Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN 2015)

hosted by JSAI-isAI

Raiosha-Buiding, Keio University, Yokohama,
Kanagawa, Japan
November 17-18, 2015

JURISIN 2015 Workshop Co-Chairs
Takehiko Kasahara
Ken Satoh

Preface

Juris-informatics is a new research area which studies legal issues from the perspective of informatics. The purpose of this workshop is to discuss both the fundamental and practical issues among people from the various backgrounds such as law, social science, information and intelligent technology, logic and philosophy, including the conventional "AI and law" area. JURISIN 2015 is the 9th International Workshop on Juris-informatics, which is held in association with the International Symposia on AI by Japanese Society of Artificial Intelligence (JSAI-isAI).

This year, we have twenty one submissions and each paper was reviewed by at least three program committee members and as a result, nineteen papers will be presented at the workshop.

Papers cover logical inference, ontology, natural language processing, information retrieval, and so on. As invited speakers, we have Giovanni Sartor from University of Bologna, Italy, Shiro Kawashima from Doshisha University, Japan, Do Kwan Jo from National Assembly Law Library, Korea and Miyoung Jin Kim, from Justice Law Firm, Korea. Moreover, we have a joint invited speaker, Phan Minh Dung from AIT, Thailand with the 2nd International Workshop on Argument for Agreement and Assurance (AAA 2015).

Following the previous year, JURISIN have a session on the 2nd Competition on Legal Information Extraction/Entailment (COLIEE 2015) which consists of result report of the competition and five papers for each participants.

JURISIN 2015 Workshop Co-Chairs
Takehiko Kasahara
Ken Satoh
November 17, 2015
Yokohama

Workshop Organization

Workshop Co-Chairs

Takehiko Kasahara	Toin Yokohama University
Ken Satoh	National Institute of Informatics and Sokendai

Steering Committee

Takehiko Kasahara	Toin Yokohama University
Makoto Nakamura	Nagoya University, Japan
Katsumi Nitta	Tokyo Institute of Technology, Japan
Seiichiro Sakurai	Meiji Gakuin University, Japan
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Satoshi Tojo	Japan Advanced Institute of Science and Technology, Japan
Katsuhiko Toyama	Nagoya University, Japan

Advisory Committee

Kevin Ashley	University of Pittsburgh, USA
Trevor Bench-Capon	The University of Liverpool, UK
Tomas Gordon	Fraunfoher FOKUS, Germany
Robert Kowalski	Imperial College London, UK
Henry Prakken	University of Utrecht & Groningen, The Netherlands
John Zeleznikow	Victoria University, Australia

Program Committee

Katie Atkinson	University of Liverpool, UK
Marina De Vos	University of Bath, UK
Randy Goebel	University of Alberta, Canada
Guido Governatori	Data61, Australia
Tokuyasu Kakuta	Nagoya University, Japan
Yoshinobu Kano	Shizuoka University, Japan
Takehiko Kasahara	Toin Yokohama University, Japan
Mi-Young Kim	University of Alberta, Canada
Robert Kowalski	Imperial College London, UK
Masahiro Kozuka	Okayama University, Japan
Nguyen Le Minh	JAIST, Japan
Beishui Liao	Zhejiang University, China
Minghui Ma	Southwest University, Chaina
Makoto Nakamura	Nagoya University, Japan
Katsumi Nitta	Tokyo Institute of Technology, Japan
Paulo Novais	University of Minho, Portugal
Antonino Rotolo	University of Bologna, Italy
Seiichiro Sakurai	Meiji Gakuin University, Japan
Katsuhiko Sano	JAIST, Japan
Ken Satoh	NII and Sokendai, Japan
Akira Shimazu	JAIST, Japan
Hirotoshi Taira	Osaka Institute of Technology, Japan
Fumihiko Takahashi	Meiji Gakuin University, Japan
Satoshi Tojo	JAIST, Japan
Katsuhiko Toyama	Nagoya University, Japan
Minghui Xiong	Sun Yat-sen University, China
Baosheng Zhang	China University of Politics and Law, China
Thomas Ågotnes	University of Bergen, Norway

Additional Reviewers

Trevor Bench-Capon	Danilo Carvalho
Yusuke Miyao	Yasuhiro Ogawa
Phan Minh Dung	Yuping Shen
Hai Long Trieu	Yun Xie
Ying Xu	

Table of Contents

JURISIN 2015 papers

COLIEE-2015 papers

Annotating Legal Documents with Ontology Concepts

Adebayo Kolawole John[1], Luigi Di Caro[1], Guido Boella[1]
[1] Universita degli Studi di Torino, Departimento di Informatica,
Via Pessinetto 12, 10149 Torino, Italy
{kolawolejohn.adebayo}@unibo.it
{dicaro}@di.unito.it
{guido.boella}@unito.it

Abstract. This paper describes a novel task of semantic labeling. The idea exploits ontology in providing a fine-grained conceptual document segmentation and annotation. The proposed system divides documents into semantically-coherent blocks and also performs conceptual tagging for efficient information filtering. The paper presents a promising solution. The proposed task has several applications such as granular information filtering of legal texts, text summarization and information extraction among others and has been evaluated on two tasks i.e. the conceptual tagging of semantic segments in text as well as a fine grained conceptual document IR task with promising result..

Keywords: Conceptual Tagging, Semantic Annotation, Information Retrieval, Text Segmentation

1 Introduction

The increasing use of computer, coupled with the growth of internet and emerging powerful database technologies implies that data accumulates in an unprecedented manner. Fortunately, these data are in their electronic form otherwise called Electronically Stored Information (ESI). With the rising influence of Electronic Discovery (eDiscovery), the field of law (especially in terms of litigation) has tremendously benefitted from the availability and growth of ESI by allowing huge documents to be tendered in law courts for cross examination without the documents being in their physical form[1]. Just like the saying goes, "big data means big headache". The 'big headache' in eDiscovery is technically a scaled-up Information Retrieval (IR) task in which manual classification of documents into either being relevant or producible or not for a civil, criminal or regulatory matter under litigation, is automated [2].

In this paper, we pursue a form of fine-grained information retrieval on legal text. The goal of this research is not to develop an eDiscovery system but rather a system whose output can further improve accuracy of eDiscovery systems as well as other tasks by introducing our ideas which seeks to give more structure to legal texts as well as performing Semantic Annotation (SA) on legal texts for improved search facilities.

We describe our idea in dividing legal documents into semantically coherent blocks as well as tagging such blocks with specific concept(s) (from a pool of concepts from Eurovoc[1]) that describes the meaning of the content of the block. We opine that dividing documents into semantically coherent units and labeling each unit with its inferred concept can aid fine-grained information retrieval. To buttress this, we introduce the idea of semantic scopes to documents. This could be local or global scopes, describing how significant a concept is in representing the meaning of a document. The most significant concept being the global scope concept and the others are the local scope concepts for each document. Information search can thereafter be simplified by varying the query on this global and local scope for documents.

The remaining part of the paper describes the proposed task and a theorized solution. First, we give summary of some related works. Subsequently, we describe our proposal, the methodology as well as evaluation.

2 Background and Related Work

Semantic Annotation formalizes and structures document with well-defined semantics specifically linked to a defined ontology [3]. Generally, annotation can aid structured organization of documents for optimized search. For instance, users may search information by well-defined general concepts that describe the domain of information need rather than use keywords.

Ontology is a formal conceptualization of the world, capturing consensual knowledge [4]. It lists the concepts along with their properties and the relationships that exist between them. An example of a common knowledge resource used in NLP is Wordnet[2], a domain independent knowledge base of over 100,000 concepts in English in which a synset correspond to concept. Relationship between concepts are also well defined e.g. synonymy, antonymy, hyponymy etc. This study uses the Eurovoc concept list. A concept can be annotated using lexicon based annotation or named entity identification. In the former, a dictionary of terms linked to each ontology class is maintained, reducing the annotation task to term matching. The latter uses recognition of named entities, with the entities mapped to the concept that gives their meaning.

Semantic annotation can also be viewed as a classification task in which features are defined for each class and a Machine Learning (ML) classifier is trained to learn and group input document into its category [5][6][7]. Several SA systems have been implemented, for instance GoNTogle [8] uses weighted k Nearest Neighbor (kNN) classifier for document annotation and retrieval. A system widely used in semantic web domain is KIM [3] which assigns semantic descriptions to named entities (NE) in a text. The system is able to annotate and create hyperlinks to NEs inside a text and can then index and retrieve documents using these entities. Regular Expression (RE) has also been used to identify semantic elements in a text [9][10]. It works by

[1] Eurovoc is available at http://eurovoc.europa.eu/.

[2] Wordnet is Available at https://wordnet.princeton.edu/

mapping part of a text related to semantic context and matching the subsequent sequence of characters to create an instance of the concept. Application of these systems includes document retrieval especially in the semantic web domain [11][12]. Eneldo and Johannes [6] performed semantic annotation on legal documents for document categorization. Using Eurovoc concept descriptors on EurLex- a large database of legal documents, a ML classifier was trained for multi-label classification. While their work looks similar to our proposal, there are significant differences. First, their goal was a document categorization task considering and grouping a document as a whole while ours is two-fold, automatic segmentation of legal text into semantically coherent block and conceptual annotation of different segments with its rightful concept(s). Thus our goal leans toward IR task than document categorization. In fact, while they deal with documents, contrarily we deal with document parts.

3 Research Motivation

We hypothesize that conceptual zoning of document into semantically coherent blocks can enhance fine-grained IR and specifically enrich document retrieval systems as in eDiscovery procedures. Inspired by the recent successes in the area of Computational Linguistics (CL), we propose an automatic segmentation and semantic labeling of segmented portions of text. We introduce the idea of semantic scopes showing how important a concept is to a document, with the assumption that such semantic scopes can greatly enrich conceptual querying of big document corpus for IR, once the documents in the corpus are well annotated. The idea can also help in scanning a voluminous legal text for specific part that is of utmost interest to the reader. For instance, in an IR task, the facts needed by a user in a document lie in a small portion of the whole document. Even if such facts are contained within a paragraph, a reader looking for specific information would have to read through the whole text in search of the fact, thus pilfering through 'un-needed' information. With our idea, the portion of a document can be labeled according to its related concept; then the user (or further computational tools) have to only look for the text blocks that is tagged with a specific concept, this greatly speed up information gathering and simplify further, the task of information extraction from such processed text.

To summarize the whole proposal without exploring the technical details, the system aims at achieving these stated goals. First, segmentation of text into semantically coherent blocks[3] is done. We take idea from TextTiling [13][14] which divides text into contiguous, non-overlapping discourse units that correspond to the pattern of subtopics in a text; we advance this approach by incorporating some distributional analysis-informed heuristics (concept zoning[4]) to achieve the task.

Secondly, we extract from a document the text portion(s) which best fit a specific information request based on concept filtering. To achieve this, we motivate the idea

[3] Throughout the paper, we interchangeably use the words segment and block to mean the same thing. Also, concept and class are used interchangeably.

[4] We take idea of zoning from the work of S. Teufel in 'Argumentative Zoning : Information Extraction from Scientific Text' (1999) which divides scientific papers into different sections called zones.

of conceptual scope of a specific document. For instance, a document can be described in terms of its *global* or *local* semantic scope; also, a concept could be local to a document or global to that document. We assume that a global scope concept has a higher argumentative weight in terms of its representation in the document while a local scope concept is lightly referred in terms of its representation in the document. The local or global scope representation further allow some degree of variableness in terms of how the system might be queried for information filtering and retrieval, providing a form of advanced search by enabling users to try different queries in respect of different context or condition and get responsive result sets. For instance, we may be able to vary the search query in terms of a global scope and one or more local scope(s) and get different result based on the local scope concept. As an example, a simple search query like "find all documents that talk about *x* in the context of *y* and not in the context of *z*" becomes possible, where *x* is a global scope category and *y* and *z* are local scope categories. As an example, let us assume that a document refer to three concepts 'public-health', 'Animal-feed' and 'European-standard'. We can also assume that public health has global scope while Animal feed and European standard both have local scope. Using this information, a user might attempt different queries as stated below and get different result:

- Retrieve the document(s) (and their specific parts) where the theme is generally about public health, while also talking about European standard
- Retrieve the documents and their parts where the general topic is about public health in relation to animal feed
- Retrieve the specific part(s) of a document that talks about European standards on animal feed in relation to public health.

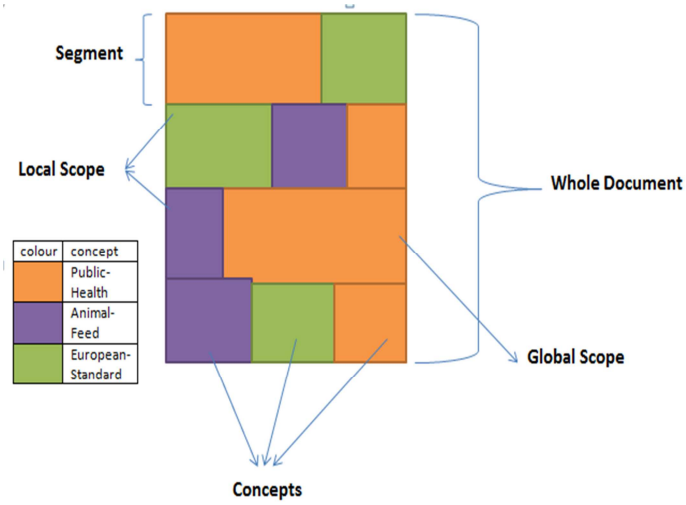


Fig. 1. Concept overlap and Semantic Scoping of Document

The fig. 1 above gives a description of the semantic scoping of document into global and local scopes in terms of each concept's representation of the document.

Relying on a simple statistical procedure as visualized above, we can conclude that *public-health* is the most significant concept since it appears in every blocks. Also, its strength of representation in each of the blocks it appears in is not negligible and thus, it can be labeled as the global concept. Whereas *Animal-feed* and *European standards* also appear in three of the four blocks, their representation in the document as a whole is incomparable to the former. We can also statistically determine the most significant concept that is local to each block as this can enable '*structured in-text scanning*'. Thus, we may for example conclude that *Animal-feed* is global to the fourth block, even though it is itself local to the entire document as shown in the figure.

This makes information filtering very advanced, for instance if a document talks about *Animal feed* but not in the context of *European standards* then it becomes irrelevant for the third query above and it is not retrieved. Also, if for example we have two documents A and B, with A having concepts *Public-Health (X)*, *Animal-Feed (Y)* and *European-Standard* (*Z*) while B contains only concepts *Public-Health* (*X*) and *Animal-Feed (Y)* without European-Standard. Then, queries of the form " SELECT FROM corpus WHERE concept = X , Y AND NOT concept = Z" can make a distinction between these concept overlaps and retrieve only document A while leaving out document B, even though it contains two of the mentioned concepts.

We take a block as an information unit in a document, defined by different levels of granularity (i.e., sentence, paragraph or section holding many paragraphs). Our goal is to assign a label to each specific block with the assigned label corresponding to a specific concept in the ontology. The following processes are carried out:

1. Create concept profiles by looking at frequent and discriminant words calculated over texts-to-concepts occurrences (using TF-IDF).
2. Parse the text segment, sentence by sentence, calculating the similarity of each sentence with the concept profiles. Taking ideas from existing research on text segmentation such as TextTiling [13] which was improved in [14] and topic modeling [15] as well as TextRank [20], we identify the following flags in the text: (a) when the text in a block starts talking about a concept, (b) when it stops talking about the concept, and (c) when it starts talking about a new concept. Reference [16] contains a detail review of text segmentation approaches for the reader's interest.
3. Perform concept association, which maps a contiguous text block to a semantic concept that literarily gives a summary of its content.
4. Identify the global and local scopes of the concepts in a document by analyzing how the related concept block overlaps.

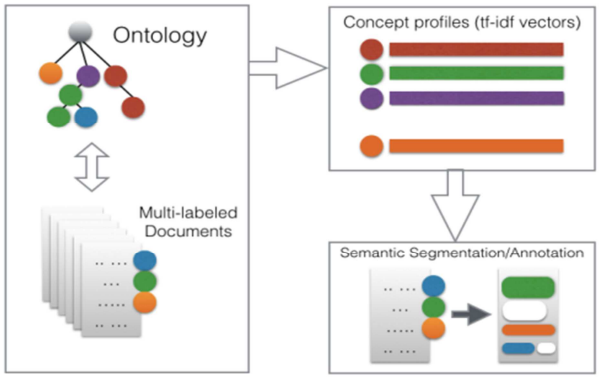


Fig. 2. Schematic representation of the proposed task

3 Methodology

Let us consider a text t_i in a collection T of n = |T| documents where 0 < i <= n. Each document is associated to a set of concepts taken from a thesaurus σ (i.e., a multi-labeled text collection).

Thus we have some concepts $\{C_1, C_2, C_3 \ldots . C_n\} \in \sigma$. We can formalize the task as that of approximating a target function

$$\Phi: B_{n\text{-}1,\,d} \times C\boldsymbol{\lambda} \longrightarrow \{1, 0\}$$

that describes the conceptual mapping of a text segment. Where $B_{n\text{-}1,\,d}$ is a block or segment of text in a document *d* and *n* is a unique incremental number assigned as the identifier for each block. $C\boldsymbol{\lambda}$ is a set of concepts according to a specified ontology. The goal is to find

$$\Phi\,(B_{n\text{-}1,\,d} \times C\boldsymbol{\lambda}) = 1 \qquad (1)$$

Such that $C\boldsymbol{\lambda}i \equiv B_{i,d}$, that is, we want to have a mapping of a block to a concept successfully such that each block is associated to its concept[5]. The equivalence relation above ensures that a concept is attached to one or more blocks of a document *d*. It is possible for text blocks to share same concept as well as a single block being labeled by more than one concept.

The first step is the creation of concept profiles, i.e., numeric vectors representing the contextual meaning of the concepts calculated through a TF-IDF weighting scheme over the concept-term matrix (which is built on the basis of the input multi-labeled document collection). This way, each concept C_p is associated to a vector v_p where *0*

[5] *i* signifies an iterative number, incrementing over the sets of concepts and blocks in a document *d*.

$< p <= |\sigma|$. Then, considering each text t_i as a sequence, $Seq_i = < s_1,s_2,...,s_k >$ of sentences *s*, the idea is to label each sentence with zero or one element belonging to the set of concepts σ.
This automatic segmentation/annotation of a text t_i is done as follows:

- Parse the sequence of sentences Seq_i, then
- For each S_j in Seq_i, calculate the cosine similarity between the frequency vector of S_j and each concept vector as below:

$$Sem_{dist}\left(C_p, S_j\right) = \frac{\vec{a}\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \quad (2)$$

Where

$$\vec{a}\vec{b} = \|\vec{a}\|\left\|\vec{b}\right\| \cos\theta \quad (3)$$

$$\cos\theta = \frac{\vec{a}\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \quad (4)$$

Such that $\vec{a}$ represents the vector of concept C_p with $\vec{b}$ representing the vectors of S_j in the text block.
Then the similarity is obtained by the formula:

$$Sim\left(C_p, S_j\right) = \frac{1}{Sem_{dist}\left(C_p, S_j\right)} \quad (5)$$

- If the similarity between a concept c_p and a sentence s_j is *z* times higher[6] than the rest of concept-to-sentence similarities, then s_j is associated to c_p, otherwise the sentence is not associated to any concept. The parameter *z* is a predetermined threshold value strictly for decision making. While this value can be varied, it is set to 2 by default, making it possible for a sentence to be mapped to its most similar concept.
- Finally, semantically-contiguous sentences (i.e., sentences which are contiguous and associated to a single concept) will represent semantically-coherent segments, which are the final result of the in-text concept annotation task.
- In case the method returns an empty set of segments or an incomplete coverage of the concepts associated to the text t_i, it restarts by step 1 with $z = z/2$.

A schematic view of the task is shown in fig. 2 above.

3.1 Concept Analysis

Concept descriptors could range from unigram, bigrams to n-grams. Similar to query expansion, if a concept descriptor[7] has more than one word, we break the n-gram terms into the constituent words in a process called lexical expansion. The goal of lexical expansion is to retrieve semantically similar words such as synonyms, relying on a knowledge base such as Wordnet. This implies that the document need not contain in explicit terms the constituent words that make up the concept as part of the document's key terms. We used a concept profile, which contains keywords from the

[6] Unique parameter of the method
[7] Take for instance public-health which can be dissolved into public and health.

deflated n-grams, as well as the synonyms of each of the words for each concept. Weights are assigned to each synonym based on their path distance to each of the lexically expanded terms of concept, leading to the selection of the best ranked synsets. We then perform a form of synonym merging on the ranked terms for each concept, which combines these terms by collapsing them into a single block of information unit. With the lexical expansion, vectorization [17][18] is done to build a vector of terms that describe the information content of each concept using.

For simplicity, if σ is a set such that $\sigma = \{c_1, c_2, \ldots, c_k\}$ which is a list of all the concepts for that document. A concept c_i may also have multiple terms e.g. Public-health, each of this, along with its synonyms is a term in the vector space whose frequency of occurrence in each text block is quantified. TF-IDF is used to create vector representations of each concept as well as each text block. Each component of a vector corresponds to the TF-IDF value of a particular term in the text block dictionary. Dictionary terms that do not occur in a block are weighted zero, taking the representation as query vectors that can be compared to vectors of documents (here each text block). The semantic distance between a concept and the text block is calculated using equation 2 and the similarity between a concept and the text bloc is calculated according to equation 5. Iteration is made over each concept in σ, calculating how similar it is to the text block and if similar based on a fixed parameter z, such concept is tagged with that block. The process is repeated for all the blocks in the document which leads to the idea of concept(s)-block tagging. Fig. 3 below shows the general system architecture.

