entrance exam questions. A system solves multiple-choice questions of real university entrance exams by referring to textual knowledge such as Wikipedia and textbooks. The Entrance Exam subtask provides two types of evaluation challenges. In this paper, we treat the RITE-2 Search Style evaluation, whose explanation is given below. This style of subtask was called FV (Fact Validation) subtask in the RITEVal task. We refer to this RITE-2 Entrance Exam Search Style (ExamSearch) subtask simply as RITEVal in this paper. We only regard Japanese version of the subtask, while there were English and Chinese subtasks.

RITEVal's dataset was developed from the past Japanese National Center Test questions for University Admissions (Center Test). The Center Test asks students multiple-choice style questions.

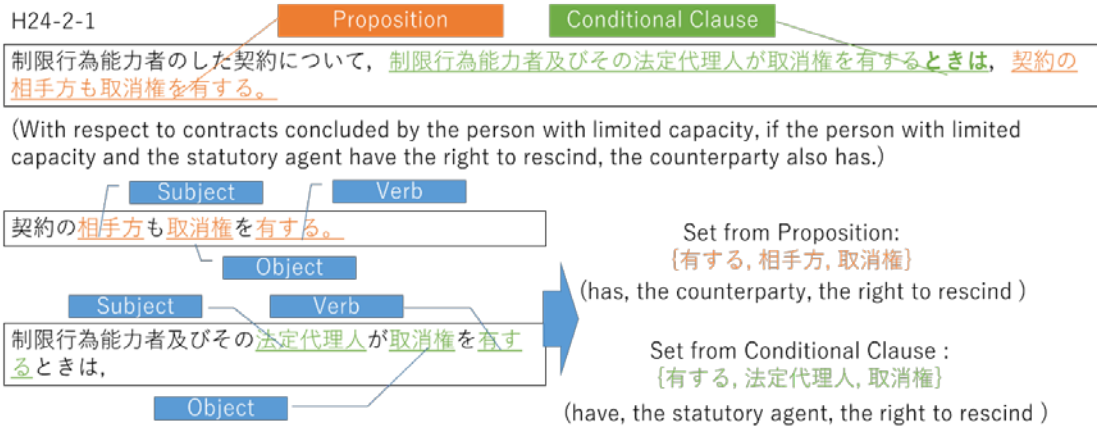

Fig. 2. A conceptual figure of our basic design with an example, showing condition clause, proposition clause, subject, object and predicate.

The RITEVal dataset consists of three types of questions, “select the correct choice” type, “select the wrong choice” type, and “combination” type.

In the RITEVal task, the original multiple-choices were not given as a whole, but given one by one. In “select the correct choice” type questions, given a choice, RITEVal participant systems are asked to return a confidence value for that choice. Evaluation is performed by comparing confidence values for each original multiple-choices, regarding the largest value as the participant system’s answer (smallest in case of “select wrong choice” type questions). In the “combination” type questions, the system is required to label Y or N for each choice and evaluated by a combination of these Y/N w.r.t the original multiple-choice question. In this paper, we focus on the “select correct/wrong choice” type questions.

2.2 JURISIN COLIEE datasets

The COLIEE (Competition on Legal Information Extraction and Entailment) shared task series were held in association with the JURISIN (Juris-informatics) workshops. The first one was the COLIEE 2014 shared task, and the second one was the COLIEE 2015 shared task (Kim et al., 2015). The previous one was the COLIEE 2016 shared task (Kim et al., 2016). This paper mainly describes our participation to the latest COLIEE 2017 shared task.

The COLIEE shared task consists of three phases until COLIEE 2016.

Phase One of this legal question answering task involves reading a legal bar exam question, and extracting a subset of Japanese Civil Code Articles.

Phase Two of the legal question answering task involves the identification of an entailment relationship. Given a question (t2) and a relevant article (t1), a participant’s system has to determine if the relevant articles entail the question or not by answering yes or no.

Phase Three is combination of Phase One and Phase Two. Phase Three requires both of the legal information retrieval system and textual entailment system. Given a set of legal yes/no questions, a participant’s system will retrieve relevant Civil Law articles. Then answer yes/no entailment relationship between input yes/no question and the retrieved articles.

The corpus of legal questions is drawn from Japanese Legal Bar exams, and the relevant Japanese Civil Law articles were also provided.