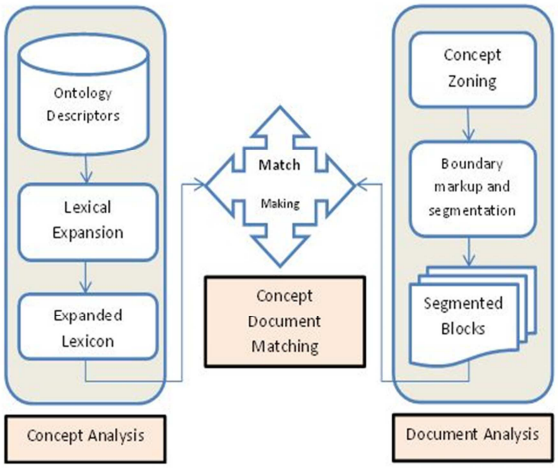


Fig. 3. Proposed System Architecture

3.2 Concept Zoning for Text Segmentation

Given a text t_i and its sequence of sentences $Seq_i = < s_1, s_2, \ldots, s_k >$. We perform concept zoning, which aggregates all the sentences or paragraphs of the text that semantically align into a group. This semantically aligned group is taken as a block,

each of which can be directly mapped to the concept descriptor from an ontology. To segment text into coherent semantic units, we employed word clustering [19]. We used the popular K-means clustering which is based on Lloyd's algorithm [21]. Here, clusters are formed of document parts that appear to be highly similar. First, text is converted into a bag-of-words with each sentence extracted as a micro-document. K-means algorithm is then applied to cluster related words together. Since the input is sentences, K-Means clusters similar sentences. A requirement of K-Means algorithm is the number of clusters to be specified in advance [22]. We here assumed a fixed cluster size of 3. Nested within larger clusters are some clusters containing less similar document segments. Thus, there is a hierarchical relationship between different segments of the document. The hierarchy is useful especially for semantic scoping of concepts such that clusters of document segment(s) that are judged to be strongly similar to each other have nested within it some other clusters containing segment(s) that are less similar. In theory, the single cluster containing the entire collection is represented by the document itself and serves as the root of the tree while the coalesced segments are leaves in the bigger tree. Each cluster is a text segment representing a semantically coherent unit of the document.

3.3 Concept-Document Mapping

Matching a given concept to a block in a text is reduced to a simple semantic relatedness task between the term-document vector of the expanded lexicon of each concept and the bag-of-word vector of each segment. TF-IDF weighing scheme was used while cosine similarity was used to measure the semantic distance between the vectors as explained earlier using the formula

$$Sim\left(C_p, S_j\right) = \frac{1}{Sem_{dist}\left(C_p, S_j\right)} \quad (5)$$

For each of the segment, the system iterates over all the term-document vectors of the expanded lexicon of each concept, measuring the distance. Similar vectors imply some level of relatedness between the concept and the segment and such segment is tagged with the concept.

5 Evaluation

We randomly sampled 5 documents from Eur-Lex database of legal documents. Eur-Lex[8] is an open and regularly updated database of over 3 million EU legal documents, covering EU treaties, regulations, legislative proposals, case-law, international agreements, EFTA documents and some other public documents of interest to EU operations. For each of the documents in the EurLex database, a list of concept(s) describing the document is already listed. We used Eurovoc concept descriptors as ontology. Eurovoc is a multilingual and multidisciplinary thesaurus. Most language versions contain over 6883 prefered concept descriptors and up to 10,592 non-prefered concept terms, organized hierarchically into 21 domains that is of interest to

[8] Available at http://eur-lex.europa.eu/content/welcome/about.html

EU's parliaments. Currently, it is available in 26 European languages. We evaluated the system on two tasks i.e. conceptual tagging and fine grained information retrieval.

- Conceptual Tagging: This task measures the performance of the system in correctly labeling a text segment with a concept. We measured the performance of the system against annotations from human judgment. To achieve this, all the documents were manually segmented by a volunteer into 3 segments each and then annotated manually with three chosen concepts by two different volunteers. The volunteers' agreement was measured on a low-to-high agreement scale of 0 to 2. 0 point means totally disagreed, 1 point means fairly agreed and 2 point means total agreement. The agreements were rated based on individual's judgment in labeling a text segment with a concept. We use the 3 (arising from cluster size) semantically coherent segments from the proposed system since the manual annotation uses 3 segments for each document. Using the manual annotation as Gold standard, we compared the performance of the system with that of manual annotation using L_1 error metric and got an accuracy of 62%.

- Fine grained IR task: The goal here is to measure the efficiency of the system in terms of its *precision* and *recall* in an IR task. We chose the 5 manually annotated documents by volunteers and perform a simple query search on the document list. For instance, given some concepts as query (*as explained under section 3*) in terms of the concept scopes, we measure responsiveness of the system in retrieving the most wanted document based on the concept used as query. We obtained a precision score of 0.210 and a recall value of 0.384.

4 Conclusion

We have described in this paper, a work-in-progress on a task that involves automatic segmentation of text into semantically-coherent blocks and semantic in-text labeling of text blocks annotated with ontology concepts.

The task employs the use of Eurovoc ontology, a multilingual thesaurus continuously updated by the EU publication's office. The task considers a new approach aimed at enhancing information filtering within text by advancing the classification tasks with semantic annotation. We proposed a formalization of the task and a baseline approach to solve it.

The task has a lot of potential applications in IR, text segmentation, topic modeling and text summarization as well as argumentation mining. For instance, a user who is more interested in a particular information in a document can easily specify such information need through an concept that describes such need and the system is able to extract the specific portion containing the requested information need. This is made possible with semantic tag associated with text portion(s), showing their semantic alignment to each ontology terms. Thus, users need not pilfer through unwanted information. We have proposed an evaluation of the system on two tasks i.e. conceptual tagging and fine grained IR task, benchmarking the system with manual annotations from human and using such manual annotations as Gold standard. The

result obtained shows that the approach is promising but requires a bigger and thorough evaluations to be able to compare with results from existing systems. Subsequent works will provide a detailed experimental analysis and evaluation results of our approach in different context and domain.

Acknowledgments
We acknowledge the efforts of anonymous reviewers.

References

1. EDRM: Electronic Discovery Reference model, http://www.edrm.net/
2. The 2008 Socha-Gelbmann Electronic Survey Report, (2008). http://www.sochaconsulting.com/2008survey.php
3. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM–semantic annotation platform. In *The Semantic Web-ISWC 2003*(pp. 834-849). Springer Berlin Heidelberg..
4. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J. R., & Mylopoulos, J. (2006). Text mining through semi automatic semantic annotation. In *Practical Aspects of Knowledge Management* (pp. 143-154). Springer Berlin Heidelberg.
5. Asooja, K., Bordea, G., Vulcu, G., O'Brien, L., Espinoza, A., Abi-Lahoud, E., & Butler, T. Semantic Annotation of Finance Regulatory Text using Multilabel Classification.(2014).
6. Daelemans, W., & Morik, K. (Eds.). (2008). *Machine Learning and Knowledge Discovery in Databases: European Conference, Antwerp, Belgium, September 15-19, 2008, Proceedings* (Vol. 5212). Springer.
7. Buabuchachart, A., Metcalf, K., Charness, N., & Morgenstern, L. (2013). Classification of Regulatory Paragraphs by Discourse Structure, Reference Structure, and Regulation Type. In *JURIX* (pp. 59-62).
8. Bikakis, N., Giannopoulos, G., Dalamagas, T., & Sellis, T. (2010). Integrating keywords and semantics on document annotation and search. In *On the Move to Meaningful Internet Systems, OTM 2010* (pp. 921-938). Springer Berlin Heidelberg.
9. Laclavík, M., Ciglan, M., Seleng, M., & Krajei, S. (2007). Ontea: Semi-automatic pattern based text annotation empowered with information retrieval methods. *Tools for acquisition, organisation and presenting of information and knowledge: Proceedings in Informatics and Information Technologies, Kosice, Vydavatelstvo STU, Bratislava, part, 2*, 119-129.

10. Laclavik, M., Seleng, M., Gatial, E., Balogh, Z., & Hluchy, L. (2007). Ontology based text annotation-OnTeA. *Frontiers in Artificial Intelligence and Applications*, *154*, 311.
11. Handschuh, S., & Staab, S. (2002, May). Authoring and annotation of web pages in CREAM. In *Proceedings of the 11th international conference on World Wide Web* (pp. 462-473). ACM.
12. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., ... & Zien, J. Y. (2003). A case for automated large-scale semantic annotation. *Web Semantics: Science, Services and Agents on the World Wide Web*, *1*(1), 115-132.
13. Hearst, M. A. (1993). *TextTiling: A quantitative approach to discourse segmentation.* Technical report, University of California, Berkeley, Sequoia.
14. Hearst, Marti A. "TextTiling: Segmenting text into multi-paragraph subtopic passages." *Computational linguistics* 23, no. 1 (1997): 33-64.
15. Riedl, Martin, and Chris Biemann. "Text segmentation with topic models."*Journal for Language Technology and Computational Linguistics* 27, no. 1 (2012): 47-69.
16. Lloret, Elena. "Topic Detection and Segmentation in Automatic Text Summarization." (2009). In Focus Journal
17. Clark, Stephen. "Vector space models of lexical meaning (to appear)."*Handbook of Contemporary Semantics. Wiley-Blackwell, Oxford* (2014).
18. Peter D. Turney and Patrick Pantel, 2010. From Frequency to Meaning : Vector Space Models of Semantics, Journal of Artificial Intelligence Research 37, Pg 141-188.
19. Pantel P, Lin D, 2002. Discovering word senses from text. In Proc 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pg. 613-619, Edmoton, Canada.
20. Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. Association for Computational Linguistics.
21. Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." *Applied statistics* (1979): 100-108.
22. Kanungo, Tapas, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. "An efficient k-means clustering algorithm: Analysis and implementation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, no. 7 (2002): 881-892.

A Belief Revision Technique to Model Civil Code Updates

Ryuta Arisaka

National Institute of Informatics
ryutaarisaka@gmail.com

Abstract. It can happen that a scenario that was not thought of at the time of enforcing a civil code article is discovered later. In case application of the civil code article in the discovered scenario is not in keeping with the intention of the article, it is necessary that the article be updated appropriately. We show that this kind of civil code update that is induced upon reaction to augmentation of knowledge through exception discoveries can be modelled in a belief revision theory. We develop our formal framework, and show one instantiation of the framework with case application of a civil code article.

1 Introduction

It can happen that a scenario that was not thought of at the time of enforcing a civil code article is discovered later. In the event that application of the civil code article in the discovered scenario is not in keeping with the intention of the article, it is necessary that the article be updated appropriately.

One past Japanese civil case over lease contract termination is an exemplar. According to the Japanese Civil Code Article 612 which states:

- A lessee may not assign the lessee's rights or sublease a leased thing without obtaining the approval of the lessor [Paragraph 1]; and
- If the lessee allows any third party to make use of or take profits from a leased thing in violation of the provisions of the preceding paragraph, the lessor may cancel the contract [Paragraph 2],

when there is a contract between A as a lessor and B as a lessee, A is granted the right to terminate the contract with B if B has allowed a third party to make use of the leased property or has subleased it without obtaining approval of A for doing so. The Supreme Court ruled (Supreme Court Case:1966.1.27,20-1 Minsyu 136), however, that the

article should really be interpreted as: a lessor A may terminate a contract with a lessee B if B has either transferred B's rights as a lessee or has subleased the leased property to a third party without A's approval, and *if such action as taken by B has undermined A's trust in B so irreparably that it would be no longer possible for A to continue the contract*. In other words, where there is no evidence of the abuse of A's confidence in B, even if B transfers the rights as a lessee or subleases the leased property to a third party, the Article 612 is not applicable, and A will not be granted the legal right to terminate the contract.

An interesting aspect about this civil case is that, while it very much appears to highlight the need for default reasoning, it is not correct to say that the Article 612 normally applies unless there is evidence of no abuse of confidence. That would not justify the point that the Article 612 ceases to apply after the Supreme Court Case wherever there is shown no evidence of abuse of confidence. Rather, we should view it as a particular kind of update on reaction to knowledge augmentation. To expound, let us suppose that enforcement of a civil code article is done at time t_1, and that we stand at some future time t_2. At both t_1 and t_2, some knowledge concerning the civil code article is available, say K_1 and, respectively, K_2. It is reasonable to assume that the intention of the article is adequately expressed in what the article states at time t_1. The adequacy, however, is clearly relative to K_1. Suppose some knowledge, say K_3, then. If for instance K_2 is the summation of K_1 and K_3, it could be that K_3 consists of some scenarios relevant to the article which were not considered at t_1. In that case there is clearly no guarantee that what the article states, which sufficiently characterises its intention under K_1, also characterises it under K_2. And if there should occur any mismatches between the intention of the article and what the article states, updating the article by taking K_3 into account is a right course to resolving them. It is this need that better portrays the above-mentioned civil case example.

In this work, we show that we can model this kind of update on reaction to the discovery of unconsidered scenarios by adapting the principle of the belief revision theory with latent beliefs [2]. Roughly, a belief revision theory [1, 4, 11] is a theory that minimally updates a consistent database upon addition of new data or removal of existing data into another consistent database. Each datum is a formal expres-

sion, and the minimality is measured (ensured) by a set of conditions. As in [2] we will have two types of data. One of them will comprise civil code articles and presented facts, both represented as sentences in first-order logic, whereas the other will comprise code article update instructions which are triggered only if some trigger condition (which is a fact) logically follows from the combination of the civil code articles and the presented facts. We will: formalise our belief revision technique (in Section 2); show an instantiation of our framework with the example of the Article 612 we touched upon informally in this section (in Section 3); and compare our approach with relevant works in the literature (in Section 4), before drawing conclusions.

2 A Belief Revision Theory on Code Articles, Facts and Exceptions

We make use of first-order logic with equality but without function symbols of arity greater than or equal to 1, specifically the following languages $\mathbf{K}$ with or without a subscript ($\mathbf{K}, \mathbf{K_1}, \mathbf{K_2}, \ldots$) consisting of: (1) a fixed set of logical symbols, which we denote by **Log**; and (2) a finite number of non-logical symbols, which we denote by **NonLog** with or without a subscript.
The fixed **Log** contains the following items.

- $\forall, \exists, \wedge, \vee, \supset, \neg, \top$. These are the usual first-order logic symbols. $\supset$ is material implication.
- An equality symbol $=$.
- Parentheses, brackets, and punctuation symbols.
- An infinite number of variables, each of which is referred to by x with or without a subscript.

A **NonLog** contains the following sets:

- **Const**: consists of a finite number of finite sequences of English letters in Italic font beginning with a small letter; e.g. *lessor*, *lessee*. Each such sequence of letters is called a constant.
- **Pred**: consists of a finite number of finite sequences of English letters in sans-serif font beginning with a small letter; e.g. allowUse, sublease with some number of arguments. Each such sequence of letters with n arguments is called a n-ary predicate symbol.

The formulas are defined in the usual way, and the binding precedence, the free/bound variables, the semantics, and the satisfiability follow the standard convention. A formula without a free variable is a sentence. We denote a sentence by F with or without a subscript. We say that F in some language $\mathbf{K}$ is a logical consequence of a set of sentences $F_1, F_2, \ldots$ in the same language iff any model of the set of sentences conjunctively understood is also a model of F. Hereafter, we presume a consequence operator Cn that satisfies the following two conditions.

1. $Cn(\{F_1, F_2, \ldots\})$ contains all the sentences each of which is a logical consequence of $\{F_1, F_2, \ldots\}$, but contains no other formulas (Logical closure).
2. If F is in $Cn(\{F_1, F_2, \ldots\})$, then there exists a finite subset of $Cn(\{F_1, F_2, \ldots\})$ such that F is its logical consequence. (Compactness).

2.1 Belief sets and new information

The belief revision theories [1, 2, 4, 11] define postulates about belief changes, stating how a particular set of beliefs should rationally change as new information is added to it or existing information is removed from it. Let us define a belief set in our context. Let B with or without a subscript denote a pair: $(Cn(\mathbf{Articles} \cup \mathbf{Facts}), \mathbf{Exceptions})$ where

- **Articles** is a finite set of code articles formalised as sentences.
- **Facts** is a finite set of (ultimate) facts [9, 12, 13] again as sentences.
- Both **Articles** and **Facts** are sentences in some language $\mathbf{K}$. However, the sentences that belong to **Articles** are kept disjoint from the sentences that belong to **Facts**.
- **Exceptions** is a finite set of triples $F_1[F_a, F_2]$ for some sentence F_1 in **Articles**, and some sentences F_a and F_2 in some language(s).

Example 1. For an example of B, consider: (1) The Japanese Civil Code Article 612 for **Articles**; (2) the fact that a lessee has subleased a leased house to his friend for **Facts**; and (3) no exceptions for **Exceptions**. Then we can for instance formalise them in the pair: $(Cn(\{F_1\} \cup \{F_a\}), \emptyset)$, where

- F_1 is $[\exists x_1\ \exists x_2 . x_1 \neq x_2 \wedge (\mathsf{allowUseTo}(lessee, x_1, x_2) \vee \mathsf{subleaseTo}(lessee, x_1, x_2)) \wedge \neg \mathsf{gainPermissionFrom}(lessee, lessor)] \supset \mathsf{cancelContract}(lessor, lessee)$.

– F_a is $\mathsf{subleaseTo}(lessee, lesseeFriend, leasedHouse)$.

Now, let mapBtoK be a function such that $\mathtt{mapBtoK}(B) = (\mathbf{Log}, \mathbf{NonLog})$ and that **NonLog** is the set of all predicates and constants that occur in **Articles** or **Facts** of B. We say that **K** is the language of B iff $\mathbf{K} = \mathtt{mapBtoK}(B)$. When we say 'the first component of' or 'the second component of' B, we mean $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, or respectively **Exceptions** of B. The exact manner in which **Exceptions** interacts with $Cn(\mathbf{Articles} \cup \mathbf{Facts})$ will be formally defined later. Nonetheless, the informal meaning of a triple $F_1[F_a, F_2]$ for some F_1, F_a and F_2 is:

1. It is an instruction to update F_1 in **Articles** to F_2 in case F_a is discovered in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$. An informal example: let F_1 be a formalisation of the Japanese Civil Code Article 612; let F_b be the formalisation of the fact that confidence is abused; let F_a be $\neg F_b$; and let F_2 be $F_b \supset F_1$. Then $F_1[F_a, F_2]$ says that, if F_1 is in **Articles**, i.e. the civil code includes the Article 612, then if F_a is in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, i.e. abuse of confidence is discovered, F_1 updates to F_2 on reaction to the discovery, the applicability of the civil code article getting narrowed down.
2. F_a and F_2 do not have to be sentences in the language of B. The rationale behind it is that, while $Cn(\mathbf{Articles} \cup \mathbf{Facts})$ is supposed to be the currently available knowledge, which is hence in the language of B, F_a may be some exceptional fact that could be revealed only at some point in future, if revealed at all. Naturally, the future potential knowledge is not already visible in the current knowledge, and does not have to be expressible in the language of B. Likewise, the emergence of F_2 depends on the visibility of F_a. In case F_a is currently not in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, it does not have to be expressible within the language of B.

Let us list four desirable conditions.

1. If $F_1[F_a, F_2]$ is in **Exceptions** of B, then F_1 is in **Articles** of B (Safety).
2. If $F_1[F_a, F_2]$ is in **Exceptions** of B, then F_a is not in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$ of B (No-unused-triggers).
3. $Cn(\mathbf{Articles} \cup \mathbf{Facts})$ is consistent, i.e. if $F \in Cn(\mathbf{Articles} \cup \mathbf{Facts})$, then $\neg F \notin Cn(\mathbf{Articles} \cup \mathbf{Facts})$, and if

$\neg F \in Cn(\textbf{Articles} \cup \textbf{Facts})$, then $F \notin Cn(\textbf{Articles} \cup \textbf{Facts})$ (Consistency).

4. If $F_1[F_a, F_2]$ is in **Exceptions** of B, then for all $F_1[F_b, F_3]$ also in **Exceptions** of B, if $Cn(\{F_a\}) = Cn(\{F_b\})$ then $F_2 = F_3$ (Uniqueness).

Definition 1 (Belief sets). *We say that a given B is a belief set iff B satisfies (Safety), (No-unused-triggers), (Consistency), and (Uniqueness).*

Let us describe the intuition behind the conditions above. The (Safety) says that in order that we be able to update F_1 to F_2 when a fact F_a is presented/discovered, we must have the code article F_1 that is supposed to be updated already in **Articles**. The intent of the (No-unused-triggers) is to do any update whenever possible. By (Uniqueness) it is guaranteed that every code article can have at most one update trigger per fact at any given moment.[1]

Information to be added to or removed from a belief set B can be a code article, a fact, or an exception. It satisfies the following conditions.

1. If it is either a code article or a fact, say F, then
   (a) F is consistent (Consistency of a code article and a fact).
   (b) F is not a tautology (Non-triviality of a code article and a fact).
2. If it is an exception, say $F_1[F_a, F_2]$, then
   (a) F_2 is consistent (Consistency of the updated article).
   (b) F_2 is not a tautology (Non-triviality of the updated article).

2.2 Belief change postulates

In this subsection we define belief change operations for our belief sets: **ContractBy** for contracting it by some information `Info`; and **ReviseBy** for revising it.[2] Contraction is informally an operation that removes some information off a belief set to derive a new belief set; and revision is an operation that adds new information to a belief

[1] It having at most one update trigger per fact does not mean it having at most one update trigger.

[2] Usually a paper defines either belief revision only, or belief contraction, revision and expansion. The last, expansion, is not touched upon in this work, since there is hardly any point in distinguishing revision from it in our setting.

set, ensuring that the result be a belief set which is as consistent as possible. As some preparation, let $+$ be an operator between some $B = (Cn(\mathbf{Articles} \cup \mathbf{Facts}), \mathbf{Exceptions})$ not necessarily a belief set and some `Info` such that

- $B + \texttt{Info} = (Cn(\mathbf{Articles}_1 \cup \mathbf{Facts}_1), \mathbf{Exceptions}_1)$ where
  - If `Info` is a code article, then $\mathbf{Articles}_1 = \mathbf{Articles} \cup \{\texttt{Info}\}$, $\mathbf{Facts}_1 = \mathbf{Facts}$ and $\mathbf{Exceptions}_1 = \mathbf{Exceptions}$.
  - Else if `Info` is a fact, then $\mathbf{Articles}_1 = \mathbf{Articles}$, $\mathbf{Facts}_1 = \mathbf{Facts} \cup \{\texttt{Info}\}$ and $\mathbf{Exceptions}_1 = \mathbf{Exceptions}$.
  - Else if `Info` is some $F_1[F_a, F_2]$, then,
    - If F_1 is in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, then $\mathbf{Articles}_1 = \mathbf{Articles}$, $\mathbf{Facts}_1 = \mathbf{Facts}$ and $\mathbf{Exceptions}_1 = \mathbf{Exceptions} \cup \{\texttt{Info}\}$.
    - Else if F_1 is not in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, $\mathbf{Articles}_1 = \mathbf{Articles}$, $\mathbf{Facts}_1 = \mathbf{Facts}$ and $\mathbf{Exceptions}_1 = \mathbf{Exceptions}$ (Adequacy).

The (Adequacy) is compatible with the (Safety).

Also, let $\subseteq$, the subset relation, and $\setminus$, the set difference operator, extend to a pair of sets, so that, for any $B_1 = (Cn(\mathbf{Articles}_1 \cup \mathbf{Facts}_1), \mathbf{Exceptions}_1)$ and $B_2 = (Cn(\mathbf{Articles}_2 \cup \mathbf{Facts}_2), \mathbf{Exceptions}_2)$ not necessarily belief sets:

- $B_1 \subseteq B_2$ iff $Cn(\mathbf{Articles}_1 \cup \mathbf{Facts}_1) \subseteq Cn(\mathbf{Articles}_2 \cup \mathbf{Facts}_2)$ and $\mathbf{Exceptions}_1 \subseteq \mathbf{Exceptions}_2$ both hold true; and
- $B_1 \setminus B_2 =$ $(Cn(\mathbf{Articles}_1 \cup \mathbf{Facts}_1) \setminus Cn(\mathbf{Articles}_2 \cup \mathbf{Facts}_2), \mathbf{Exceptions}_1 \setminus \mathbf{Exceptions}_2)$.

Belief contraction is the simpler belief change operation.
$\mathbf{ContractBy}(B, \texttt{Info})$ satisfies all the following axioms.

1. B is a belief set $(Cn(\mathbf{Articles} \cup \mathbf{Facts}), \mathbf{Exceptions})$ for some $\mathbf{Articles}, \mathbf{Facts}$ and $\mathbf{Exceptions}$ (Pre-condition).
2. $\mathbf{ContractBy}(B, \texttt{Info})$ is a belief set (Closure).
3. $\mathbf{ContractBy}(B, \texttt{Info}) \subseteq B$ (Inclusion).
4. If `Info` is not in B,[3] then $\mathbf{ContractBy}(B, \texttt{Info}) = B$ (Vacuity).
5. If `Info` is a triple $F_1[F_a, F_2]$, then

[3] In the sense that if `Info` is a sentence, it is not in $Cn(\mathbf{Articles} \cup \mathbf{Facts})$, and if it is a triple, it is not in $\mathbf{Exceptions}$.

   (a) No $F_1[F_b, F_2]$ such that $Cn(\{F_a\}) = Cn(\{F_b\})$ is in **ContractBy**$(B, \texttt{Info})$ (Success 1).
   (b) $B \subseteq$ **ContractBy**$(B, \texttt{Info}) + \texttt{Info}_1$ where $\texttt{Info}_1$ is $F_1[F_b, F_2]$ for some F_b such that $Cn(\{F_a\}) = Cn(\{F_b\})$ (Reinstatement).
6. If Info is a sentence, then
   (a) Info is not in **ContractBy**$(B, \texttt{Info})$ (Success 2).[4]
   (b) The first component of B is a subset of the first component of **ContractBy**$(B, \texttt{Info}) + \texttt{Info}$ (Partial recovery).
   (c) If $F_x[F_y, F_z]$ for some F_x, F_y and F_z is in the second component of B and if F_x is in the first component of **ContractBy**$(B, \texttt{Info})$, $F_x[F_y, F_z]$ is in the second component of **ContractBy**$(B, \texttt{Info})$ (Preservation).
   (d) **ContractBy**$(B, \texttt{Info}) =$ **ContractBy**$(B, \texttt{Info}_1)$ for any $\texttt{Info}_1$ such that $Cn(\{\texttt{Info}\}) = Cn(\{\texttt{Info}_1\})$ (Extensionality).

Let us provide informal explanations to **ContractBy**. The (Vacuity) says that nothing that is not in B is removed. The (Success 1) and the (Success 2) say that the contraction operation ensures that what is equivalent to $F_1[F_a, F_2]$ up to the logical consequence is not in the result. The (Reinstatement) says that the removal of Info is done minimally, so that there exists a single $F_1[F_b, F_2]$ for $Cn(\{F_a\}) = Cn(\{F_b\})$ which, if added to the result, will completely reinstate all the elements in B. The (Preservation) says that, for any sentence in the result, it has the same set of triples as had in B.

The result can be represented also set-theoretically. Let Δ be a mapping from belief sets and new information into belief sets. For any belief set B and Info, we say that a belief set B_1 satisfying $B_1 \subseteq B$ is a maximal subset of B iff

1. Info is not in B_1.
2. If Info is a triple, then for any belief set B_2, if $B_1 \subset B_2 \subseteq B$, then $B_2 = B$.
3. Else if Info is a sentence, then
   (a) For any $F_1[F_a, F_2]$, if it is in the second component of B and if F_1 is in the first component of B_1, then $F_1[F_a, F_2]$ is in the second component of B_1.

[4] A belief revision theory usually puts an additional condition that Info is not a tautology in order for this condition to apply, which, in our setting, is ensured by (Non-triviality of a code article and a fact).

(b) For any belief set B_2, if $B_1 \subset B_2 \subseteq B$, then `Info` is in B_2.

We define $\Delta(B, \texttt{Info})$ to be the set of all the maximal subsets, provided that B is a belief set. We also define a function γ, so that, if $\Delta(B, \texttt{Info})$ is not empty, then $\emptyset \subset \gamma(\Delta(B, \texttt{Info})) \subseteq \Delta(B, \texttt{Info})$ holds true; or if $\Delta(B, \texttt{Info})$ is empty, $\gamma(\Delta(B, \texttt{Info}))$ is simply B.

Theorem 1 (Representation theorem). *Let B be a belief set. Then it follows that* ***ContractBy***$(B, \texttt{Info}) = \bigcap(\gamma(\Delta(B, \texttt{Info})))$.

Proof. Adaptation of the representation theorem proofs in [1,2] if `Info` is a sentence. Trivial if `Info` is a triple.

Belief revision is an iterating process, similar to the one in [3], which is best described as a procedure or a state transition system.
Description of ReviseBy$(B, \texttt{Info})$
Inputs: a belief set B and some information `Info` (each satisfying the conditions stated earlier).
Output: B_{out}.

L$_0$ Assign B to B_{out}.
L$_1$ If `Info` is a sentence, then assign **ContractBy**$(B_{out}, \neg\texttt{Info}) + \texttt{Info}$ to B_{out}.
L$_2$ If `Info` is a triple $F_1[F_a, F_2]$, and if there exists some $F_1[F_b, F_3]$ in the second component of B such that $Cn(\{F_a\}) = Cn(\{F_b\})$, then assign **ContractBy**$(B, F_1[F_b, F_3]) + \texttt{Info}$ to B_{out}.
L$_{3_0}$ While there exists some $F_1[F_a, F_2]$ in B_{out} such that F_a is in the first component of B_{out}, do:
  L$_{3_{1_0}}$ If $\{F_2\} \cup X$ for X being the first component of **ContractBy**(B_{out}, F_1) is consistent, then;
    L$_{3_{1_1}}$ Assign **ContractBy**(B_{out}, F_1) to B_{out}.
    L$_{3_{1_2}}$ Assign (**ContractBy**$(B_{out}, \neg F_2)) + F_2$ to B_{out}.
  L$_{3_2}$ Else assign **ContractBy**$(B_{out}, F_1[F_a, F_2])$ to B_{out}.
L$_4$ End of While loop

Let us suppose some belief set B, and the new information `Info` to revise B by. The first point of focus (up to Line 2) is to see whether insertion of `Info` into B would: (1) contradict the first component of B, i.e. the first component of B includes $\neg$`Info`, in which case we must minimally change B to B_x in order that $B_x +$ `Info` be consistent in

its first component, which is done by **ContractBy**$(B, \neg\texttt{Info})$; (2) or contradicts (Uniqueness) condition, i.e. `Info` is some triple $F_1[F_a, F_2]$ and the second component of B contains some $F_1[F_b, F_3]$ such that $Cn(\{F_a\}) = Cn(\{F_b\})$, in which case we must remove the existing $F_1[F_b, F_3]$ off B into B_x in order that $B_x + \texttt{Info}$ still satisfy (Uniqueness). Now, for the part from Line 3 to Line 4, it describes the following iterative process. We have B_y. If it is a belief set, our revision is already done, and we simply end this iteration process; otherwise, the set B_y we have does not satisfy (No-unused-triggers), since it satisfies (Safety), (Consistency) and (Uniqueness).[5] This means that there is some $F_1[F_a, F_2]$ in the second component of B_y such that F_a is in the first component of B_y, which means that F_1 in **Articles** may update to F_2. 'May update' because the updating can make the first component of the resulting set inconsistent, which we do not permit and discard $F_1[F_a, F_2]$ off B_y by **ContractBy**$(B_y, F_1[F_a, F_2])$ in that case,[6] but otherwise the update takes place by **ContractBy**(B_y, F_1), to derive B_z in the first component of which F_1 does not appear, followed by **ContractBy**$(B_z, \neg F_2) + F_2$. We regard the resulting set as B_y and repeat the iteration.

Theorem 2. *Let B be a belief set, and* `Info` *new information. Then* ***ReviseBy***$(B, \texttt{Info})$ *is a belief set.*

Proof. We show that $B_{out} = (Cn(\textbf{Articles}_x \cup \textbf{Facts}_y), \textbf{Exceptions}_z)$ satisfies (Safety), (No-unused-triggers), (Consistency) and (Uniqueness).

(Safety) **ReviseBy** is characterised by $+$ and **ContractBy**, both of which guarantee the condition.

(No-unused-triggers) From the condition of the While loop.

(Consistency) Since **ContractBy** ensures consistency and since B is assumed consistent, B_{out} on Line 0 is consistent. B_{out} on Line 1 and Line 2 are consistent, since **ContractBy** ensures consistency. B_{out} on Line 3_{1_1} is consistent for the same reason. B_{out} on Line 3_{1_2} is consistent by the condition of the If on Line 3_{1_0}. Finally, B_{out} on Line 3_2 is consistent. Hence the output of **ReviseBy**$(B, \texttt{Info})$ satisfies Consistency.

[5] See the proof of the theorem right below for details.

[6] We apply this principle on the supposition that when judges see an update necessary, they should have already checked that the update would not contradict the present civil code.

(Uniqueness) Be `Info` a sentence or a triple, on Line 3_0 B_{out} satisfies Uniqueness, due to Line 1 or Line 2. Then the output straightforwardly satisfies Uniqueness, since F_2 on Line 3_{1_2} is a sentence, and, in that case, the second component of $B_{out} + F_2$ is the same as that of B_{out} by the definition of $+$. □

3 The Example Revisited

We instantiate our framework with the example in Section 1. To model the civil code before the civil litigation, we set our language $\mathbf{K}$ to be $(\mathbf{Log}, (\mathbf{Const}, \mathbf{Pred}))$ where

- **Const**: comprises *lessee*, *lessor*, *thirdParty*, and *leasedLand*.
- **Pred**: comprises ternary predicates $\mathsf{allowUseTo}$ and $\mathsf{subleaseTo}$; binary predicates $\mathsf{gainApprovalOf}$ and $\mathsf{cancelContract}$; and unary predicates $\mathsf{isLessee}$, $\mathsf{isLessor}$, $\mathsf{isThirdParty}$ and $\mathsf{isLeasedThing}$.

We then set the initial B (which is $(Cn(\mathbf{Articles} \cup \mathbf{Facts}), \mathbf{Exceptions})$), as follows. $\mathsf{DISTINCT}(x_1, x_2, x_3, x_4)$ is the abbreviation of $\bigwedge_{\text{for all } i,j \in \{1,2,3,4\} \text{ such that } i \neq j} x_i \neq x_j$.

- **Articles**: comprises, for brevity, only the Article 612:
  $\exists x_1\ \exists x_2\ \exists x_3\ \exists x_4.\mathsf{isLessee}(x_1) \wedge \mathsf{isLessor}(x_2) \wedge \mathsf{isThirdparty}(x_3) \wedge \mathsf{isLeasedThing}(x_4)$
  $\wedge\ \mathsf{DISTINCT}(x_1, x_2, x_3, x_4) \wedge (\mathsf{allowUseTo}(x_1, x_3, x_4) \vee \mathsf{subleaseTo}(x_1, x_3, x_4))$
  $\wedge\ \neg\mathsf{gainApprovalOf}(x_1, x_2) \supset \mathsf{cancelContract}(x_2, x_1)$.
  Let this sentence be F.
- **Facts**: is empty.
- **Exceptions**: has $F[\mathsf{p}_8, F_x]$ where
  - p_8 is $\mathsf{noConfidenceAbuseBy}(lessor, lessee)$.
  - F_x models the updated Article 612:
    $\exists x_1\ \exists x_2\ \exists x_3 \exists x_4.\neg\mathsf{noConfidenceAbuseBy}(x_2, x_1) \wedge \mathsf{isLessee}(x_1) \wedge \mathsf{isLessor}(x_2) \wedge$
    $\mathsf{isThirdparty}(x_3) \wedge \mathsf{isLeasedThing}(x_4) \wedge \mathsf{DISTINCT}(x_1, x_2, x_3, x_4) \wedge$
    $(\mathsf{allowUseTo}(x_1, x_3, x_4) \vee \mathsf{subleaseTo}(x_1, x_3, x_4)) \wedge \neg\mathsf{gainApprovalOf}(x_1, x_2) \supset$
    $\mathsf{cancelContract}(x_2, x_1)$.

  .

B hence represents the civil code that has just one article which may update on reaction to the discovery of p_8.

At this point a civil litigation over cancellation of a lease contract begins between two parties. The contract states that 'lessee' is the lessee and that 'lessor' is the lessor who has let a land 'leasedLand' to 'lessee'. There is an evidence that 'lessee' has let a third party, 'thirdParty', use the land.

Let us reflect these. We add $\mathsf{isLeasedThing}(leasedLand)$ (denote it by p_1), $\mathsf{isLessee}(lessee)$ (by p_2), $\mathsf{isLessor}(lessor)$ (by p_3), $\mathsf{isThirdParty}(thirdParty)$ (by p_4), and $\mathsf{allowUseTo}(lessee, thirdParty, leasedLand)$ (by p_5) as well as $\mathsf{DISTINCT}(lessee, lessor, thirdParty, leasedLand)$ (by p_6) into **Facts** of B, deriving a new belief set B_1: $\mathbf{ReviseBy}^{\times 6}((((((B, \mathsf{p}_1), \mathsf{p}_2), \mathsf{p}_3), \mathsf{p}_4), \mathsf{p}_5), \mathsf{p}_6)$, which in this example is the same as $(((((B + \mathsf{p}_1) + \mathsf{p}_2) + \mathsf{p}_3) + \mathsf{p}_4) + \mathsf{p}_5) + \mathsf{p}_6$.

The lessor now proves that the lessee let the third party use the land without his approval.

We have $B_2 = \mathbf{ReviseBy}(B_1, \neg\mathsf{gainApprovalOf}(lessee, lessor))$. At this point, it is true that $\mathsf{cancelContract}(lessor, lessee)$ is in the first component of B_2.

However, judges at the Supreme Court notice an exceptional circumstance in this litigation, which is that there is no evidence that confidence of the lessor in the lessee has been abused by the lessee. They conclude that the civil code article is applicable only if abuse of confidence is evidenced.

Here B_2 is revised by new information $\mathsf{noConfidenceAbuseBy}(lessor, lessee)$, which incidentally is p_8. We have: $\mathbf{ReviseBy}(B_2, , \mathsf{p}_8)$. Notice that the language of B_2, which is still $\mathbf{K}$, does not involve p_8. Once the belief revision takes place, however, we see to it that it becomes $\mathbf{K_1} : (\mathbf{Log}, (\mathbf{Const}, \mathbf{Pred} \cup \{\mathsf{noConfidenceAbuseBy}\}))$, $\mathsf{noConfidenceAbuseBy}$ the p_8 being the newly discovered fact that triggers the update of F to F_x. And because of this, the $\mathsf{cancelContract}$ can no longer be in the first part of the resulting belief set: let B_a be the belief set that is revised by the triple; by the Line 3_{1_1} of **ReviseBy**, contraction of B_a by F takes place, at which point $\mathsf{cancelContract}(lessor, lessee)$ must be removed by (Closure) and (Success 2) of **ContractBy**, and the addition of the updated code article F_x on Line 3_{1_2} does not recover

cancelContract$(lessor, lessee)$ because it is not a logical consequence of F_x in $\mathbf{K_2}$.

4 Related Works and Discussion

Traditional belief revision approaches revise a belief set (in the AGM sense [1]) with new information by possibly removing existing information in contention first, and by then adding the new information. Some argue that revision/contraction by modification rather than by deletion of existing information reflect norm changes more faithfully [5, 10]. In [10] Maranhão proposes a weak belief acceptance model. When new information F_1 is in conflict with existing information, his model accepts $F_2 \supset F_1$ instead of F_1, that way precluding inconsistency in the resulting belief set. He also considers the cases where existing information is instead weakened by a conditional proposition. These refinement operations can be used to accommodate new information as much as possible without contradicting an existing belief set. In [5], Giusto and Governatori consider what they call the minimax revision for belief bases not necessarily closed under the logical consequence relation. They, too, show how propositions can be weakened to avoid inconsistencies. Compared to their works, our approach is in a way more traditional. To update F_1 into F_2 in a belief set in our framework, the belief set is contracted (deleted) by F_1, which is then revised by F_2. However, because we make use of an extended belief set which has not only propositions but also proposition triples as latent update instructions, the two belief change operations: contraction first and then revision, are coupled very tightly as if they were one single belief change operation. In this sense our approach can also be used to model norm changes. Flexibility is ensured by not predestining in what way an existing belief as a civil code article must be changed. Also, that the update instructions do not actively interfere with the propositions in the same belief set means that the AGM postulates are left untouched, which is - at the very least from a compositional perspective - a nice property. We believe that our approach can be adjusted to capture different ways of changing norms such as annulments and abrogations [6, 7], but leave the extension to a future work.

5 Conclusion

We developed a formal framework to handle civil code updates on reaction to the discovery of exceptional circumstances, which we went through with one civil case example. Based firmly on a belief revision theory, our framework guarantees that a civil code update be modelled in a logically reasonable manner.

Acknowledgements

This work would not have existed without insight and helpful comments given to us by Ken Satoh. Proofreading was kindly done by Thomas Given-Wilson. Reviewers helped us improve this paper.

References

1. Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, 50:510–530, 1985.
2. Ryuta Arisaka. How do you revise your belief set with %$;@*? *ArXiv e-prints:1504.05381*, 2015.
3. Ryuta Arisaka. Latent Belief Theory and Belief Dependencies: A Solution to the Recovery Problem in the Belief Set Theories. *ArXiv e-prints:1507.01425*, 2015.
4. Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–29, 1997.
5. Paolo Di Giusto and Guido Governatori. A New Approach to Base Revision. In *EPIA*, pages 327–341, 1999.
6. Guido Governatori, Monica Palmirani, Régis Riveret, Antonino Rotolo, and Giovanni Sartor. Norm Modifications in Defeasible Logic. In *JURIX*, pages 13–22, 2005.
7. Guido Governatori and Antonino Rotolo. Changing legal systems: legal abrogations and annulments in Defeasible Logic. *Logic Journal of the IGPL*, 18(1):157–194, 2010.
8. Guido Governatori, Antonino Rotolo, Francesco Olivieri, and Simone Scannapieco. Legal contractions: a logical analysis. In *ICAIL*, pages 63–72, 2013.
9. Shigeo Ito. *Lecture Series on Ultimate Facts (In Japanese)*. Shojihomu, 2008.
10. Maranhão Juliano. Refinement. *ICAIL*, pages 52–60, 2001.
11. Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.
12. Ken Satoh, Kento Asai, Takamune Kogawa, Masahiro Kubota, Megumi Nakamura, Yoshiaki Nishigai, Kei Shirakawa, and Chiaki Takano. PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology. In *New Frontiers in Artificial Intelligence*, volume 6797, pages 153–164, 2011.
13. Ken Satoh, Masahiro Kubota, Yoshiaki Nishigai, and Chiaki Takano. Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming. In *JURIX*, pages 162–171. IOS Press, 2009.

Using Ontologies to Model Data Protection Requirements in Workflows

Cesare Bartolini[1], Robert Muthuri[2], and Cristiana Santos[3]

[1] University of Luxembourg, Luxembourg,
cesare.bartolini@uni.lu
[2] University of Turin, Italy,
robert.kiriinya@unito.it
[3] Institute of law and technology, University of Barcelona (IDT-UAB),
cristiana.teixeirasantos@gmail.com

Abstract. Data protection, currently under the limelight at the European level, is undergoing a long and complex reform that is finally approaching its completion. Consequently, there is an urgent need to customize semantic standards towards the prospective legal framework. The aim of this paper is to provide a bottom-up ontology describing the constituents of data protection domain and its relationships. Our contribution envisions a methodology to highlight the (new) duties of data controllers and foster the transition of IT-based systems, services/tools and businesses to comply with the new General Data Protection Regulation. This structure may serve as the foundation in the design of present and future information systems abiding to data protection legal requirements.

Keywords: Legal ontology; data protection; General Data Protection Regulation; compliance; business process; BPMN.

1 Introduction

The goal of the privacy and data protection domains of law is to protect the personal information of individuals in a given jurisdiction. With the advent of social media and the uptake of digital technology, the availability of digital services and the soon-to-be Internet of Things have dramatically increased the amount of information collected and processed by governments and companies. Accordingly, businesses are continually developing techniques (such as machine learning, big data analytics, natural language processing) and applications to exploit data assets, to the detriment of new concerns of profiling, identification and re-identification risks.

The European Union (EU) is in the process of upgrading the current data protection law, which is based on the so-called Data Protection Directive (DPD), to a more modern and uniform legislation [33], in accordance with the recent technological progresses. The objective is to enhance individuals' rights, give them more control over their own data, simplify the regulatory environment for

businesses, and set the foundation for the Digital Single Market [15]. The main legislative document of the reform is the General Data Protection Regulation (GDPR), which constitutes the basis for the general protection of personal data. Although the new legislation is in its final stages, it will not be in force before 2018. The text of the GDPR is not finalized yet, and the latest official version released by the Commission dates back to early 2012 (although versions amended by the Parliament and the Council have either been published or leaked to the general public).