While there was an English translation version of the dataset provided, we only used the original Japanese version.

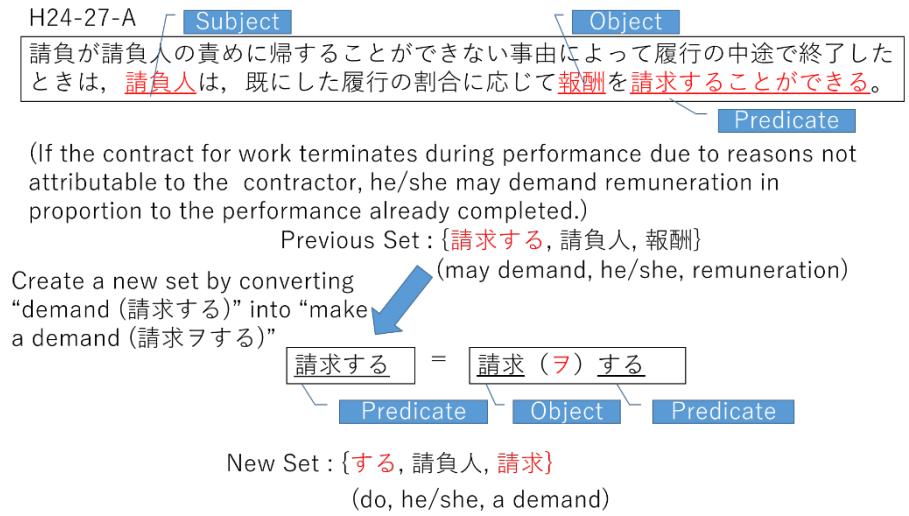

Fig. 3. An example converting an implicit predicate in a noun into an explicit predicate in a verb.

In COLIEE 2017, the original Phase Two was omitted and the original Phase Three is called Phase Two. We use the COLIEE 2017 numbering in this paper.

Fig. 1 is an example of the COLIEE dataset but can also be regarded as an example set of choices in the RITEVal dataset. This example shows the original Phase Two. Therefore, t1, an Civil Law article to be compared, is given. This t1 is not given in the original Phase Three and COLIEE 2017 Phase Two.

3 System Design and Method

We prepared three modules where each module individually outputs Yes/No. Our final output integrates outputs of these modules. We describe details of each module and their integration in this section.

3.1 Common Design Concept

We defined our own *clause* unit (“節”) in order to recognize *condition clauses* and main clauses precisely, which are included in a single sentence. After recognizing conditional clauses and main clauses in a sentence, we compare corresponding clauses between a given question and civil law articles.

A clause should include a predicate as a core element of that clause. We apply a dependency parser that makes chunks (“文節”) of a couple of morphemes. Starting form a chunk that includes a predicate, we aggregate neighboring chunks when a neighboring chunk does not include any predicate.

A predicate is not always suitable to be a core predicate of a clause. For example, *holding* in “condition holding a court” could be regarded as a predicate. However, this is not suitable as a core predicate in a clause because we wish to compare larger predicate-argument structures rather than such a noun phrase.

Fig.2 illustrates our common design described above.