The new legislation is set to introduce fines as high as 5% of the annual global turnover (depending on the version of the GDPR) and new inquisitory powers of Data Protection Authorities (DPAs) [25]. Enterprises will then be pressed to avoid infringements, but most of the duties of the data controller are expressed in evaluative terms - one above all, the "appropriate technical and organizational measures" for security (Article 30 of the draft Regulation) - making it difficult for the data controller to know the exact extent of its obligations.

Achieving compliance is no easy task. The transition of organizational and technical measures adopted by businesses would be eased by the existence of standards to adopt, and auditing companies to verify the adherence to those standards. However, no significant standards currently exist for data protection, much less in the light of the upcoming reform. However, in computer science, data protection (often called *privacy*) is considered as a sub-domain of security [29,24]. Indeed, some overlapping exists between the two. For example, some provisions in data protection legislation require that the data processing be performed under appropriate security measures. An early-stage research [6] aims at evaluating the overlapping between the GDPR and security standards, such as the ISO 27000 family, and in particular ISO 27001:2013 [21], to measure the degree of coverage of the data protection rules a security standard would cover. This would help a data controller who adopts a widespread security standard (which relies upon many years of expertise and consolidated auditing firms and methodologies) better understand what is required on their part to achieve GDPR compliance.

Our previous research [7] has introduced a basic ontology of the data protection domain in the context of the GDPR, with a focus on the scope and extent of the duties and obligations of the data controller. The purpose of this paper is to develop on that ontology, laying out the basis of a methodology to extend a workflow model (such as a business process) with annotations that express data protection requirements by means of the proposed ontology. In other words, the ontology will constitute the knowledge base from which the concepts to annotate the workflow model are extracted. Such an approach can provide benefits for a number of stakeholders:

- data controllers would have a clearer view of their duties with respect to data protection in the context of their business;
- the auditors would have a first-look model to assess the GDPR compliance;
- DPAs would have a structured approach to detect potential violations.

The paper is organized as follows. In section 2, we describe the related work concerning domain legal ontologies within data protection and privacy, and busi-

ness processes. section 3 presents the ontology definition, explaining how to describe data protection concepts by means of ontologies and describing the ontology requirements and construction. section 4 portrays a sample extension of business processes using the envisioned legal ontology. Finally, in section 5, we give a set of conclusions and future work.

2 Related Work

"Domain ontologies" in the legal field focus on a particular area of law, but their relevance is constrained by their subject-matter modeling [11] and only some have been applied beyond the prototype stage. Some of the pertinent domain ontologies are briefly mentioned in terms of their purpose, subject-matter, reusability, and availability. Despite efforts in representing data protection domain, according to our best knowledge, there is no ontological representation that specifically addresses the data protection legislation in the light of the reform, as well as the duties of data controllers and the corresponding rights of data subjects.

The LegLOPD ontology [26] was applied for the preservation of privacy in location-based services. It modelled concepts from the Spanish data protection law, reused concepts from the LRI Core, and its ontological development process followed the TERMINAE method. The essential structure to be protected in LegLOPD is the concept of *private data*, derived from an LRI Core abstract concept.

The OntoPrivacy [10] ontology modelled a glossary of keywords from the Italian Personal Data Protection Code. A bottom-up approach was used as the lexicon was the basis to build the ontology. It consisted of a domain ontology reusing top level ontologies. OntoPrivacy has been created to support a tool that allows to query the functional profile of legislative data.

The Neurona Ontologies [12] are application-oriented, and modelled the knowledge for the development of data protection compliance to offer reports regarding the correct application of security measures to data files containing personal data. Their design is based on a Data Protection Knowledge Ontology, which contains the core concepts of the system, and a Data Protection Reasoning Ontology, to assess data protection compliance. These ontologies provide legal professionals and citizens with better access to legal information, but could also support data protection and privacy compliance in organizations and administrations. However, there are several problems that make them unsuitable for the purposes of the current research: they are proprietary and focused on the requirements of the Spanish national data protection law; and their point of view is not focused on the duties of the data controller.

The Privacy by Design (PbD) approach requires that data protection measures are implemented already at the determination of the means of the processing (Article 23 of the GDPR), addressing the design and the implementation of a system. An ontology framework based on the PbD approach [23] consists of nine base ontologies, eight domain ontologies and four application specific on-

tologies. Another interesting approach is presented in [30]. However, that work is not focused on the obligations of the data controller, but rather on expressing the legal norms using an ontology to enforce access control policies.

The idea of extending notations by means of ontologies is not novel [31,28]. It has been acknowledged [18] that ontologies can be integrated in the Software Development Life Cycle (SDLC) in any situation when a domain (the data protection requirements in our case) is frequently used. However, the proposal of this paper addresses the use of the ontology in software design not for the purposes of detailing the application domain of the software, but to specify legal constraints with which the software, or more generally the business process, must comply with. This approach will allow a more consistent interaction between the data controller, the auditors, and the DPAs to ease the transition to the GDPR.

3 An ontology for data protection rules

3.1 Ontology Requirements

The ontology requirement specification is a methodology that facilitates the development of the ontology [35]; in particular, it i) enhances the search for available and existing knowledge resources to be reused in the ontology development; and ii) permits content verification of the ontology's requirements. In this section, we follow that approach.

A collaboration between experts in the domains of legal ontologies and data protection reform (in order to interpret their precepts) is an inherent part of the present ontology development; the involvement of legal experts strengthens the correctness and the application of this domain ontology.

Our *ontological commitment* [14] provides a foundational structure in relation to the new data protection reform. In particular it identifies the scope and extent of the obligations of the data controller, especially in relation to the rights of the data subject.

The ontology development methodology is based on a bottom-up approach, manually extracting terms from official legal documents and using knowledge sources for definitions. We supported our knowledge acquisition step in the selection of the relevant knowledge sources. As established by most ontology methodologies nowadays, the development consists of 1) a preparatory phase (the specification of ontology requirements); 2) the development phase (knowledge acquisition, experts, reuse conceptualization, definition of classes, relations, properties, instances, expert validation and formalization); and 3) the forthcoming evaluation phase (internal consistency, requirements, competency questions, and expert evaluation), which will be part of a future work. The domain-specific knowledge was harvested from a documental corpus from official normative sources, partic-

ularly the DPD, the GDPR (permeable to changes in the final text[4]), and the Handbook on European data protection law [16].

Pursuing the *context of use* (users and use), this work anticipates the impact that the GDPR is likely to have on firms once it enters into force. While businesses have a legitimate interest in collecting personal data as information assets to achieve their business goals, they should also comply with regulatory requirements. The chosen context of use improves the understanding of this domain, and enhances integration/interoperation within a business process.

Competency Questions (CQs) for ontology requirement description, development and further evaluation can help expound scope and domain in ontology design methodologies [17,36]. For our data protection ontology, the following are CQs: 1. What are the obligations of a data controller? 2. What are the functions of a data processor? 3. What are the rights of the data subject? 4. How do the rights of the data subject relate to the obligations of the data controller and the functions of the processor? 5. How can a data subject interact and/or enforce their rights against a data controller? 6. What are the possible fines and sanctions issued in response to violations by data controllers? 7. Who supervises a data controller?

Concerning *non-functional requirements*, this ontology is expressed in Web Ontology Language (OWL) [5] and uses Protégé [27] as the ontology development environment. A graphical depiction of the ontology is shown in Figure 1. The framework presented in this paper relies on previous efforts of the community in the field of legal knowledge representation, therefore we reuse concepts from LKIF Core and SKOS.

The *structure* [9] of the presented ontology is the result of the combination of two different approaches: the ontology design is derived from [16], with only a few slight modifications to address the changes introduced in the GDPR. The conceptualization focuses on constructing formal specifications highlighting the obligations of the data controller and (when possible) matching them with the corresponding rights of the data subject. In this sense, the ontology is structured in a way similar to the Hohfeldian model [20] (as described in subsection 3.2).

3.2 Describing data protection concepts

The perspectives on the sources of legal knowledge in terms of users of legal knowledge (the end-users we described above), provide a specification of what *terms* and *definitions* are used [8] (rather than a broad legal context of data protection). Legal knowledge analysis and acquisition is based on [16], which constitutes our conceptual framework (together with the GDPR) and provides a high-level partitioning of the European data protection. The ontology's architecture follows the same structure, and therefore it is made up of the following blocks:

[4] We used the official Commission text, COM(2012) 11 final. To better sharpen the scope, the ontology does not refer to decisions of courts or DPAs. The purpose is not to define a model of the legal text, but to model the requirements that the controller must meet to be compliant with the legislation.

1. the basic data protection principles;
2. the rules of data processing (constituting most of the duties of the data controller);
3. the data subject's rights.

An ontology entails a given level of consensus in a particular community. Within the data protection domain this includes basic data protection principles, as they have been established over the years by the Council of Europe (CoE), the EU, and the national DPAs. These serve as the foundation for our ontology. It is from these concepts that we derive and define the obligations of the data controller while contrasting them to the rights of the data subject. The result of the principles analysis is a set of ontology classes, their attributes and the relations between them.

The following is an enumeration of some of the principles, as classified under the European Data Protection Handbook [16]: lawfulness principle; purpose limitation principle (personal data must be processed for specified and lawful purposes); data quality principles (data must be adequate, relevant and not in excess in relation to the purpose of the processing, accurate, up to date); principle of data minimization, among others. A more detailed description of the principles underlying the ontology is given in [7].

The data protection principles constitute the unifying harmony underlying controller's obligations (called *Rules* in the ontology, to ensure consistency with the knowledge sources) and data subject's rights. Since they are reifications of the general principles, in the ontology every data processing rule or data subject's right is a subclass of some principles. For example, the LawfulnessPrinciple entails a LawfulnessRule, which *is a* processing rule, and can consist of the data subject's Consent, a LegalObligation of the controller, a VitalInterest, a Contract, and so on.

To relate the data subject's rights with the corresponding rules of the controller, we define the deontic concepts in terms of correlative relations between right and rule (obligation), assuming symmetric roles. As an example of such an approach, the data subject's right to access corresponds to the obligation of the controller to provide means to request access to the data. To exercise the right, the data subject must perform a single access, which is defined as a subclass of the right to access, and is bound by a relationship with the data for which access is requested; similarly, the data subject can exercise the right to object to the processing of personal data. The objection (a subclass of the right to object) is related to a specific processing by a functional property called isObjected, defined in the domain of Processing. This property is also used to define the lawfulness of the processing, because personal data cannot be lawfully processed if the data subject has exercised the right to object.

Table 1 shows the hierarchy of the main concepts of the ontology.

Relations bind two resources (normally classes), and for each relation a *domain* and a *range* can be defined. A domain is the set of possible classes where the relation can be applied, and a range is the set of possible values of a relation. Table 2 shows the main relations in the ontology.

Root Classes	Subclasses
Data Processing	Processing activity, Processing Mode, Lawful processing
Data Subject Right	Right to rectification, Right to object, Right to no profiling, right to portability, right to erasure
Processing Rule	Compliance, Impact Assessment, Transparent information, Security, Lawfulness Rule

Table 1. Top-level hierarchy.

Relation	Domain	Range
hasObligation	Controller	Legal Obligation
notifyBreach	Controller	Data Breach
consentGrantedBy	Consent	Data Subject
AccessData	Right of Access	Personal Data

Table 2. Main relations.

To formalize the ontology, a useful subset of classes were reused from LKIF Core in order to offer a solid support for the acquisition, sharing and reuse of legal knowledge. LKIF Core [19] is an established legal ontology. Our most generic concepts were linked with LKIF-Core concepts (such as the right, rule, legal person and natural person) using the SKOS data model[5]. Our alignment is compliant to it, but axiomatizes domain concepts of data protection, which is our priority and ontological commitment. There was therefore no need to extend the core ontology.

4 Extending business process notation

The ontology described in section 3 can be used to aid a data controller in being compliant with the GDPR. When developing a software system, the PbD approach mentioned in section 2 means that the SDLC should address data protection. The SDLC is a workflow which can be expressed by means of formal notations such as Unified Modeling Language (UML) [22]. However, UML is domain-neutral. To express data protection, the data protection ontology would be useful: by exploiting UML's extensibility features [3], such as profiles, the expressiveness of (for example) activity or sequence diagrams can be enhanced, to specify data protection activities or requirements that a certain software routine, component, class should address.

But this is not sufficient for GDPR compliance. Many of the obligations of the GDPR involve organizational requirements (such as a risk assessment), and also manual processing. Some of these activities have nothing to share with the SDLC, but are still subject to the GDPR.

In this perspective, business processes [13] are more suited to embrace all the activities that can be subject to the GDPR, regardless of whether they

[5] `http://www.w3.org/2009/08/skos-reference/skos.html`.

are performed manually or software-based, or have a technical or organizational nature. Business processes are used to provide a description of the relationships between the various activities performed within a business, at various degrees of detail [34].

Various notations exist for specifying business processes, the most popular of which are Web Services Business Process Execution Language (WS-BPEL) [4] and Business Process Model and Notation (BPMN) [2], which have some similarities but still differ in scopes and domains [32]. They are based on an eXtensible Markup Language (XML) grammar and have extensibility features, including the possibility of using tags from different XML languages (such as the OWL/XML serialization of the ontology). We chose to implement an extension of BPMN to demonstrate the possibilities offered by the present work, but the methodology is a general one that can be applied to any extensible notation.

Introducing data protection requirements by means of an ontology is a methodology that can be used in conjunction with different technologies, and it also provides a means to make heterogeneous models interoperable. In other words, the description of a workflow process might use different models at different levels (e.g., UML and BPMN): if both are extended using the same ontology, the data protection requirements would be consistent, thus easing the integration and auditing of the overall workflow.

4.1 BPMN implementation

The proposed approach has been integrated, although at a very basic level, in a BPMN 2.0 modeling tool. BPMN does not have a uniform implementation. Although it is defined as a standard, it is designed so that its implementation is platform-specific. For the purposes of the present paper, we have selected the Eclipse BPMN2 Modeler[6]. It is an Eclipse plugin which implements BPMN features using Model-Driven Engineering (MDE) techniques and the Ecore metamodel. The Eclipse version used is 4.5 (Mars).

The BPMN standard defines several different diagrams (Process, Collaboration, Choreography and Conversation) which serve different purposes. For this example, we only focused on the Process diagram, although the same methodology can be extended to all diagram types. We created an Eclipse extension plugin for BPMN, defining a new type of Task called Data Protection Task. The new task type has a distinctive graphic appearance (marked with a red icon) and supports annotations extracted from the ontology. The properties of the new Data Protection Task include a new tab which allows to introduce the annotations for data protection.

The implementation of the form to add the annotations parses through the data protection ontology using the OWL Application Programming Interface (API)[7]. Since our purpose is to offer a way to specify the activities that a data controller must perform for GDPR compliance, the reasoner selects the OWL

[6] https://www.eclipse.org/bpmn2-modeler/

[7] `http://owlapi.sourceforge.net/`

classes that are descendants of the Rule class. This is a rough implementation, but it can be refined at the desired level, using the ontology structure or its instances, adding extra parameters and so on.

Figure 2 shows the interface of the extension plugin in operation and a sample application of the extended notation[8]. The example, which is built upon the official BPMN example from [1, p. 170], is not a real business process, but only aims at showing the possibilities of our approach. Some TASKs have been replaced with DATA PROTECTION TASKs. So, for example, the Handle Order activity has been annotated with the following three ontology classes:

Consent because the data subject must consent to the processing;
Security to ensure the protection of security measures;
AppropriateSafeguards because the customer's data might have to be transmitted to a vendor which might be located in a non-EU country.

5 Conclusions

In this work, the authors have presented a methodology to address the GDPR requirements within a business process by means of an ontology for personal data protection, to assist data controllers in achieving compliance with the upcoming data protection reform. The modeling was carried out manually by a legal expert. The provisions that contain duties for the data controller and rights for the data subject, have been selectively identified and built into the ontology.

The granularity of the ontology is still quite coarse. High detail would be required in a judicial perspective, but not in the scope of the current research. However, some of the concepts expressed in the ontology are vague because they are expressed as such in the law, and not fit for direct usage. These concepts must be coordinated with knowledge from other domains. Computer security standards can partly fill these gaps, so understanding the relationship between them and the GDPR would be key to a fast transition to the new legislation.

This ontology is by definition a work in progress. It will have to be adapted to the changes in the legal text when a final version of the GDPR is released. However, in the final text, the core concepts expressed in the ontology won't drift significantly from the current ones. This structure is the basis for further refinements. It will act as a starting point which was necessary to pursue the long-term goal of verifying compliance with the GDPR.

The approach presented in this work may ease the transition from the DPD to the GDPR and provide a basis for the PbD model. It can provide benefits to all end-users. Data controllers and processors will be able to determine what their duties are, on the basis of the rights of the data subject. Auditors will have a structured knowledge that can dissipate the mists of terminological uncertainties. DPAs can speed up their procedures thanks to a clearer notation.

[8] The sources are available at `https://bitbucket.org/guerret/lu.uni.eclipse.bpmn2`. The "resources" folder contains the OWL file with the ontology.

The formalization of the meaning of legal terms in an ontology could help compare the impact of the new legislation on the existing national regimes, as well as overcome linguistic differences in data protection across the EU. Also, expressing the controller's requirements through an ontology will allow to more easily adapt the design to changes in the law and its interpretation, in a dynamic perspective.

The ontology may encompass automated classification to facilitate finding documents. Querying performance is foreseen as a future development in our ontology, using SPARQL-DL to ascertain the corresponding rights and duties. For example, a database structured according to the ontology could be queried by data subjects to retrieve the rights and remedies in case of breaches and violations; by data controllers, to understand their obligations; by data processors, to clarify their functions.

The long-term aims of the current research focus on assessing compliance to the GDPR by means of security standards. This purpose will require development a similarly-structured ontology for security standards and a methodology to compare the degree of overlapping between the two normative bodies.

From a technical perspective, there are many improvements that can be investigated as well. The sample plugin introduced in section 4 could benefit from a more formal implementation using MDE, for example by defining the meta-model of the extension, integrating it with the meta-models of BPMN and OWL, and using it to generate the supporting classes.

Regardless of the underlying technologies used, the integration of the SDLC or business process notation with the data protection annotations from the ontology could also be enhanced with metrics to analyze the degree of coverage of the GDPR. Finally, the proposed extension still lacks an adequate validation methodology.

6 Acknowledgments

This paper is supported by the Joint International Doctoral Degree in Law, Science and Technology (Last-JD) coordinated by the University of Bologna, and by the Luxembourg Privacy Cluster (LPC) at the University of Luxembourg.

References

1. BPMN 2.0 by example. Tech. Rep. dtc/2010-06-02, Object Management Group (June 2010)
2. Business process model and notation (BPMN). Tech. Rep. formal/2011-01-03, Object Management Group (January 2011)
3. Alhir, S.S.: Guide to Applying the UML. Springer Professional Computing, Springer New York (2002)
4. Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C.K., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A.: Web services business process execution language version 2.0. Tech. rep., OASIS (April 2007), `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html`

5. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, chap. 4, pp. 67–92. International Handbooks on Information Systems, Springer Berlin Heidelberg, second edn. (2004)
6. Bartolini, C., Gheorghe, G., Giurgiu, A., Sabetzadeh, M., Sannier, N.: Assessing IT security standards against the upcoming GDPR for cloud systems. In: Proceedings of the Grande Region Security and Reliability Day (GRSRD) 2015. pp. 40–42 (March 2015)
7. Bartolini, C., Muthuri, R.: Reconciling data protection rights and obligations: An ontology of the forthcoming eu regulation. In: Proceedings of the Workshop on Language and Semantic Technology for Legal Domain (LST4LD), Recent Advances in Natural Language Processing (RANLP) (September 2015), to be published.
8. Breuker, J., Boer, A., Hoekstra, R., van den Berg, K.: Developing content for LKIF: Ontologies and frameworks for legal reasoning. In: Proceedings of the International Conference on Legal Knowledge and Information Systems (JURIX). pp. 169–174 (December 2006)
9. Breuker, J., Valente, A., Winkels, R.: Use and reuse of legal ontologies in knowledge engineering and information management. In: Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A. (eds.) Law and the Semantic Web, Lecture Notes in Computer Science, vol. 3369, chap. 2, pp. 36–64. Springer Berlin Heidelberg (2005)
10. Cappelli, A., Lenzi, V.B., Sprugnoli, R., Biagioli, C.: Modelization of domain concepts extracted from the Italian privacy legislation. In: Proceedings of the 7^{th} International Workshop on Computational Semantics (IWCS-7) (January 2007)
11. Casellas, N.: Legal Ontology Engineering Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge, Law, Governance and Technology Series, vol. 3. Springer Netherlands (2011)
12. Casellas, N., Nieto, J.E., Roig, A.M.A., Torralba, S., Reyes, M., Casanovas, P.: Ontological semantics for data privacy compliance: The NEURONA project. In: Proceedings of the Intelligent Privacy Management Symposium. pp. 34–38 (March 2010)
13. Davenport, T.H., Short, J.E.: The new industrial engineering: Information technology and business process redesign. Sloan Management Review 31(4), 11–27 (Summer 1990)
14. Davis, R., Shrobe, H., Szolovits, P.: What is a knowledge representation? AI Magazine 14(1), 17–33 (Spring 1993)
15. European Commission: A digital single market strategy for Europe. `http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52015DC0192&from=EN` (May 2015)
16. European Union Agency for Fundamental Rights: Handbook on European data protection law (April 2014)
17. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. Proceedings of the 1995 International Joint Conference on AI, Workshop on Basic Ontological Issues in Knowledge Sharing (August 1995)
18. Hesse, W.: Ontologies in the software engineering process. In: Lenz, R., Hasenkamp, U., Hasselbring, W., Reichert, M. (eds.) Proceedings of the 2^{nd} GI-Workshop on Enterprise Application Integration (EAI). pp. 3–15 (June 2005)
19. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A.: LKIF Core: Principled ontology development for the legal domain. In: Breuker, J., Casanovas, P., Klein, M.C., Francesconi, E. (eds.) Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood, Frontiers in Artificial Intelligence and Applications, vol. 188, pp. 21–52. IOS Press (January 2009)

20. Hohfeld, W.N.: Fundamental legal conceptions as applied in judicial reasoning. The Yale Law Journal 26(8), 710–770 (June 1917)
21. International Organization for Standardization: ISO/IEC 27001 – Information technology – Security techniques – Information security management systems – Requirements, second edn. (October 2013)
22. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley, Reading, Massachussetts (1999)
23. Kost, M., Freytag, J.C., Kargl, F., Kung, A.: Privacy verification using ontologies. In: Proceedings of the Sixth International Conference on Availability, Reliability and Security (ARES). pp. 627–632 (August 2011)
24. Massacci, F., Prest, M., Zannone, N.: Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation. Tech. rep., University of Trento (November 2003)
25. Mikkonen, T.: Perceptions of controllers on EU data protection reform: A Finnish perspective. Computer Law & Security Review 30(2), 190–195 (April 2014)
26. Mitre, H.A., González-Tablas, A.I., Ramos, B., Ribagorda, A.: A legal ontology to support privacy preservation in location-based services. In: Meersman, R., Tari, Z., Herrero, P. (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Lecture Notes in Computer Science, vol. 4278, pp. 1755–1764. Springer Berlin Heidelberg (2006)
27. Noy, N.F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R.W., Musen, M.A.: Creating semantic web contents with Protégé-2000. IEEE Intelligent Systems 16(2), 60–71 (March–April 2001)
28. Paulheim, H., Probst, F.: Ontology-enhanced user interfaces: A survey. International Journal on Semantic Web & Information Systems 6(2), 36–59 (April 2010)
29. Pfleeger, C.P., Pfleeger, S.L.: Security in Computing. Prentice Hall, Upper Saddle River, NJ, USA, fourth edn. (October 2006)
30. Rahmouni, H.B., Solomonides, T., Casassa Mont, M., Shiu, S.: Privacy compliance and enforcement on European healthgrids: an approach through ontology. Philosophical Transactions of the Royal Society A 368(1926), 4057–4072 (September 2010)
31. Rebstock, M., Fengel, J., Paulheim, H.: Ontologies-Based Business Integration. Business Information Systems, Springer-Verlag Berlin Heidelberg (2008)
32. Recker, J.C., Mendling, J.: On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In: Latour, T., Petit, M. (eds.) The 18^{th} International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium. pp. 521–532. Namur University Press (June 2006)
33. Reding, V.: The upcoming data protection reform for the European Union. International Data Privacy Law (November 2010)
34. Reijers, H.A.: Design and Control of Workflow Processes: Business Process Management for the Service Industry, Lecture Notes in Computer Science, vol. 2617. Springer-Verlag Berlin Heidelberg (2003)
35. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to write and use the ontology requirements specification document. In: Meersman, R., Dillon, T., Herrero, P. (eds.) On the Move to Meaningful Internet Systems: OTM 2009. Lecture Notes in Computer Science, vol. 5871, pp. 966–982. Springer Berlin Heidelberg (November 2009)
36. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on AI (IJCAI) (August 1995)

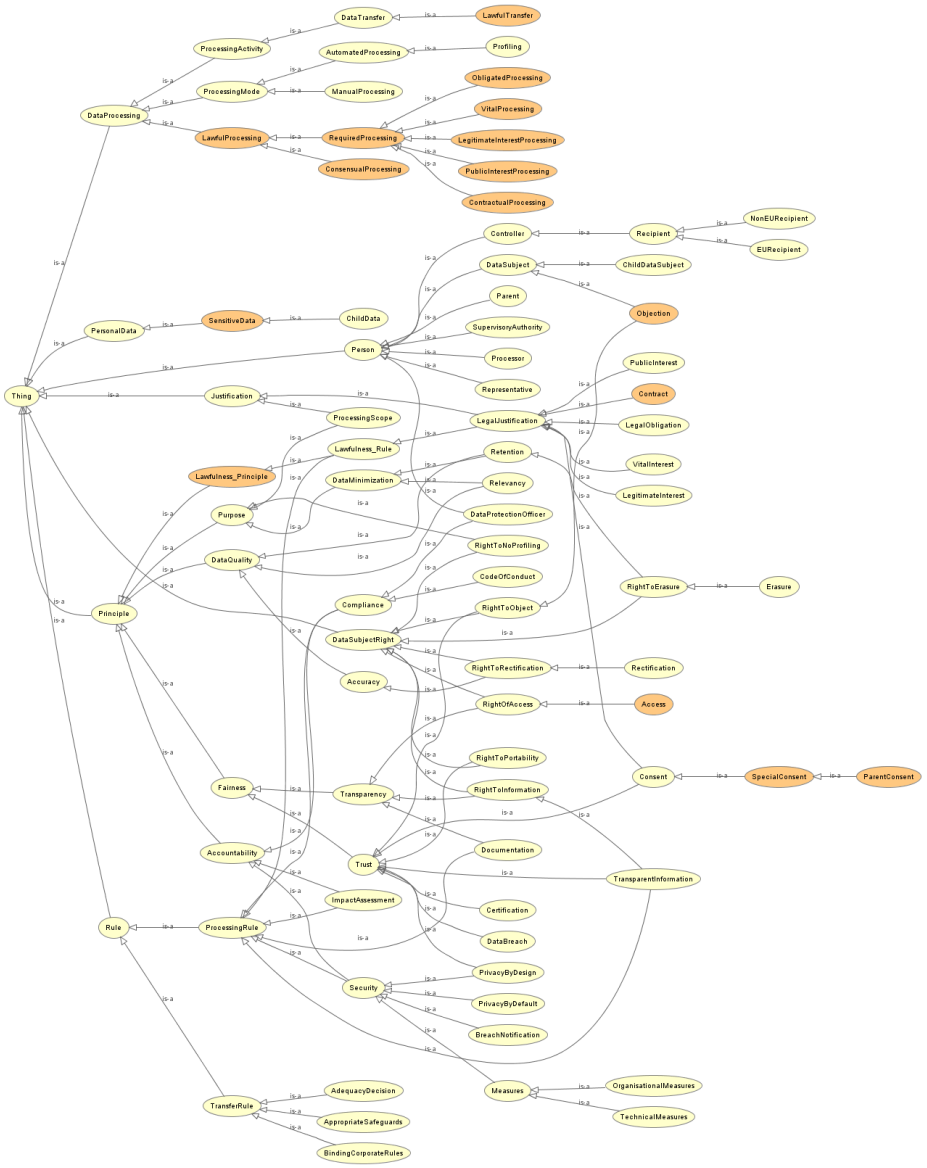
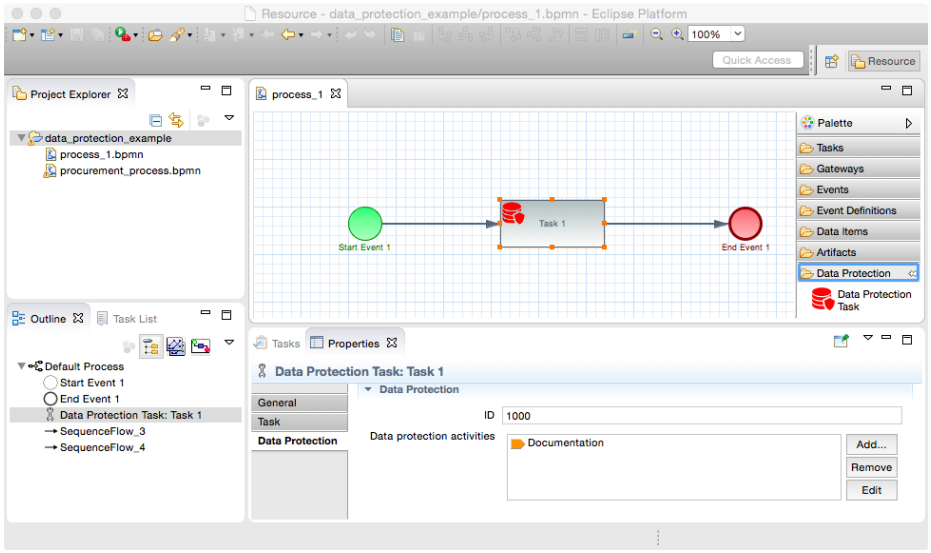
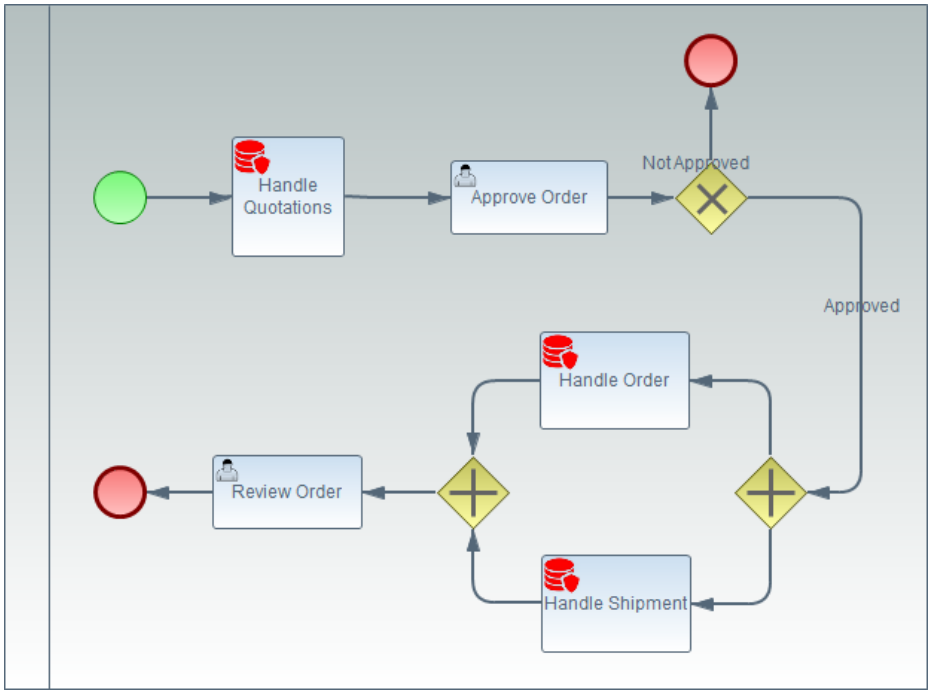


Fig. 1. Schema of the data protection ontology.



(a) The BPMN extension plugin.



(b) Procurement process example.

Fig. 2. The data protection ontology extension plugin.

Abductive Logic Programming for normative reasoning and ontologies

Marco Gavanelli[1], Evelina Lamma[1], Fabrizio Riguzzi[2], Elena Bellodi[1], Riccardo Zese[1], and Giuseppe Cota[1]

[1] Dipartimento di Ingegneria – University of Ferrara
[2] Dipartimento di Matematica e Informatica – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy [name.surname]@unife.it

Abstract. Abductive Logic Programming (ALP) has been exploited to formalize societies of agents, commitments and norms, taking advantage from ALP operational support as a (static or dynamic) verification tool. In [7], the most common deontic operators (obligation, prohibition, permission) are mapped into the abductive expectations of an ALP framework for agent societies. Building upon such correspondence, in [5], authors introduced $Deon^{+}$, a language where obligation and prohibition deontic operators are enriched with quantification over time, by means of ALP and Constraint Logic Programming (CLP).
In recent work [28, 29], we have shown that the same ALP framework can be suitable to represent Datalog$^{\pm}$ ontologies. Ontologies are a fundamental component of both the Semantic Web and knowledge-based systems, even in the legal setting, since they provide a formal and machine manipulable model of a domain.
In this work, we show that ALP is a suitable framework for representing both norms and ontologies. Normative reasoning and ontological query answering are obtained by applying the same abductive proof procedure, smoothly achieving their integration. In particular, we consider the ALP framework named $\mathcal{S}$CIFF and derived from the IFF abductive framework, able to deal with existentially (and universally) quantified variables in rule heads and CLP constraints.
The main advantage is that this integration is achieved within a single language, grounded on abduction in computational logic.

1 Introduction

Norms represent desirable behaviors of members of a human or artificial society.

A normative system is a set of norms, together with mechanisms to reason about, apply, and modify them. Norms can be encoded by exploiting the notions of obligation, permission and prohibition, often modelled as modal operators, in the tradition of Deontic Logic [43].

As for the structure of the formulas representing norms, a widely adopted approach is to encode norms as logical rules in the form of implications, since: (i) implications correspond intuitively to conditional norms, which state that some deontic consequence (such as the obligation for an agent to perform an

action) follows from a state of affairs; (*ii*) rule-based systems also provide an operational support for reasoning, and draw conclusions (regarding, e.g., expected behavior, or norm violations and related sanctions). In the legal domain, the British Nationality Act was formalized using Logic Programming [41]; later, argument-based extended logic programming with defeasible priorities [37] and the use of defeasible logic was proposed [31]. Satoh et al.'s PROLEG is a Prolog implementation of the Presupposed Ultimate Fact Theory of the Japanese Civil Code [40]. The contract cancellation under the Japanese law was also formalized in computational logics [20].

Normative systems have been also advocated as a tool to model and reason upon a single agent, as in [13] where a normative system can be seen as a normative agent, equipped with mental attitudes, about which other agents can reason, choosing either to fulfill their obligations, or to face the possible sanctions.

More often, normative systems regulate interaction in multi-agent systems [12]. Among the organizational models, [21, 23, 22] exploit Deontic Logic to specify the society norms and rules. The whole research project ALFEBIITE [10] was focused on the formalization of an open society of agents using Deontic Logic.

The EU IST Project SOCS proposed a Computational Logic approach to multi-agent systems. The SOCS social model represents and verifies both social interaction protocols among members regulated via abductive expectations [2], and member specifications themselves [14] in Abductive Logic Programming (ALP). Both approaches have been later applied to model and reason about norms with deontic flavours [7, 38].

In the EU project IMPACT [11, 25], agent programs may be used to specify what an agent is obliged to do, what an agent may do, and what an agent cannot do on the basis of deontic operators of Permission, Obligation and Prohibition (whose semantics does not rely on a Deontic Logic semantics).

In the meantime, legal ontologies have proved crucial for representing, processing and retrieving legal information. A collective reflection on the theoretical foundations of legal ontology engineering is [39]. The ESTRELLA project [26] aimed at developing a standard based platform allowing public administrations to deploy comprehensive legal knowledge management solutions. To this purpose, the project developed a Legal Knowledge Interchange Format (LKIF), building upon OWL, and modeled European tax related legislation and national tax legislation of two European countries as case studies. Integration of rules and ontologies has been faced in [9], which proposes a normative language that combines expressivity of Logic Programming and Description Logic (DL) for hybrid knowledge bases modeling human laws, with examples from the Portuguese Penal Code. Another approach is the mapping of DL into computational logic; for example the $\mathcal{ALCN}$ Description Logics was also mapped into Open Logic Programming [42], an extension of ALP. Notable work has been done also in applying abductive reasoning to DL [34, 24].

In Computational Logic, ALP has been proved a powerful tool for knowledge representation and reasoning [33], taking advantage from ALP operational support as (static or dynamic) verification tool. ALP languages are usually equipped

with a declarative (model-theoretic) semantics, and an operational semantics given in terms of a proof-procedure. Several abductive proof procedures have been defined, with many applications (diagnosis, monitoring, verification, etc.). The IFF abductive proof-procedure [27] was proposed to deal with forward rules, and with non-ground abducibles. It has been later extended [4], and the resulting proof procedure, named $\mathcal{S}$CIFF, can deal with both existentially and universally quantified variables in rule heads and Constraint Logic Programming (CLP) constraints [32]. The resulting system was used for modeling and implementing several knowledge representation frameworks, such as deontic logic [7], where the deontic notions of obligation and permission are mapped into special $\mathcal{S}$CIFF abducible predicates, normative systems [5], interaction protocols for multi-agent systems [8], Web services choreographies [1], etc.

In this work, we move a step forward, by showing that ALP, and $\mathcal{S}$CIFF in particular, is a suitable framework for representing and integrating both norms and Datalog$^{\pm}$ ontologies. Normative reasoning and ontological query answering are obtained by applying the $\mathcal{S}$CIFF abductive proof procedure. The main advantage is that this integration is achieved within a single language, grounded on abduction in Computational Logic.

The paper is organized as follows. We first introduce ALP and the $\mathcal{S}$CIFF framework in Section 2, also mentioning its underlying proof procedure. Then, in Section 3, we introduce (a subset of) the deontic language $Deon^{+}$ and show its mapping into $\mathcal{S}$CIFF. Section 4 introduces Datalog$^{\pm}$ to formalize ontologies, and shows its correspondence to $\mathcal{S}$CIFF integrity constraints. This paves the way to the integration of norms and ontologies, discussed in Section 5, where we show a simple example, with norms and an ontology, and discuss the kind of inference supported by the $\mathcal{S}$CIFF proof procedure. More relevant related work is mentioned throughout the various sections. In Section 6 we conclude the paper, and outline future work.

2 ALP and the $\mathcal{S}$CIFF language

Abductive Logic Programming (ALP) is a family of programming languages that integrate abductive reasoning into logic programming. An ALP program consists of a logic program, also called *knowledge base* KB, that is a set of clauses $head \leftarrow body$; beside the defined literals, the body can also contain literals of some distinguished predicates, belonging to a set $\mathcal{A}$ and called *abducibles*. The aim is finding a set of abducibles $\Delta^{\mathcal{A}}$, built from symbols in $\mathcal{A}$, that, together with the KB, is an explanation for a given known effect (called *goal* $\mathcal{G}$) and satisfies a set of logic formulae, called *Integrity Constraints (ICs)*:

$$KB \cup \Delta^{\mathcal{A}} \models \mathcal{G}$$
$$KB \cup \Delta^{\mathcal{A}} \models IC.$$

$\mathcal{S}$CIFF [4] is a language in the ALP class, extension of the IFF proof-procedure [27]. As in the IFF, integrity constraints are in the form $body \rightarrow head$ where *body* is a conjunction of literals and *head* is a disjunction of conjunctions

of literals. While in the IFF the literals can be built only on defined or abducible predicates, in $\mathcal{S}$CIFF they can also be CLP constraints, occurring events (only in the body), or positive and negative expectations, as explained in the following.

Definition 1. *A* $\mathcal{S}$CIFF Program *is a pair* $\langle KB, \mathcal{IC} \rangle$ *where* KB *is a set of clauses and* $\mathcal{IC}$ *is a set of logic implications called* Integrity Constraints.

$\mathcal{S}$CIFF considers a (possibly dynamically growing) set of facts (called *history*) **HAP**, that contains ground atoms $\mathbf{H}(Event[, Time])$. This set can grow dynamically, during the computation, thus implementing a dynamic acquisition of events. Some distinguished abducibles are called *expectations*. A *positive expectation*, written $\mathbf{E}(Event[, Time])$ means that a corresponding event $\mathbf{H}(Event[, Time])$ is expected to happen, while $\mathbf{EN}(Event[, Time])$ is a *negative expectation*, and requires events $\mathbf{H}(Event[, Time])$ not to happen. To simplify the notation, we will omit the $Time$ argument from events and expectations when not needed.

Variables occurring only in positive expectations are existentially quantified (expressing the idea that a single event is enough to support them), while those in negative expectations are universally quantified, so that any event matching with a negative expectation leads to inconsistency with the current hypothesis. CLP [32] constraints can be imposed on variables. The computed answer includes in general three elements: a substitution for the variables in the goal (as usual in Prolog), the constraint store (as in CLP), and the set $\Delta^{\mathcal{A}}$ of abduced literals.

The declarative semantics of $\mathcal{S}$CIFF includes the classic conditions of ALP:

$$KB \cup \mathbf{HAP} \cup \Delta^{\mathcal{A}} \models \mathcal{G} \tag{1}$$

$$KB \cup \mathbf{HAP} \cup \Delta^{\mathcal{A}} \models \mathcal{IC} \tag{2}$$

plus specific conditions to support the confirmation of expectations.

As the history can be dynamically growing, it makes sense to adopt either an open or closed world assumption on the history, depending on the envisaged application. $\mathcal{S}$CIFF can support both types of reasoning. In a *skeptical* reasoning attitude, all hypotheses that are not explicitly confirmed are rejected, assuming that no more events can happen (closed world assumption on the history): all the positive expectations that are not matched with an actual event are disconfirmed (symmetrically, the pending negative expectations are confirmed). In a *credulous* reasoning attitude, a set of hypotheses is acceptable even if some hypotheses are not explicitly confirmed, as long as the set is consistent. Declaratively, in skeptical reasoning positive expectations are confirmed if

$$KB \cup \mathbf{HAP} \cup \Delta^{\mathcal{A}} \models \mathbf{E}(X) \rightarrow \mathbf{H}(X), \tag{3}$$

which is not required in a credulous attitude. In this paper we adopt a credulous reasoning attitude. In both cases, negative expectations should not be disconfirmed by actual events:

$$KB \cup \mathbf{HAP} \cup \Delta^{\mathcal{A}} \models \mathbf{EN}(X) \wedge \mathbf{H}(X) \rightarrow \mathit{false}. \tag{4}$$

and the same event cannot be expected both to happen and not to happen:

$$KB \cup \mathbf{HAP} \cup \Delta^{\mathcal{A}} \models \mathbf{E}(X) \wedge \mathbf{EN}(X) \rightarrow \mathit{false}. \tag{5}$$

Definition 2 ($\mathcal{S}$CIFF answer). *Given a $\mathcal{S}$CIFF program $\langle KB, \mathcal{IC}\rangle$ and a history* **HAP***, a goal $\mathcal{G}$ is a $\mathcal{S}$CIFF answer if there is a set $\Delta^{\mathcal{A}}$ such that (1), (2), (4) and (5) are satisfied. In this case, we write*

$$\langle KB, \mathcal{IC}\rangle \models_{\mathbf{HAP}} \mathcal{G}.$$

The $\mathcal{S}$CIFF proof-procedure is a rewriting system that defines a proof tree, that represents all abductive solutions and whose nodes represent states of the computation. A set of transitions rewrite a node into one or more children nodes. $\mathcal{S}$CIFF inherits the transitions of the IFF proof-procedure [27], and extends it in various directions. We recall the basics of $\mathcal{S}$CIFF; a complete description is in [4], with proofs of soundness, completeness, and termination. An efficient implementation of $\mathcal{S}$CIFF is described in [6].