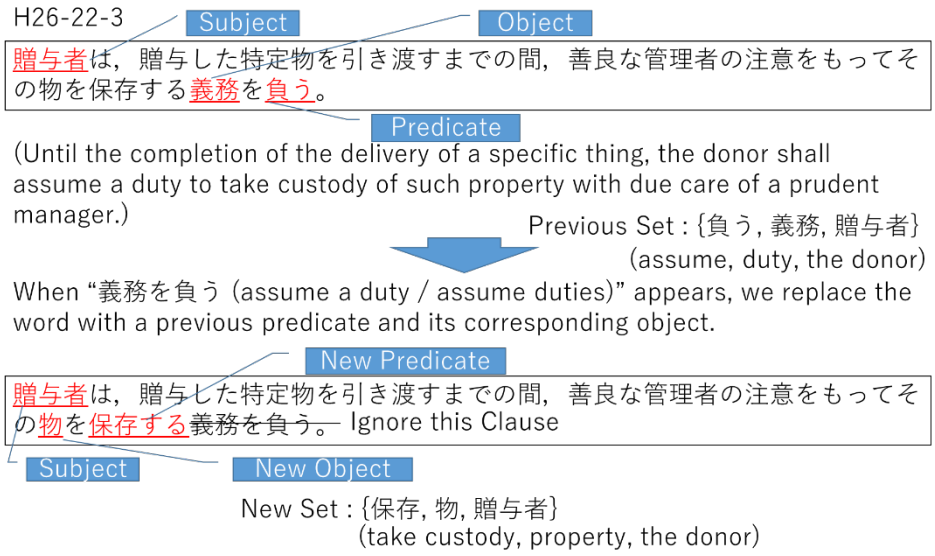

Fig. 4. An example ignoring ineffective expressions.

There are sometimes an implicit predicate which appear as a noun. We convert such an implicit predicate into a normal verb predicate. Fig.3 illustrates this conversion.

Another issue comparing predicates is that there are often ineffective predicates appear in legal documents. In the example of Fig. 4, “… shall assume a duty to …” is such an ineffective expression. We ignore such an expression, and take inner predicate for comparisons.

We define two types of special clauses: proposition clauses and condition clauses. We regard a clause as a condition clause when that clause includes specific patterns e.g. “when …”, “in case of …”, etc. In addition, we used a *force condition clause option* when deciding clause border. If this option is on, a chunk including the specific condition pattern or a previous chunk including a predicate will be forced to be a center of a clause.

3.2 Precise Match

In our precise match module, we extract civil law articles which proposition clauses match with a given problem’s proposition clauses. For each predicate, we find a subject and an object by their case markers. We perform exact matches for predicate itself, its subject and its object. When we could not find any subject nor any object, we skip that sentence. Predicates are compared in their base forms.

When the proposition clauses match in these criteria, we compare condition clauses if any exists. We use the same method for matching condition clauses, i.e. a pair of predicates, subjects and objects.

Finally, we determine a *yes/no* output for each problem; we output *yes* if everything matched, else output *no*. When there is any negation either in problem clause or in article clause, we reverse the *yes/no* output.

3.3 Loose Match

Loose match is a looser version of the precise match. When comparing proposition clauses and condition clauses, we output yes if either subjects or objects are match in addition to matching predicates.

3.4 Rough Match

Rough match is the loosest match in our modules. We only compare predicates of proposition clauses.

3.5 Module Integration

The precise match is the ideal method for us because the precise match tries to imitate human language processing to some extent. However, performance of current natural language processing tools and related databases are not enough to cover required document texts sufficiently, e.g. text structure, hierarchy of lexicon, etc. For this reason, we implemented the three matches as individual modules above, from strict to loose. When a stricter module cannot cover a given specific text, a looser module could augment the stricter module.

We prepared two ways for this integration.

In our first way, filtered integration, we try applying the precise match module first. When the precise match module cannot be applied, we apply the loose match module. If the loose match module cannot be applied as well, we try applying the rough match module.

In our second way, SVM integration, we used SVM (Support Vector Machine) to integrate the modules. Each module outputs a confidence value based on their match result. We trained SVM using these confidence values as training features. We dare designed to loosely integrate our modules rather than to put low level features directly, because we aim to construct an analyzable and human process based system.

4 Experiment and Result

Team ID	Accuracy	Language
iLis7	0.564103	English
iLis9-1	0.576923	English
iLis9-2	0.538462	Japanese
JAISTNLP2-2a-1a-norerank	0.512821	English
JAISTNLP2-2a-1b-rerank	0.474359	English
JAISTNLP2-2b-1a-norerank	0.487179	English
JAISTNLP2-2b-1b-rerank	0.500000	English
JNLP1-R	0.435897	English
JNLP1-RT	0.487179	English
KIS-YN-A	**0.538462**	Japanese
KIS-YN-CM	**0.538462**	Japanese
KIS-YN-CS	**0.589744**	Japanese
KIS-YN-M	**0.576923**	Japanese
KIS-YN-S	**0.653846**	Japanese
NAIST1	0.615385	Japanese

NAIST2	0.653846	Japanese
NAIST3	0.474359	Japanese
NOR17	0.538462	English
UA-LM	**0.717949**	Japanese
UA-TFIDF	0.692308	Japanese

Table 1. Results of COLIEE 2017 Phase Two formal run. Team IDs with prefix "KIS" are our results. Our results and the best team result are highlighted.