Each node of the proof is a tuple $T \equiv \langle R, CS, PSIC, \Delta^{\mathcal{A}}\rangle$, where R is the resolvent, CS is the CLP constraint store, PSIC is a set of implications (called *Partially Solved Integrity Constraints*) derived from propagation of integrity constraints, and $\Delta^{\mathcal{A}}$ is the current set of abduced literals. The main transitions, inherited from the IFF are:

Unfolding replaces a (non abducible) atom with its definitions;
Propagation if an abduced atom $\mathbf{a}(X)$ occurs in the condition of an IC (e.g., $\mathbf{a}(Y) \rightarrow p$), the atom is removed from the condition (generating $X = Y \rightarrow p$);
Case Analysis given an implication containing an equality in the condition (e.g., $X = Y \rightarrow p$), generates two children in logical or (in the example, either $X = Y$ and p, or $X \neq Y$);
Equality rewriting rewrites equalities as in the Clark's equality theory;
Logical simplifications other simplifications like (true $\rightarrow A$) $\Leftrightarrow A$, etc.

$\mathcal{S}$CIFF includes also the transitions of CLP [32] for constraint solving.

In this paper we consider the *generative version* of $\mathcal{S}$CIFF, called g-$\mathcal{S}$CIFF [3, 35], in which also the **H** events in IC's heads are considered as abducibles. A history **HAP** is provided as input, and further **H** atoms can be assumed like the other abducible predicates; they are then collected in a set $\mathbf{HAP}' \supseteq \mathbf{HAP}$.

Definition 3 (g-$\mathcal{S}$CIFF answer). *Given a $\mathcal{S}$CIFF program $\langle KB, \mathcal{IC}\rangle$ and a history* **HAP***, we say that a goal $\mathcal{G}$ is a g-$\mathcal{S}$CIFF answer if there exist a set $\Delta^{\mathcal{A}}$ and a set $\mathbf{HAP}' \supseteq \mathbf{HAP}$ such that (1), (2), (4) and (5) are satisfied.*[3] *In this case, we write*

$$\langle KB, \mathcal{IC}\rangle \models^{g}_{\mathbf{HAP} \rightsquigarrow \mathbf{HAP}'} \mathcal{G} \qquad \text{or simply} \qquad \langle KB, \mathcal{IC}\rangle \models^{g}_{\mathbf{HAP}} \mathcal{G}.$$

[3] In (1), (2), (4) and (5), one should read $\mathbf{HAP}'$ instead of **HAP**.

3 Norms in $\mathcal{S}$CIFF

In [5], $\mathcal{S}$CIFF was exploited to support legal regulations expressed in a deontic language, named $Deon^+$. In $Deon^+$, (positive) actions are represented by terms and, as usual in logic programming, terms can contain variables, constants, terms. Building upon this action language, obligations are represented as $\mathcal{S}$CIFF atoms $\mathbf{E}(A, T)$, where A is any action description, and T is a CLP variable, existentially quantified. For instance, the sentence "It is mandatory that John answers me", corresponds to:

$$\exists T \qquad \mathbf{E}(answer(john, me), T)$$

as any reply in any time complies to the obligation, and the obligation will no longer hold after John sends his answer.

In a similar way, prohibitions are represented as atoms $\mathbf{EN}(A, T)$, where again A is any action description, and T is a CLP variable, universally quantified (unless it occurs also in a $\mathbf{H}$ or $\mathbf{E}$ atom, in such a case it is existentially quantified). For instance, the sentence "It is forbidden that John smokes", corresponds to:

$$\forall T \qquad \mathbf{EN}(smoke(john), T)$$

because in any time John is not allowed to smoke, and the fact he did not smoke one minute ago does not allow him to smoke now and later on.

A notable advantage of adopting ALP is that the action language is not limited to the propositional case, as in the examples above, but it can contain variables in its turn, quantified (existentially or universally) as the T variable.

The adoption of CLP variables for representing time adds expressiveness to deontic operators and easily recovers deadlines by constraints over time variables. Constraints imposed on universally quantified variables are considered as quantifier restrictions [16]; a sentence like "It is forbidden that John leaves the meeting before 10" is therefore represented in $Deon^+$ as:

$$\forall T : T < 10 \qquad \mathbf{EN}(leave(john, meeting), T).$$

Integrity constraints can be also exploited to represent conditional obligatoriness and the deontic logic of deadlines, as shown in [7]. For instance, integrity constraints of the kind $\mathbf{H}(B) \rightarrow \mathbf{E}(A)$ are suitable to represent the obligatoriness of A given B, and the dyadic deontic operator $\mathbf{Obl}(A|B)$ in particular.

Deontic logic with deadlines and the operator $O(\rho \leq \delta)$ proposed by [15], meaning that the action ρ ought to be brought about before (or at the same time) another action δ happens, can be mapped into the integrity constraint:

$$\mathbf{H}(\delta, T_\delta) \rightarrow \mathbf{E}(\rho, T_\rho), T_\rho \leq T_\delta.$$

The integrity constraint above therefore represents the obligatoriness of action ρ at time $T_\rho \leq T_\delta$, if δ happens at time T_δ.

4 Datalog$^{\pm}$ Ontologies in $\mathcal{S}$CIFF

W3C has supported the development of a family of knowledge representation formalisms of increasing complexity for defining ontologies, called Web Ontology Language (OWL). DLs were chosen as the logic-based counterpart for the OWL family of languages.

In the Computational Logic realm, more recently, [19, 18] proposed Datalog$^{\pm}$, an extension of Datalog with existential rules for representing lightweight ontologies, encompassing the DL-Lite family, and achieving decidability and tractability [17] under appropriate syntactic conditions.

In short, Datalog$^{\pm}$ extends Datalog by allowing existential quantifiers, the equality predicate and the constant *false* in rule heads. Any Datalog$^{\pm}$ theory may, in fact, include three types of implication rules: Tuple-Generating Dependencies (TGDs), Negative Constraints (NCs) and Equality Generating Dependencies (EGDs), as shown in the following. In standard Datalog, we can represent a rule stating that the father X of any person Y is also a person:

$$\forall X \, \forall Y \; fatherOf(X, Y) \; \wedge \; person(Y) \; \rightarrow \; person(X)$$

In Datalog$^{\pm}$, we get higher expressiveness, and by a TGD rule we can state that any person X has a father Y who is also a person:

$$\forall X \, person(X) \; \rightarrow \; \exists Y \; fatherOf(Y, X) \; \wedge \; person(Y)$$

or that for any person X, and any couple of his/her fathers Y and Z, then Y and Z must be the same, by the following EGD:

$$\forall X \, \forall Y \, \forall Z \; fatherOf(Y, X) \; \wedge \; fatherOf(Z, X) \; \rightarrow \; Y = Z$$

Finally, we can also state, by the following NC, that for any X, his/her father and mother cannot be the same Y:

$$\forall X \, \forall Y \; fatherOf(Y, X) \; \wedge \; motherOf(Y, X) \; \rightarrow \; false$$

Declaratively, given a finite set of relation names $\mathcal{R}$, a Datalog$^{\pm}$ theory T on $\mathcal{R}$, and a (ground) database D for $\mathcal{R}$, the set of models of D given T, denoted $mods(D, T)$, is the set of all (possibly infinite) databases B such that $D \subseteq B$ and every $F \in T$ is satisfied in B.

In Datalog$^{\pm}$ the set of answers to a Conjunctive Query (CQ) q on D given T, denoted $ans(q, D, T)$, is the set of all tuples $\mathbf{t}$ such that $\mathbf{t} \in q(B)$ for all $B \in mods(D, T)$. With abuse of notation, we will write $q(\mathbf{t})$ to mean answer $\mathbf{t}$ for q on D given T.

Operationally, Datalog$^{\pm}$ query answering for CQs and Boolean Conjunctive Queries (BCQs) is achieved via the *chase*, a bottom-up algorithm for deriving atoms entailed by a given database D and a Datalog$^{\pm}$ theory. Informally, the chase works on the database D and extends it through the so-called TGD and EGD chase rules. When the body of a TGD is true in the database D, the TGD

chase rule adds to D new atomic formulas corresponding to TGD's heads with new (null) variables for the existential ones, that are not already in D. Without loss of generality, we assume that every TGD has a single atom in its head (by introducing new predicate symbols, we can always transform TGD-rules with multiple atoms in the head in sets of rules with only one). The EGD chase rule, when the body of an EGD is true in the database D, tries to unify the two terms implied in the equality in the EGD's head. A *hard violation* is raised if the unification fails. A more formal description can be found in [18, 17].

In [28, 29] we have shown that, by suitably extending the $\mathcal{S}$CIFF proof-procedure, we are able to represent in $\mathcal{S}$CIFF a Datalog$^{\pm}$ program, and to use it for ontological reasoning. $\mathcal{S}$CIFF abductive declarative semantics provides the model-theoretic counterpart to Datalog$^{\pm}$ semantics. Operationally, query answering is achieved bottom-up via the *chase* in Datalog$^{\pm}$, while in the ALP framework it is supported by the $\mathcal{S}$CIFF proof procedure, which applies ICs in a forward manner.

In Datalog$^{\pm}$, tuples can be added to the database through the TGD chase rule, and unifications can be done via the EGD chase rule. We mimic the chase through the propagation of integrity constraints in $\mathcal{S}$CIFF. The $\mathcal{S}$CIFF syntax for integrity constraints is extended to allow for **H** literals in the head of integrity constraints. **H** atoms are now considered as abducible atoms, so that, through the propagation of integrity constraints, they are assumed as true if they occur in the head of a (transformed) TGD. Coherently with both the Datalog$^{\pm}$ and $\mathcal{S}$CIFF syntax, variables that occur only in the head of an IC and that occur in a **H** literal are implicitly existentially quantified.

The finite set of relation names of a Datalog$^{\pm}$ relational schema $\mathcal{R}$ is mapped into the set of terms occurring in the **H** predicates of the corresponding $\mathcal{S}$CIFF program. A Datalog$^{\pm}$ database D for $\mathcal{R}$ corresponds to the (possibly infinite) $\mathcal{S}$CIFF history **HAP**, since there is a one-to-one correspondence between each tuple in D and each (ground) fact in **HAP**.

A Datalog$^{\pm}$ theory T is mapped into a $\mathcal{S}$CIFF program with an empty KB, and $IC = \tau(T)$, where ICs have (conjunctions of) **H** atoms and CLP constraints (equalities in particular), or $false$ in their heads, and are obtained by the τ mapping from the original TGDs, EGDs, and NCs.

Terms and atomic formulas have the usual Logic Programming meaning.

The τ mapping is recursively defined as follows, where A is an atom, and F_1, F_2, ..., are Datalog$^{\pm}$ formulae:

$$\begin{aligned}
\tau(Body \rightarrow Head) &= \tau(Body) \rightarrow \tau(Head) \\
\tau(A) &= \mathbf{H}(A) \\
\tau(F_1 \wedge F_2) &= \tau(F_1) \wedge \tau(F_2) \\
\tau(false) &= false \\
\tau(Y_i = Y_j) &= Y_i = Y_j \\
\tau(\exists \mathbf{X}\ A) &= \mathbf{H}(A)
\end{aligned}$$

where the last equation comes from the fact that the quantification for variables that occur only in **H** literals in the head of an IC is always existential, and it is implicit in the $\mathcal{S}$CIFF syntax.

A Datalog TGD F of the kind *body* $\rightarrow$ *head* is mapped into the $\mathcal{S}$CIFF integrity constraint $IC = \tau(F)$, where the *body* is mapped into conjunctions of $\mathcal{S}$CIFF atoms, and *head* into conjunctions of $\mathcal{S}$CIFF abducible $\mathbf{H}$ atoms. Existential quantifications of variables occurring in the *head* of the TGD are maintained in the head of the $\mathcal{S}$CIFF IC, but they are left implicit in the $\mathcal{S}$CIFF syntax, while the rest of the variables are universally quantified with scope the entire IC.

Finally, Datalog$^{\pm}$ NCs are mapped into $\mathcal{S}$CIFF ICs with head *false*, and EGDs into $\mathcal{S}$CIFF ICs, each one with an equality CLP constraint in its head.

Other possible mappings could be considered; interesting research directions could be to map the database D to the KB, and/or atoms to normal abducible predicates, instead of $\mathbf{H}$ events.

Given a Datalog$^{\pm}$ theory T, let us denote the mapping of T into the corresponding set $\mathcal{IC}$ of $\mathcal{S}$CIFF integrity constraints, as $\mathcal{IC} = \tau(T)$.

Recall that, given a Datalog$^{\pm}$ theory T on $\mathcal{R}$, and a database D for $\mathcal{R}$, the set of models of D given T, denoted $mods(D, T)$, is the set of all (possibly infinite) databases B such that $D \subseteq B$ and every $F \in T$ is satisfied in B. For any such database B, in [28, 29], we have proved that there exists an abductive explanation $\mathbf{HAP'} = \tau(B)$ such that:

$$\mathbf{HAP'} \models \mathcal{IC}$$

where $\mathbf{HAP'} \supseteq \mathbf{HAP} = \tau(D)$, and $\mathcal{IC} = \tau(T)$.

Therefore, informally speaking, the set of models of D given T, $mods(D, T)$, corresponds to the set of all the abductive explanations $\mathbf{HAP'}$ satisfying the set of $\mathcal{S}$CIFF integrity constraints $\mathcal{IC} = \tau(\mathcal{T})$.

In [28, 29], we have stated and proved theorems for (model-theoretic) completeness of query answering. Informally, we recall here the completeness result for CQ-answering: for each answer $q(\mathbf{t})$ of a CQ $q(\mathbf{X}) = \exists \mathbf{Y} \Phi(\mathbf{X}, \mathbf{Y})$ on D given T, in the corresponding $\mathcal{S}$CIFF program $\langle \emptyset, \mathcal{A}, \tau(T) \rangle$ there exists an answer substitution θ and an abductive explanation $\mathbf{HAP'}$ for goal $G = \tau(\Phi(\mathbf{X}, \_))$ such that:

$$\langle \emptyset, \mathcal{IC} \rangle \models^{g}_{\mathbf{HAP}} G\theta$$

where $\mathbf{HAP} = \tau(D)$, $\mathcal{IC} = \tau(T)$, and $G\theta = \tau(\Phi(\mathbf{t}, \_))$.

The $\mathcal{S}$CIFF proof procedure was proved sound and complete w.r.t. $\mathcal{S}$CIFF declarative semantics in [4], thus for each abductive explanation δ for a given goal G in a $\mathcal{S}$CIFF program, there exists a $\mathcal{S}$CIFF-based computation producing a set of abducibles (positive expectations to our purposes) $\delta' \subseteq \delta$, and a computed answer substitution for goal G possibly more general than θ.

Example 1 (A real estate information extraction system in ALP). In [30], the authors present a simple ontology for a real estate information extraction system[4]:

$F_1 = ann(X, label), ann(X, price), visible(X) \rightarrow priceElem(X)$

[4] The universal quantifiers are usually left implicit.

If X is annotated as a label, as a price and is visible, then it is a price element.

$F_2 = ann(X, label), ann(X, priceRange), visible(X) \rightarrow priceElem(X)$

If X is annotated as a label, as a price range, and is visible, then it is a price element.

$F_3 = priceElem(E), group(E, X) \rightarrow forSale(X)$

If E is a price element and is grouped with X, then X is for sale.

$F_4 = forSale(X) \rightarrow \exists P\ price(X, P)$

If X is for sale, then there exists a price for X.

$F_5 = hasCode(X, C), codeLoc(C, L) \rightarrow loc(X, L)$

If X has postal code C, and C's location is L, then X's location is L.

$F_6 = hasCode(X, C) \rightarrow \exists L\ codeLoc(C, L), loc(X, L)$

If X has postal code C, then there exists L s.t. C has location L and so does X.

$F_7 = loc(X, L1), loc(X, L2) \rightarrow L1 = L2$

If X has the locations $L1$ and $L2$, then $L1$ and $L2$ are the same.

$F_8 = loc(X, L) \rightarrow advertised(X)$

If X has a location L then X is advertised.

The TGDs F_1-F_8 from the Datalog$^{\pm}$ ontology above are one-to-one mapped into the following $\mathcal{S}$CIFF ICs:[5]

$IC_1 : \mathbf{H}(ann(X, label)), \mathbf{H}(ann(X, price)), \mathbf{H}(visible(X)) \rightarrow \mathbf{H}(priceElem(X))$
$IC_2 : \mathbf{H}(ann(X, label)), \mathbf{H}(ann(X, priceRange)), \mathbf{H}(visible(X)) \rightarrow \mathbf{H}(priceElem(X))$
$IC_3 : \mathbf{H}(priceElem(E)), \mathbf{H}(group(E, X)) \rightarrow \mathbf{H}(forSale(X))$
$IC_4 : \mathbf{H}(forSale(X)) \rightarrow (\exists P)\ \mathbf{H}(price(X, P))$
$IC_5 : \mathbf{H}(hasCode(X, C)), \mathbf{H}(codeLoc(C, L)) \rightarrow \mathbf{H}(loc(X, L))$
$IC_6 : \mathbf{H}(hasCode(X, C)) \rightarrow (\exists L)\ \mathbf{H}(codeLoc(C, L)), \mathbf{H}(loc(X, L))$
$IC_7 : \mathbf{H}(loc(X, L1)), \mathbf{H}(loc(X, L2)) \rightarrow L1 = L2$
$IC_8 : \mathbf{H}(loc(X, L)) \rightarrow \mathbf{H}(advertised(X))$

The database in [30] is mapped into the following history $\mathbf{HAP}$:

$$\{\mathbf{H}(codeLoc(ox1, central)), \mathbf{H}(codeLoc(ox1, south)),\\ \mathbf{H}(codeLoc(ox2, summertown)), \mathbf{H}(hasCode(prop1, ox2)), \mathbf{H}(ann(e1, price)),\\ \mathbf{H}(ann(e1, label)), \mathbf{H}(visible(e1)), \mathbf{H}(group(e1, prop1))\}$$

The $\mathcal{S}$CIFF proof procedure applies ICs in a forward manner, and it infers the following set of abducibles from the program above:

$$\mathbf{HAP'} = \{\mathbf{H}(priceElem(e1)), \mathbf{H}(forSale(prop1)), \exists P\ \mathbf{H}(price(prop1, P)),\\ \mathbf{H}(loc(prop1, summertown)), \mathbf{H}(advertised(prop1))\}$$

Each of the (ground) atomic queries (BCQs) outlined in [30] is also entailed in the $\mathcal{S}$CIFF program above. In particular, there exist a set $\mathbf{HAP'}$ such that:

$$\mathbf{HAP'} \models \mathbf{H}(priceElem(e1)), \mathbf{H}(forSale(prop1)), \mathbf{H}(advertised(prop1))$$

Also, the CQ $\exists L\ \mathbf{H}(loc(prop1, L))$ is entailed as well (with unification $L = summertown$, as in [30]) since:

$$\mathbf{HAP'} \models \mathbf{H}(loc(prop1, summertown))$$

[5] We show for the sake of clarity the quantification of existentially quantified variables, although in the $\mathcal{S}$CIFF syntax the quantification is implicit.

5 Integrating norms and ontologies in $\mathcal{S}$CIFF

After mapping a Datalog$^{\pm}$ ontology into $\mathcal{S}$CIFF, we are now ready to show how normative reasoning is smoothly integrated with ontological reasoning within the $\mathcal{S}$CIFF abductive framework. We will show this via a simple example, starting from the real estate ontology, and enriching it with normative rules for some interacting agents in a real estate scenario.

As discussed in Section 3, the set of regulations holding in the given society of agents can be expressed again through integrity constraints.

In the previous example, suppose that a real estate agent (rea) is in charge of selling a property, and it uses the data from the real estate information extracted from the ontological knowledge. If another agent has seen an announcement that a property (e.g., a flat) is for sale, it can request to buy it. In such a case, for some fixed time ΔT, the real estate agent is obliged to reserve the flat for that client. For ΔT time units, the real estate agent cannot sell the flat to other agents and, if the buyer issues a payment, the flat must be declared sold (within some deadline D). The expected behavior of the real estate agent can be defined by the following rules:

$$\mathbf{H}(tell(X, rea, buy(E), T), \mathbf{H}(forSale(E)), \mathbf{H}(price(E, P)),$$
$$\rightarrow \mathbf{EN}(sell(rea, Y, E), T_s), Y \neq X, T_s < T + \Delta T$$

$$\mathbf{H}(tell(X, rea, buy(E), T), \mathbf{H}(forSale(E)), \mathbf{H}(price(E, P)),$$
$$\mathbf{H}(pay(X, rea, P), T_p), T_p < T + \Delta T \rightarrow \mathbf{E}(sell(rea, X, E), T_s), T_s < T_p + D.$$

If some agent e asks to buy property $prop1$ at price $e1$ at time 1, i.e., $\mathbf{H}(tell(e, rea, buy(prop1), 1)$, the proof procedure is able to infer the following information about the expected behavior of the rea agent:

$$\forall T_s \text{ s.t. } T_s < 1 + \Delta T, \ \forall Y \neq e \qquad \mathbf{EN}(sell(rea, Y, prop1), T_s)$$

i.e., agent rea is not allowed to sell property $prop1$ to any other agent until ΔT time units have passed. Let us suppose that $\Delta T = 5$ and $D = 3$; if agent e actually executes the payment, e.g. $\mathbf{H}(pay(e, rea, e1), 3)$, agent rea is now expected to sell $prop1$, and the following expectation is raised:

$$\exists T_s \text{ s.t. } T_s < 6 \qquad \mathbf{E}(sell(rea, e, prop1), T_s).$$

In this way, not only the $\mathcal{S}$CIFF proof procedure is able to infer the knowledge from the ontological database, but also to provide the expected behavior of the agents (in our example, the rea agent) including obligations and prohibitions.

6 Conclusions and Future Work

We have shown that Abductive Logic Programming (ALP) is a powerful tool for knowledge representation and reasoning about norms and ontologies. We have

focused in detail about the $\mathcal{S}$CIFF ALP framework, used in the past to model and verify agent societies, interaction protocols for multi-agent systems [8], Web services choreographies [1], but also powerful enough to represent deontic operators [7] and normative systems [5]. Its underlying $\mathcal{S}$CIFF proof procedure [4], derived from the IFF one, can deal with both existentially and universally quantified variables in rule heads and CLP constraints. Recently, in [28, 29], $\mathcal{S}$CIFF has been also proved useful for representing ontologies expressed in Datalog$^{\pm}$.