COLIEE 2017 provided a problem set of six years (504 problems) for training, one year for the formal run testing. In the *SVM integration*, we determined parameters of SVM by cross-fold validations of the given training data. Table 1 shows the results of COLIEE 2017 formal run for all teams, where prefix of KIS means our team results.

KIS-YN-CM and KIS-YN-CS uses the *force condition clause option.* KIS-YN-CM and KIS-YN-M use the *filtered integration*, while KIS-YN-S and KIS-YN-CS use the *SVM integration*. Among these options, KIS-YN-S obtained the best score.

As far as we observe in this result, the *force condition clause option* was not effective, showing several points decrease. The *SVM integration* was effective, showing 5-8 points increase.

5 Discussion and Future Work

As the number of formal run submissions are limited, we need analysis on training data as follows.

The precise match module could be applied to around 30% of the problems, while the loose match module could be applied to around 50% of the problems. These observations are just same as we assumed beforehand. We suffer from many missing elements when perform-ing linguistic analyses.

For example, there are many omissions e.g. subjects or objects, especially in case of Japanese expressions. Predicate-argument structures are sometimes implicit and difficult to extract.

Semantic hierarchy of predicates is another important issue unresolved. We need to regard different expressions as same meaning from the yes/no answering point of view. However, it depends on broad context whether a different expressions could be regarded as same mean-ing or not.

We observed in the training data that the confidence values contributed to the evaluation scores. More fine-tuned confidence values might increase the scores. However, humans can normally decide their answers in deterministic ways for such legal domain problems. As we suggested in our system design, current NLP technology performance and database coverage are still far insufficient for a system to work like humans. Although this is true, our result is competitive with other teams who used machine learning as their cores. More precise approach would be meaningful as an extension of our system.

We need to grasp distribution of the problems in order to interpret the results objectively.

We observed several points of fluctuations of evaluation scores between years of problems. Very roughly speaking, the COLIEE dataset includes two types of problems: very easy and very difficult.

The very easy problems have almost same text strings in the Civil Law articles. Such a problem can be solved by superficial word-level methods. For example, we found 6 very easy problems in 77 problems (H24 dataset) by our manual verification. Because 6/77 equals 7.8 points and the random baseline is around 50 points, it would be easily available to achieve around 60 points by any relevant method using string/word based features.

The very difficult problems are really hard to solve, they may include logic, abstraction, complex syntactic structure, etc. We do not believe that any current system could handle such issues in

sufficient performance. Rather it might have captured tendency of problem writers. Our aim was to solve problems in the middle of very difficult and very easy ones. While this aim was achieved to some extent, there are still many important issues remaining unresolved. We would need structured lexical database with the semantic hierarchy, at least for this legal domain.

Because our system is designed in an analyzable way, further analyses are available to find what sort of methods/features were effective. Such a deeper and detailed analyses are required future work.

6 Conclusion

We proposed an analyzable and human language process based legal yes/no question answering system. Our team was second best (0.6538) in the COLIEE 2017 yes/no question task (Phase 2). We aimed to solve problems of middle level difficulty by linguistic analyses. This aim was achieved to some extent, while leaving several important issues, such as lexical semantic ontology, unresolved for future work.

Acknowledgements

This research was partially supported by MEXT Kakenhi and JST CREST.

References

Arai, N. H. (2015). The impact of AI—can a robot get into the University of Tokyo? *National Science Review*, *2*(2), 135–136. doi:10.1093/nsr/nwv011

Bian, J., Liu, Y., Agichtein, E., & Zha, H. (2008). Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 467–476). inproceedings, New York, NY, USA: ACM. doi:10.1145/1367497.1367561

Braschler, M. (2001). CLEF 2000 - Overview of Results. In *Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation* (pp. 89–101). inproceedings, London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=648263.753369

Cui, H., Sun, R., Li, K., Kan, M.-Y., & Chua, T.-S. (2005). Question Answering Passage Retrieval Using Dependency Relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 400–407). inproceedings, New York, NY, USA: ACM. doi:10.1145/1076034.1076103