In this work, we have exploited the $\mathcal{S}$CIFF framework for representing and integrating both norms and Datalog$^{\pm}$ ontologies. Normative, rule-based reasoning and ontological query answering are obtained by applying the $\mathcal{S}$CIFF abductive proof procedure. Both norms and a Datalog$^{\pm}$ theory can be encoded as $\mathcal{S}$CIFF integrity constraints. The main advantage is that this integration is achieved within a single language, grounded on ALP. Nonetheless, different ALP approaches might be possible.

A number of issues are subject of future work. First of all, we have not focused here on complexity results. Identifying syntactic conditions that guarantee tractable ontologies in $\mathcal{S}$CIFF, in the style of what has been done for Datalog$^{\pm}$, is crucial for achieving nice computational performance.

A second issue for future work concerns experimentation with real cases, in the normative and legal domain, and on real-size ontologies.

Finally, different mappings of Datalog$^{\pm}$ to ALP might exist, possibly enjoying different properties: another research direction is oriented toward identifying the mapping that suits best for legal reasoning.

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Montali, M.: An abductive framework for a-priori verification of web services. In: Maher, M. (ed.) Proceedings of the Eighth Symposium on Principles and Practice of Declarative Programming. pp. 39–50. ACM Press, New York, USA (Jul 2006)
2. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based software tool. Applied Artificial Intelligence 20(2-4), 133–157 (Feb-Apr 2006)
3. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Security protocols verification in abductive logic programming: a case study. In: Dikenelli, O., Gleizes, M.P., Ricci, A. (eds.) ESAW 2005 Post-proceedings. pp. 106–124. No. 3963 in LNAI, Springer-Verlag, Kusadasi, Aydin, Turkey (2006)
4. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the SCIFF framework. ACM Transactions on Computational Logic 9(4) (2008)
5. Alberti, M., Gavanelli, M., Lamma, E.: Deon+ : Abduction and constraints for normative reasoning. In: Artikis, A., Craven, R., Cicekli, N.K., Sadighi, B., Stathis, K. (eds.) Logic Programs, Norms and Action - Essays in Honor of Marek J. Sergot on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 7360, pp. 308–328. Springer (2012)
6. Alberti, M., Gavanelli, M., Lamma, E.: The CHR-based implementation of the SCIFF abductive system. Fundamenta Informaticae 124(4), 365–381 (2013)

7. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Sartor, G., Torroni, P.: Mapping deontic operators to abductive expectations. Computational and Mathematical Organization Theory 12(2–3), 205 – 225 (Oct 2006)
8. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and verification of agent interactions using social integrity constraints. Electronic Notes in Theoretical Computer Science 85(2) (2003)
9. Alberti, M., Gomes, A.S., Gonçalves, R., Leite, J., Slota, M.: Normative systems represented as hybrid knowledge bases. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) CLIMA XII. LNAI, vol. 6814 (2011)
10. ALFEBIITE: A Logical Framework for Ethical Behaviour between Infohabitants in the Information Trading Economy of the universal information ecosystem. IST-1999-10298 (1999)
11. Arisha, K.A., Ozcan, F., Ross, R., Subrahmanian, V.S., Eiter, T., Kraus, S.: IMPACT: a Platform for Collaborating Agents. IEEE Intelligent Systems 14(2) (1999)
12. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational and Mathematical Organization Theory 12, 71–79 (2006)
13. Boella, G., van der Torre, L.: Attributing mental attitudes to normative systems. In: Rosenschein, J.S., Sandholm, T., Wooldridge, M., Yokoo, M. (eds.) Proc AAMAS-2003. pp. 942–943. ACM Press (Jul 14–18 2003)
14. Bracciali, A., Demetriou, N., Endriss, U., Kakas, A.C., Lu, W., Mancarella, P., Sadri, F., Stathis, K., Terreni, G., Toni, F.: The KGP model of agency for global computing: Computational model and prototype implementation. In: Priami, C., Quaglia, P. (eds.) Global Computing. LNCS, vol. 3267. Springer (2004)
15. Broersen, J., Dignum, F., Dignum, V., Meyer, J.J.C.: Designing a deontic logic of deadlines. In: Lomuscio, A., Nute, D. (eds.) DEON. Lecture Notes in Computer Science, vol. 3065, pp. 43–56. Springer (2004)
16. Bürckert, H.: A resolution principle for constrained logics. Artificial Intelligence 66, 235–271 (1994)
17. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. In: International Conference on Principles of Knowledge Representation and Reasoning. pp. 70–80. AAAI Press (2008)
18. Calì, A., Gottlob, G., Lukasiewicz, T.: A general Datalog-based framework for tractable query answering over ontologies. In: Symposium on Principles of Database Systems. pp. 77–86. ACM (2009)
19. Calì, A., Gottlob, G., Lukasiewicz, T.: Tractable query answering over ontologies with Datalog$^{\pm}$. In: International Workshop on Description Logics. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
20. De Vos, M., Padget, J., Satoh, K.: Legal modelling and reasoning using institutions. In: Onada et al. [36], pp. 129–140
21. Dignum, V., Meyer, J.J., Dignum, F., Weigand, H.: Formal specification of interaction in agent societies. In: Proceedings of the Second Goddard Workshop on Formal Approaches to Agent-Based Systems (FAABS), Maryland (Oct 2002)
22. Dignum, V., Meyer, J.J., Weigand, H.: Towards an organizational model for agent societies using contracts. In: Castelfranchi, C., Lewis Johnson, W. (eds.) AAMAS-2002. pp. 694–695. ACM Press, Bologna, Italy (Jul 15–19 2002)
23. Dignum, V., Meyer, J.J., Weigand, H., Dignum, F.: An organizational-oriented model for agent societies. In: Proc. International Workshop on Regulated Agent-Based Social Systems: Theories and Applications. AAMAS'02, Bologna (2002)
24. Du, J., Wang, K., Shen, Y.D.: A tractable approach to ABox abduction over description logic ontologies. In: Brodley, C., Stone, P. (eds.) Proc. AAAI 2014 (2014)

25. Eiter, T., Subrahmanian, V., Pick, G.: Heterogeneous active agents, I: Semantics. Artificial Intelligence 108(1-2), 179–255 (March 1999)
26. ESTRELLA: European project for Standardized Transparent Representations in order to Extend Legal Accessibility. IST-2004-027655 (2004), home Page: `http://www.estrellaproject.org`
27. Fung, T.H., Kowalski, R.A.: The IFF proof procedure for abductive logic programming. Journal of Logic Programming 33(2), 151–165 (Nov 1997)
28. Gavanelli, M., Lamma, E., Riguzzi, F., Bellodi, E., Zese, R., Cota, G.: An abductive framework for Datalog$^{\pm}$ ontologies. In: Eiter, T., Toni, F. (eds.) Technical Communications of the 31st Int'l. Conference on Logic Programming (ICLP 2015). CEUR Workshop Proceedings, Sun SITE Central Europe, Aachen, Germany (2015)
29. Gavanelli, M., Lamma, E., Riguzzi, F., Bellodi, E., Zese, R., Cota, G.: Abductive logic programming for Datalog$^{\pm}$ ontologies. In: Ancona, D., Maratea, M., Mascardi, V. (eds.) Proceedings of the 30th Italian Conference on Computational Logic (CILC2015), Genova, Italy, July 1-3, 2015. CEUR Workshop Proceedings, vol. 1459, pp. 128–143. Sun SITE Central Europe, Aachen, Germany (2015)
30. Gottlob, G., Lukasiewicz, T., Simari, G.I.: Conjunctive query answering in probabilistic Datalog+/- ontologies. In: International Conference on Web Reasoning and Rule Systems. LNCS, vol. 6902, pp. 77–92. Springer (2011)
31. Governatori, G., Rotolo, A.: BIO logical agents: Norms, beliefs, intentions in defeasible logic. Autonomous Agents and Multi-Agent Systems 17(1), 36–69 (2008)
32. Jaffar, J., Maher, M.: Constraint logic programming: a survey. Journal of Logic Programming 19-20, 503–582 (1994)
33. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive Logic Programming. Journal of Logic and Computation 2(6), 719–770 (1993)
34. Klarman, S., Endriss, U., Schlobach, S.: ABox abduction in the description logic $\mathcal{ALC}$. J. Autom. Reasoning 46(1), 43–80 (2011)
35. Montali, M., Torroni, P., Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P.: Verification from declarative specifications using logic programming. In: de la Banda, M.G., Pontelli, E. (eds.) Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings. Lecture Notes in Computer Science, vol. 5366, pp. 440–454. Springer (2008)
36. Onada, T., Bekki, D., McCready, E. (eds.): New Frontiers in Artificial Intelligence - JSAI-isAI 2010 Workshops, LNCS, vol. 6797. Springer (2011)
37. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics 7(1) (1997)
38. Sadri, F., Stathis, K., Toni, F.: Normative KGP agents. Computational & Mathematical Organization Theory 12(2-3), 101–126 (2006)
39. Sartor, G., Casanovas, P., Biasiotti, M.A., Fernández-Barrera, M. (eds.): Approaches to Legal Ontologies. Law, Governance and Technology, Springer (2011)
40. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In: Onada et al. [36]
41. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. Commun. ACM 29, 370–386 (1986)
42. Van Belleghem, K., Denecker, M., De Schreye, D.: A strong correspondence between description logics and open logic programming. In: Naish, L. (ed.) Proc. Fourteenth International Conference on Logic Programming. MIT Press (1997)
43. Wright, G.: Deontic logic. Mind 60, 1–15 (1951)

Predictability of Agent in Legal Cases

Tetsuji Goto and Satoshi Tojo

School of Information Science, JAIST,
19th Floor, Shinagawa Inter-City Tower A, Konan 2, Minato-ku, Tokyo, Japan
{goto.tetsuji,tojo}@jaist.ac.jp
http://www.jaist.ac.jp/

Abstract. In criminal cases, we need to decide whether the crime is caused intentionally or not by the accused and this judgement is greatly influenced by the predictability of results and the awareness of facts. We have modeled several criminal cases in Dynamic Epistemic Logic (DEL) and Action Model. But simple DEL can not handle the predictability well, because the prediction in legal cases depends on the degree of perception of agents and differs from knowledge states. It often occurs that .the accused dare to take the action knowing the illegality indefinitely. In this paper, we introduce the concept of awareness into DEL or Action Model of multi-agent to reason about the predictability and model the sensitive cases such as the willful negligence. Furthermore, we implement an extended DEMO (Dynamic Epistemic MOdeling), including the awareness in DEL and Action Model, and observe the relation between the judgements of cases and the epistemic states or awareness.

Keywords: Dynamic Epistemic Logic, Action Model, Model Checking, Penal Code

1 Introduction

To determine a criminality specified in the Penal Code, the judge examines whether the accused's action comes under the external factors defined by the Penal Code (*Actus reus*)[1]. If these conditions are matched, the criminality is decided. (We take the position that the external factor includes the intent and the lapse. [20] There are also several opposite theories against this, such that both intent and lapse are regarded as the responsibility and should not be included in the external factors [25].)

The evaluation of the accused's action by intent (*Mens rea*)[2] or by lapse[3] is greatly influenced by the predictability of results or causation. For example, the intent is determined based on the awareness about a fact and the prediction of a causation. For lapse, the predictability and the duty to prevent the result are the issue.

[1] The guilty acts or typified criminal acts, sometimes called as the objective element of a crime
[2] A guilty mind or an intention to commit a crime
[3] A failure to take reasonable care when they act by taking account of the potential harm to other people

1.1 Awareness and Predictability in the Penal Code

Intent in the Penal Code Intent is "the intent to commit a crime". (The Penal Code Article 38 paragraph 1 [23]). At least, awareness of an objective external factor such as an action, a result and prediction for causation are needed. Furthermore, in general, the probability of occurrence or the admittance of the results by the accused [4] are taken into account by the judge. There is an uncertain intent, for example the willful negligence (*dolus eventualis*)[4] is classified as this type.
Lapse in the Penal Code The lapse(*negligentia*) is defined in the Penal Code Article 38 paragraph 1 as "The action without intent to commit a crime, and it is not punishable. However, if there is a special provision in the code, this shall not be applied to." [23]. The lapse is applied in the case where the accused did not foresee the result which might have been able to predict. Recently the duty of the accused to avoid criminal results is more emphasized [10].
Causation A relationship between an action and a result is called causation. In order to affirm the causation, there should be not only a conditional relationship (without that there should not be this) between an action and a result, but also it is required to be regarded reasonable from the experience of the social life of ordinary people (Legally sufficient cause [25]).

1.2 Motivation and related works

Dynamic Logic [7][2][13] or DEL (Dynamic Epistemic Logic) [1] can represent the change of states or epistemic states and have been applied to describe belief revision. In the field of legal reasoning, extended DEL, such as with the dynamic operator of agent's commitment, or of the degree of reliability of information sources, are used to model the cases [15][17]. On the other hand, there are related works dealing with agents' intention (such as [22][12]) in which the intention is defined as the one of multi modalities. We select DEL to describe the intention, prediction and predictability because we need to evaluate the degree of intention itself.

Previously we investigated the evaluation process of the external factors and tried to model the criminal cases by DEL and Action Model [11] because Action Model in DEL can represent the local epistemic states and can update the states by various epistemic actions and found a problem with describing the prediction or predictability and the awareness which are concerning to the information in the agent's consciousness. Furthermore in modeling the precedents the description becomes very complicated and it is hard to deduce the judgement by manually, so the modeling tools which is adequate for the legal cases are expected.

Related to the issue about awareness, the awareness logic [8] which originates from the omniscience problem is developed and the dynamics of the awareness and the concept of explicit knowledge are studied (for example, [3]). This logic is extended to belief revision [24]. We think this concept of awareness can be used to describe the prediction of the agent in legal cases.

[4] The accused is uncertain about the realization of crime but knowing that crimes may be implemented and he has accepted it.

In this paper, first we summarize DEL and Action model in section2, and expand these models by introducing the concept of awareness in section3. Then we expand DEMO (Dynamic Epistemic MOdeling) [6] to handle this concept and use this software to check the selected criminal cases in section 4 and discuss these result in section 5. Finally we conclude and point out the future work to be investigated.

2 Preliminary

2.1 DEL and Action Model

Knowledge and belief are not static because of the communication between agents. DEL is an extension of epistemic logic [14] with dynamic operators '[ ]', and $[\pi]\varphi$ is read as "successfully executing program π yields a φ state" [5]. Namely, given a model M and a possible world s,

$$M,s \models [\pi]\varphi$$

iff M is properly changed by the execution of π and as a result φ holds. Public Announcement Logic [21] is an example of DEL where the epistemic action is only restricted to public announcement.

Action Models [1] are used to describe epistemic actions. Let $\mathscr{L}$ be any logical language for given parameters agents A and atoms P. An $S5$ action model M is a structure $\langle \mathrm{S}, \sim, \mathrm{pre} \rangle$ where S is a domain of action points and for each agent a, $\sim_a$ is an equality relation in S, stating that two states are indistinguishable for a. pre: $\mathrm{S} \to \mathscr{L}$ is a preconditions function that assigns a formula in $\mathscr{L}$ to each $s \in S$.

An epistemic state can be changed by an epistemic action, so the new state after updating is described as a pair of an old world with an action that has taken place in that state. The expression (s, s) indicates that action s is executable in the state s. $M,s \models \mathrm{pre}(\mathrm{s})$ The two factual states are indistinguishable, if the following relation exists.

$$(s,\mathrm{s}) \sim_a (t,\mathrm{t}) \text{ iff } s \sim_a t \text{ and } \mathrm{s} \sim_a \mathrm{t} \text{ (in S5 action model)}$$

The updated model $M'(= M \otimes \mathrm{M})$ is a restricted modal product of an epistemic model and an action model, which is defined as an structure $\langle S', \sim', V' \rangle$ where $S' = \{(s,\mathrm{s}) \mid s \in S, \mathrm{s} \in \mathrm{S} \text{ and } M,s \models pre(\mathrm{s})\}$

Definition 1 (Syntax of Action Model Language).
The language of action model logic is the union of the formulas of static epistemic logic and that of epistemic actions.
$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid C_B\varphi \mid [\alpha]\varphi$
$\alpha ::= (\mathrm{M},\mathrm{s}) \mid (\alpha \cup \alpha)$

Definition 2 (Semantics of Action Model).
The semantics of Action Model can be defined as follows. The first 5 definitions are the same as the logic of Public Announcement with Common Knowledge.
$M,s \models p$ iff $s \in V_p$
$M,s \models \neg\varphi$ iff $M,s \not\models \varphi$
$M,s \models \varphi \wedge \psi$ iff $M,s \models \varphi$ and $M,s \models \psi$

$M,s \models K_a\varphi$ iff for all $s' \in S : s \sim_a s'$ implies $M,s' \models \varphi$
$M,s \models C_B\varphi$ iff for all $s' \in S : s \sim_B s'$ implies $M,s' \models \varphi$
$M,s \models [\alpha]\varphi$ iff for all $M',s' : (M,s)[\![\alpha]\!](M',s')$ implies $M',s' \models \varphi$
where $(M,s)[\![\mathrm{M,s}]\!](M',s')$ iff $M,s \models \mathrm{pre(s)}$ and $(M',s') = (M \otimes \mathrm{M},(s,\mathrm{s}))$

Example of Updating: May Read There are two epistemic states (0/1) where the proposition P is true (p) or false ($\neg p$) respectively. At first Agent a and b don't know whether the value of P is true or false and there is a link between 0 and 1 for a and b. This means that they cannot distinguish these states. Suppose that the event that Agent a *may read* a letter occurred, a knows p if he reads the letter that tells p and knows $\neg p$ if he reads the letter that contains $\neg p$, and does not know p or $\neg p$ if he does not read. But b can not distinguish these three points. After the action (*may read*) is executed, the new epistemic state consists of four states. The states that nothing happens is described as two states bottom of the right figure of Fig.1[5] where the link for agent a and b remains and two states above describe the occurrence of the read action. The underlines for the states in the figure indicate the actual worlds.

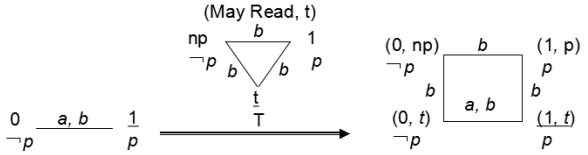


Fig. 1. state transition by an action May Read

2.2 Limited reasoning and awareness

Awareness logic [9] is an extension of the Epistemic Logic [14] to explain the problem of logical omniscience by making two kinds of information different which are implicitly known information which the agent can eventually get, and explicitly known information that the agent actually has in his conscious. The agent has to be aware of the information to make it explicitly known besides having it as implicit information. The syntax of awareness logic is as follows. The awareness logic has the added operator $A\varphi$: "the agent is aware of φ".

Definition 3 (Syntax of awareness logic).
$\varphi ::= p \mid A\varphi \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi$

Awareness logic extends Kripke models with a function A that assigns a set of formulas to the agent in each possible world. The semantics are as follows.

Definition 4 (Semantics of awareness logic).
the model is described as $\langle W,R,A,V\rangle$ where
W: a set of worlds
R: an accessibility relation ($R \subseteq (W \times W)$)
V: a valuation ($\mathrm{P} \to \mathscr{P}(W)$)
A: an awareness by a function $\mathrm{A} : (W \to \mathscr{P}(\mathfrak{L}))$
$M,s \models K\varphi$ iff for all $s' \in S : s \sim s'$ implies $M,s' \models \varphi$
$(M,w) \models A\varphi$ iff $\varphi \in \mathrm{A}(w)$

For example, Fagin and Halpern [9] defined the explicit knowledge $\mathrm{Ex}\varphi := K\varphi \wedge A\varphi$, and Benthem et al. [3] defined it as $\mathrm{Ex}\varphi := K(\varphi \wedge A\varphi)$ using this definition of awareness.

3 Formalization

3.1 Multi-agent Dynamic Epistemic Logic and Action Model with Awareness

To describe the intention, we introduce the concept of awareness into the DEL. While referring to awareness logics, we describe the awareness as the subset of formulas of the knowledge at a state and add the awareness modal operator. The index of agent to this operator describes the difference between the states of awareness of agents. The syntax of Multi-agent Dynamic Epistemic Logic with awareness is as follows.The dynamic operator [ ] is as same as the one of DEL.

Definition 5 (Syntax of Dynamic Epistemic Logic with Awareness).
$\varphi ::= p \mid A_i\varphi \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\alpha]\varphi$
$\alpha ::= (\mathrm{E},\mathrm{s}) \mid (\alpha \cup \alpha)$
((E, s) is an action model defined below.)

The semantics of the above language are as follows.. The model are defined by four tuples which consists of three elements of DEL and an awareness element.

Definition 6 (Semantics of Dynamic Epistemic Logic with Awareness).
The model is defined as $M = \langle W, V, A_i, R_i \rangle$
$M,s \models p$ iff $s \in V_p$
$M,s \models \neg\varphi$ iff $M,s \not\models \varphi$
$M,s \models \varphi \wedge \psi$ iff $M,s \models \varphi$ and $M,s \models \psi$
$M,s \models K_i\varphi$ iff for all $s' \in S : s \sim_i s'$ implies $M,s' \models \varphi$
$(M,w) \models A_i\varphi$ iff $\varphi \in \mathrm{A}(i,w)$
(function $\mathrm{A}(i,w)$: Agent $i \times W \to \varphi(\mathscr{L})$ gives the formulas the agent i is aware of at the state w.)
$M,s \models [\alpha]\varphi$ iff for all $M',s' : (M,s)[\![\alpha]\!](M',s')$ implies $M',s' \models \varphi$

It is reasonable to think the awareness is preserved under epistemic accessibility. And we will focus on the dynamics of awareness, so we take these assumptions described below concerning the relation of the worlds to avoid the complexity and to get a good prospect.