Dagan, I., Glickman, O., & Magnini, B. (2005). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Dagan, I., Glickman, O., & Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment* (pp. 177–190). inproceedings, Berlin, Heidelberg: Springer-Verlag. doi:10.1007/11736790_9

Dumais, S., Banko, M., Brill, E., Lin, J., & Ng, A. (2002). Web Question Answering: Is More Always Better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research*

and Development in Information Retrieval (pp. 291–298). inproceedings, New York, NY, USA: ACM. doi:10.1145/564376.564428

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., … Yates, A. (2004). Web-scale Information Extraction in Knowitall: (Preliminary Results). In *Proceedings of the 13th International Conference on World Wide Web* (pp. 100–110). inproceedings, New York, NY, USA: ACM. doi:10.1145/988672.988687

Ferrucci, D. (2012). Introduction to "This is Watson." *IBM Journal of Research and Development*, *56*(3.4), 1:1-1:15. doi:10.1147/JRD.2012.2184356

Ferrucci, D. A. (2011). IBM's Watson/DeepQA. *SIGARCH Comput. Archit. News*, *39*(3). Journal Article. doi:10.1145/2024723.2019525

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (pp. 1–9). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1654536.1654538

Jeon, J., Croft, W. B., & Lee, J. H. (2005). Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (pp. 84–90). inproceedings, New York, NY, USA: ACM. doi:10.1145/1099554.1099572

Kanayama, H., Miyao, Y., & Prager, J. (2012). Answering Yes/No Questions via Question Inversion. In *the 24th International Conference on Computational Linguistics (COLING 2012)* (pp. 1377–1391). Mumbai, India.

Kando, N., Kuriyama, K., & Nozue, T. (1999). NACSIS Test Collection Workshop (NTCIR-1) (Poster Abstract). In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 299–300). inproceedings, New York, NY, USA: ACM. doi:10.1145/312624.312730

Kim, M.-Y., Goebel, R., Kano, Y., & Satoh, K. (2016). COLIEE-2016: Evaluation of the Competition on Legal Information Extraction/Entailment. In *Tenth International Workshop on Juris-informatics (JURISIN 2016)*.

Kim, M.-Y., Goebel, R., & Ken, S. (2015). COLIEE-2015: Evaluation of Legal Question Answering. In *Ninth International Workshop on Juris-informatics (JURISIN 2015)*. Keio University, Yokohama, Japan.

Kwok, C. C. T., Etzioni, O., & Weld, D. S. (2001). Scaling Question Answering to the Web. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 150–161). inproceedings, New York, NY, USA: ACM. doi:10.1145/371920.371973

Lin, D., & Pantel, P. (2001). Discovery of Inference Rules for Question-answering. *Nat. Lang. Eng.*, *7*(4), 343–360. article. doi:10.1017/S1351324901002765

Matsuyoshi, S., Miyao, Y., Shibata, T., Lin, C.-J., Shih, C.-W., Watanabe, Y., & Mitamura, T. (2014). Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task. In *the 11th NTCIR (NII Testbeds and Community for information access Research) workshop* (pp. 223–232). inproceedings.

Negri, M., Marchetti, A., Mehdad, Y., Bentivogli, L., & Giampiccolo, D. (2012). Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation* (pp. 399–407). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2387636.2387700

Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and*

development in informaion retrieval (pp. 235–242). New York, NY, USA: ACM. doi:10.1145/860435.860479

Ravichandran, D., & Hovy, E. (2002). Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 41–47). inproceedings, Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1073083.1073092

Shima, H., Kanayama, H., Lee, C., Lin, C., Mitamura, T., Miyao, Y., … Takeda, K. (2011). Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In *NTCIR-9 Workshop* (pp. 291–301). inproceedings.

Taniguchi, R., & Kano, Y. (2016). Legal Yes/No Question Answering System using Case-Role Analysis. In *Competition on Legal Information Extraction/Entailment (COLIEE 2016), Tenth International Workshop on Juris-informatics (JURISIN 2016)*. Yokohama, Japan.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press.

Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.-W., … Takeda, K. (2013). Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In *the NTCIR-10 Workshop* (pp. 385–404). Tokyo, Japan.

Xue, X., Jeon, J., & Croft, W. B. (2008). Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 475–482). New York, NY, USA: ACM. doi:10.1145/1390334.1390416

Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 129–136). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1119355.1119372