- introspection: If $M,s \models A_i\varphi$, then $M,t \models A_i\varphi$ for all t such that $(s,t) \in R_i$ (The awareness does not disappear.)
  In this context, the definitions of "explicit knows" become equivalent. ($K(\varphi \wedge A_i\varphi) \leftrightarrow K\varphi \wedge A_i\varphi$) [3]

- equality relation as R

We define the Action Model with awareness conditions which decides the awareness state after the action is executed. ($\mathfrak{L}$ stands for the language of this model)

Definition 7 (Action model with awareness condition for multi-agent).
$\mathscr{P}(\Sigma)$ indicates an arbitrary set of formulas in Σ.
the model is defined as E = $\langle$ S, Pre, Awc, R $\rangle$ where
Pre : s $\rightarrow \mathfrak{L}$ The precondition function is as same as Action Language and indicates the conditions where each event can be executed.
Awc : (Agent $i \times$ S $\times$ $\mathscr{P}(\mathfrak{L})) \rightarrow \mathscr{P}(\mathfrak{L})$ The awareness condition function, which assigning a new set of formulas in $\mathfrak{L}$ to every tuple of an agent, event, and previous (before the action) set of formulas in $\mathfrak{L}$.
R : (Agent $i \times$ S $\times$ S) the relation is as same as Action Model of DEL.

3.2 Updating of epistemic states by action

To describe the dynamic change of epistemic states in agents, we define the product update of epistemic model and action model for multi-agent with awareness.

Definition 8 (Product updating of epistemic states for multi-agent with awareness).
Epistemic model with awareness: $M = \langle W, V, A, R \rangle$ and
Action model with awareness conditions: $\mathrm{E} = \langle \mathrm{S},\ \mathrm{Pre},\ \mathrm{Awc},\ \mathrm{R} \rangle$,
Product updating: $M \otimes \mathrm{E} = \langle W', R', A', V' \rangle$ where
$W' := \{(w, \mathrm{s}) \mid (M, w) \models Pre(\mathrm{s})\}$
$V'(p) := \{(w, \mathrm{s}) \in W' \mid w \in V(p)\}$
$\mathrm{A}'(i, (w, \mathrm{s})) := \mathrm{Awc}(i, \mathrm{s}, \mathrm{A}_i(w))$
$(M, w) \models [(\mathrm{E},\ \mathrm{s})]\varphi$ iff $(M, w) \models \mathrm{Pre}(\mathrm{s})$ implies $(M \otimes \mathrm{E}, (w, \mathrm{s})) \models \varphi$
$R_i'(w, \mathrm{s})(w', \mathrm{s}')$ iff wR_iw' and $\mathrm{sR_is}'$

Example of updating: Aware of the relevant proposition (single agent case) There are four epistemic states (0/1/2/3) where the proposition P/Q is true (p/q) or false ($\neg p/\neg q$) respectively. At first Agent a doesn't know whether the value of P/Q is true or false and there is a link between 0, 1, 2 and 3 for a . Also he is aware of nothing at states 0,1,2 but is aware of p at state 3.. The action model consists of 2 points. The one is an action of doing something and the actor becomes aware of $q\ if\ p$. The other is an event that nothing happens. Then the initial epistemic states are updated by this action (may do) and the result states are 5 states which include a state where p is true and a state where p and q are true. (Fig.2) The agent is aware of only p at the initial state, but after the updating he is aware of both p and q by the awareness condition of this action (may do).

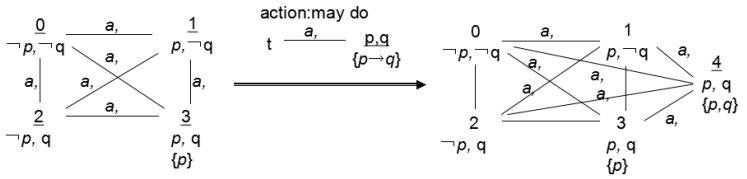


Fig. 2. Update by an action (may do)

4 Implementation and application

4.1 Implementation

DEMO [6] is a modeling tool for Dynamic Epistemic Logic and programmed in Haskell [16]. It allows modeling epistemic updates, display of action models, formula evaluation in epistemic models, so DEMO can be used to check semantic intuitions about what goes on in epistemic update situations. We extended DEMO to check the defined model with awareness. The awareness tuples are added to Epistemic model and Action model as following Haskell codes and related functions in these models are replaced to match these models.

```
-- The definition of Epistemic Model with awareness
data Model state formula awfrm = Mo
          [state]  -- a set of states/worlds
          [(state,formula)]  -- a set of valuations
          [(Agent,state,awfrm)]  --a set of awareness states
          [(Agent,state,state)]  --a set of relations
          deriving (Eq,Ord,Show)
-- The definition of Action Model with awareness
data ACM state = Acm
          [state]  -- a set of action/event
          [(state,Form)]  --a set of preconditions
          [(Agent,state,Subst)]  --a set of awareness conditions
          [(Agent,state,state)]  --a set of relations
          deriving (Eq, Show, Ord)
```

Further the updating functions are replaced to change the awareness states according to the conditions.

4.2 Criminal cases for application

Concerning the predictability in criminal cases, we consider that an agent can *predict* the consequence, if he/she is aware of the causal relation. In the judge's evaluation process, there are two problems where predictability is the issue The first is the intention of the accused and the issue of willful negligence contained in this problem. The second is the lapse which depends on the predictability and the possibility of avoiding results. To model these problems, some typical examples are selected and listed below [19] (Table 1).

Table 1. Sensitive criminal cases

Cases	Precedents	Outline
Case1 Willful negligence	Dealing with stolen goods case (No. RE238, 1947)	The accused bought stolen clothes without deterministic awareness that they were stolen. The judge applied the crime of paid acquisition of stolen goods which were stolen by another
Case2 (lapse) Predictability (delinquency of duty of care)	Hokkaido University electric scalpel case (No. U219, 1974)	The doctor and the nurse made a mistake of connecting the cable of the scalpel incorrectly, and the patient's right foot was damaged. The judge ruled professional lapse resulting in bodily injury to the nurse.

4.3 Common Action Models

To simulate the above cases, some action models are commonly used through cases and defined as follows and Table 2. We use notations of "agent *a*" for the accused and "agent *b*" for the judge.

- public: The actor publicly announces p to everyone. This action causes no influence on the awareness states of each agent at the world. (in our program, depicted as " `(public p)`" and this can be read as "p is announced publicly")
- see it: The actor sees the fact vaguely. It is not known that he is aware of the fact. (in our program, " `(seeit a p)`" and this can be read as "Agent *a* vaguely sees *p*.")
- perceive: The actor implicitly knows and perceives the fact about the action. In this paper, we assume the introspection, so this means he knows the fact explicitly. (in our program, " `(perceive b p)`" and this can be read as "The agent *b* explicitly knows *p*.")

Table 2. Common Action Models

Action	Definitions in the program		
public	Action	$\underline{0}$	
	Precondition	0: p	
	Aware condition	a 0:[]	b 0:[]
	Relation	a [0]	b [0]
seeit	Action	0, 1, $\underline{2}$	
	Precondition	0: q, 1: q, 2:$\top$	
	Aware condition	a 0:$\top \rightarrow q$, 1:[] 2:[]	b 0:[] , 1:[], 2:[]
	Relation	a [0,1],[2]	b [0,1,2]
perceive	Action	$\underline{0}$, 1	
	Precondition	0: p, 1: $\top$	
	Aware condition	a 0:[], 1:[]	b 0:$\top \rightarrow p$, 1:[]
	Relation	a [0,1]	b [0].[1]

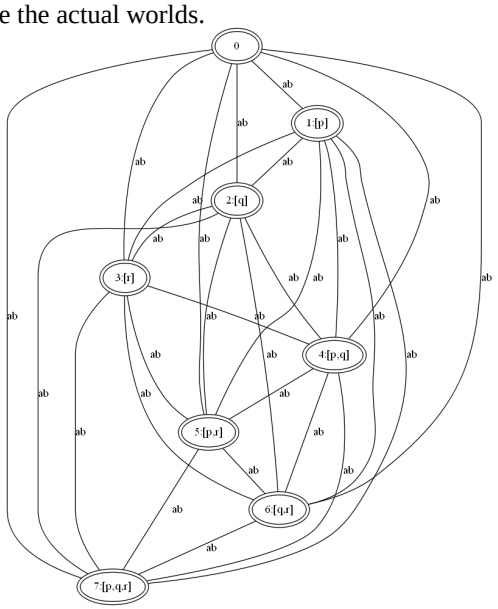
a, b: agents [](blank) in the Aware condition indicates no condition
The numbers in [] of the Relation indicates the equal events which can not be distinguished by the agent
The action points (events) are notated by the number and the actual events are underlined.

4.4 Case1：The Willful Negligence

The accused bought stolen clothes without deterministic recognition of their being stolen. He is accused of paid acquisition of stolen goods.
The propositions are as follows.

- *p*: The clothes are stolen.
- *q*: The seller is suspicious and in a hurry.
- *r*: The accused buys the stolen goods.

The initial state of the epistemic model is described in Fig. 3. None of the accused and the judge knows the information and is aware of the fact. The double lined circles indicate the actual worlds.



The awareness of Fig. 3

St	Aw
0	-/-
1	-/-
2	-/-
3	-/-
4	-/-
5	-/-
6	-/-
7	-/-

St: States, Aw: awareness for agent *a*/*b*
"-" indicates the agent is aware of nothing.

Fig. 3. The initial epistemic state

The specific action models used in this case are in Table 3.
The accused's actions which update the epistemic states through the crime are defined as follows by using the action models.

- "The accused think the seller is suspicious." `(seeit a q)`
- "The accused buy the clothes from the man." `(a_buy_cloth)`

The judge's actions which update the epistemic states through the trial are as follows.

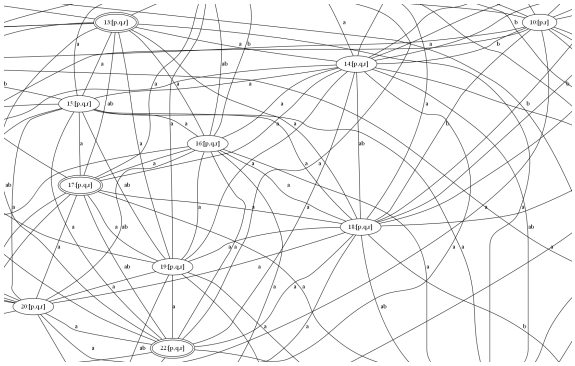
- "The judge convinces that the clothes are stolen." `(perceive b p)`
- "The judge verifies that the accused buy stolen goods." `(b_investigate)`

And the resulting states after these updating are in Fig. 4 and Table 4.
In Case1, the accused is not aware of the cause of crime (proposition *p*) in all actual

Table 3. The action models for case 1

Action	Definitions in the program		
a_buy_	Action	0, 1, 2	
clothes	Precondition	0: $\neg p \wedge \neg r$, 1: $p \wedge r$, 2: $p \wedge q \wedge r$	
	Aware condition	a 0:[], 1: p $\rightarrow$ r, q $\rightarrow$ r	b 0:[], 1:[]
	Relation	a [0],[1]	b [0,1]
b_investi-	Action	0, 1, 2	
gate	Precondition	0: $\neg p \wedge \neg r$, 1:$\top$, 2:$p \wedge r$	
	Aware condition	a 0,[], 1:[] 2:[]	b 0: $\neg p \rightarrow \neg r$, 1:[], 2:$p \rightarrow r$
	Relation	a [0,1,2]	b [0],[1],[2]

a, b: agents []blank) in Aware condition indicates there is no condition
The numbers in [] of Relation indicates the events which can not be distinguished by the agent
The action points (events) are notated by the number and the actual events are underlined.



Relations for agent a:{0,1,6,7},{2,3,4,5},{8,9,10, 11,21,23,26,27},{12,13,14,15, 16,17,18,19,20,22,24,25}
Relations for agent b:{0,2,4,6},{1,3,5,7,10,14,18, 24,26},{8,12,16,20,21},{9,13, 17,22,23},{11,15,19,25,27}
The states in the same bracket can not be distinguished by the agent.

Fig. 4. The final state of case 1 (a part around the actual worlds)

states (No. 13, 17, 22). This means that the accused is not convinced that this will result in a crime and corresponds to the willful negligence. On the other hand, he is aware of the criminal fact or external element that he buys stolen goods (proposition r) in some actual states and is not aware of in some actual worlds. This means that he can predict the crime by the awareness of supplemental elements (proposition q in this case "The accused thinks the seller is suspicious.") and indeed he predicts the crime in some scenarios. The judge knows and is aware of the crime (proposition r) in all actual state. This means he makes this judgement with convinced awareness. To this conclusion, he proves the cause of a crime (proposition p in this case "The clothes are stolen.") and he finds out the accused's partial (in some actual worlds) or indeterministic intention (He dares to take a criminal action.).

4.5 Case2：The delinquency of duty of care

The doctor and the nurse made a mistake of connecting the cable of the scalpel incorrectly, and the patient's right foot below knee was damaged and resulted in an amputation. The nurse is accused of professional lapse resulting in injury.
The propositions are as follows.

– p: The broken electric circuit without a safety breaker causes a dangerous electric current.

Table 4. States of case 1

St	Pr	Aw	St	Pr	Aw	St	Pr	Aw	St	Pr	Aw
0	-	-/-	7	q	-/-	14	p,q,r	q/-	21	p,q,r	-/p
1	-	-/-	8	p,r	-/p	15	p,q,r	q/-	22	p,q,r	-/p,r
2	q	q/q	9	p,r	-/p,r	16	p,q,r	q,r/p	23	p,q,r	-/p,r
3	q	q/q	10	p,r	-/-	17	p,q,r	q,r/p,r	24	p,q,r	-/-
4	q	-/-	11	p,r	-/-	18	p,q,r	q,r /-	25	p,q,r	-/-
5	q	-/-	12	p,q,r	q/p	19	p,q,r	q,r/-	26	p,q,r	-/-
6	q	-/-	13	p,q,r	q/p,r	20	p,q,r	-/p	27	p,q,r	-/-

St: States (The actual states are underlined.), Pr: Propositions which are valid at the state,
Aw: The formulas in the awareness of agent a/b
"-" indicates there is no valid proposition or the agent is aware of nothing.

- q: The wrongly connected cables causes an dangerous electric circuit.
- r: The patient is injured by the operation.

The initial state of the epistemic model is as same as that of Case1.
The specific action models used in this case are as follows (Table 5).

Table 5. The action models for case 2

Action	Definitions in the program		
help ope	Action	0, 1	
	Precondition	0: $\top$, 1: $p \wedge r$	
	Aware condition	a 0:[], 1: p $\rightarrow$ r	b 0:[], 1:[]
	Relation	a [0],[1]	b [0,1]
think	Action	0, 1	
	Precondition	0: $\top$, 1: $p \wedge q$	
	Aware condition	a 0,[], 1:[]	b 0:[], 1:q $\rightarrow$ p
	Relation	a [0,1]	b [0],[1]

a, b: agents [](blank) in Aware condition indicates no condition
The numbers in [] of Relation indicates the events which can not be distinguished by the agent
The action points (events) are notated by the number and the actual events are underlined.

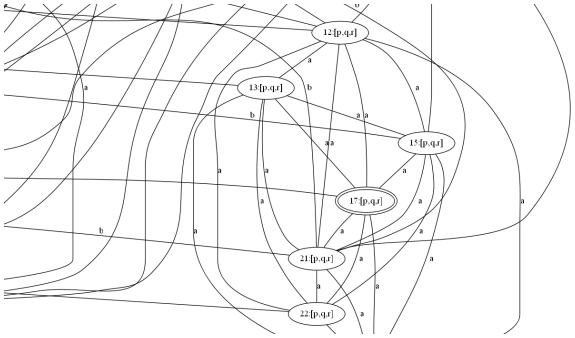
The accused's actions which update the epistemic states through the crime are as follows.

- "It is generally accepted that wrong connected cables cause abnormal electric current." `(public q)`
- "The accused sets the electric scalpel and helps the operation." `(a_help_ope)`

The judge's actions which update the epistemic states through the trial are as follows.

- "The judge convinces that the cables are connected wrongly and the abnormal electric current happens." `(perceive b q)`
- "The judge presumes that the wrongly connected cables cause the abnormal electric current and accused buy stolen goods." `(b_think)`
- "The judge verifies that the accused injures the patient by mistake." `( b_investigate)`

And the result of updating is in Fig. 5 and Table 6.
In Case2, the accused is not aware of the criminal fact or external element (proposition



Relations for agent *a*:
{0,1,6,7},{2,3,4,5},{8,9,10,11,
21,23,26,27},{12,13,14,15,16,
17,18,19,20,22,24,25}
Relations for agent
b:{0,2,4,6},{1,3,5,7,10,14,18,
24,26},{8,12,16,20,21},{9,13,
17,22,23},{11,15,19,25,27}

Fig. 5. The final state of case 2 (a bold underline indicates the actual world)

Table 6. States of case 2

St	Pr	Aw	St	Pr	Aw	St	Pr	Aw	St	Pr	Aw
0	q	-/q	6	p,q	-/-	12	p,q,r	-/q	18	p,q,r	-/-
1	q	-/q	7	p,q	-/-	13	p,q,r	-/q	19	p,q,r	-/-
2	q	-/-	8	q,r	-/q	14	p,q,r	-/p,q	20	p,q,r	-/-
3	q	-/-	9	q,r	-/-	15	p,q,r	-/p,q	21	p,q,r	-/-
4	p,q	-/q	10	p,q,r	-/q	16	p,q,r	-/p,q,r	22	p,q,r	-/-
5	p,q	-/p,q	11	p,q,r	-/q	17	p,q,r	-/p,q,r	23	p,q,r	-/-

St: States (The actual states are underlined.), Pr: Propositions which are valid at the state, Aw: the formulas in the awareness of agent *a*/*b*
"-" indicates there is no valid proposition or the agent is aware of nothing.

r) in all actual worlds (No. 17). This means he has not predicted the result at all. On the other hand, the judge is aware of both the cause of crime (proposition *p*) and the criminal fact (proposition *r*). From this view, he can conclude that generally the result can be predicted, but the prediction is neglected by the accused.

4.6 Discussion

According to these observations, followings can be suggested. In both cases, the judge is aware of the cause (proposition *p*) and the external factor (proposition *r*) in all actual cases, so he/she can make a judgement with confidence. On the other hand, the accused is not aware of the cause and is or is not aware of the external factor in actual cases. This distinguishes the lapse and willful negligence.

- For the non-deterministic criminal intention cases, such as willful negligence, the existence of a world where the accused is aware of the criminal fact can be seen. This can be modeled by the expression as follows.
  $\langle\alpha\rangle A_i\varphi$ (α: criminal action, φ: criminal fact)
- For the lapse cases, there is no world where the accused is aware of the criminal fact, but the judge can be aware of the fact. This is described by the expression as follows.
  $\neg\langle\alpha\rangle A_i\varphi \wedge [\alpha]A_j\varphi$ (α: criminal action, φ: criminal fact)

5 Conclusion and Further Directions

5.1 Conclusion

We have searched the criminal cases on the border line of intention and lapse and selected some typical cases. Through this observation, it can be seen that the predictability plays an important role in judgement.

We used the awareness in DEL and Action models to describe the predictability of agents, which are realized by adding an aware state function of agents, worlds to DEL and by adding an aware condition function of agents, worlds, and awareness states to Action Model. We extended DEMO to handle the above model with awareness. And using this extended program, we reproduce the knowledge states and the awareness sates of the accused and the judge respectively, according to the process of the crimes and the judgements of selected typical cases.

Through the observation of these results, we find that the existence of a world where the agent is aware of the criminal fact is important and suggest that the model patterns for willful negligence and lapse. We showed the usefulness of DEL and action model with the awareness to describe the sensitive cases where the predictability is a issue.

5.2 Further Directions

To describe lapse or intension, the awareness is useful, but concerning to the application of this concept to the real cases, there are some problems to be solved.

In implementing these precedents, the number of the states expands and the computational complexity increased more rapidly than those of only knowledge states without awareness. We can easily guess that if we deal with actual cases in detail (for example, the action should be divided into some actions according to the progress of time and the propositions should be divided into some detailed propositions, too.), as the number of factors is much larger, the epistemic states and link between epistemic states would be too complicated. If we could reduce the complexity of actual cases to the tractable size, our analysis can be applied to more actual cases and can be verified. Summarizing and picking up patterns, or omitting the calculation of the states which don't have the relation to the actual worlds can be thought.

Furthermore to describe the more real and complicated cases, the conditions of becoming aware should be studied further. We defined the awareness conditions in Action Model in advance, but in the judgement, the verification of the awareness condition would be the main issue. For example, the restrictions from the preconditions and the epistemic states should be considered form the legal view points.

References

1. Baltag, A. Moss, L.S. 2004. Logics for epistemic program: Synthese, vol.139 Issue 2, pp.165-224.
2. Benthem, J.V. Eijck J.V. Kooi, B. 2006. Logics of communication and change. In: Information and Computation Vol. 204, pp.1620-1662. Elsevier.

3. Benthem, J.V. and Velázquez-Quesada, F.R. 2010. The dynamics of awareness. Synthese vol. 177, pp.5-27, Springer.
4. Dandou, S. 1990. The criminal law essentials, general remarks, 3rd edit. Soubunsya (in Japanese).
5. Ditmarsch, H.V. et al. 2008. Dynamic Epistemic Logic. Springer.
6. Eijck, J.V. 2004. Dynamic epistemic modeling: Technical Report. Centrum voor Wiskunde en Informatica, Amsterdam.
7. Eijck, J.V. and Wang, Y. 2008. Propositional Dynamic Logic as a Logic of Belief Revision. In: Logic, Language, Information and Computation, pp.136-148. Springer.
8. Fagin R., Halpern, J.Y., Moses, Y. and Vardi, M.Y. 1995. Reasoning about Knowledge. The MIT Press.
9. Fagin, R. and Halpern, J.Y. 1988. Belief, awareness, and limited reasoning. Artificial Intelligence, 34(1), pp.39-76.
10. Fujiki, H. (eds.) 1975. Criminal lapse, a dispute on the old and new lapse theory. Gakuyoushobou (in Japanese)
11. Goto, T. and Tojo, S. 2014. Classification of Precedents by Modeling Tool for Action and Epistemic State: DEMO In:Proceedings of the 8h International Workshop on Juris-Informatics.
12. Governatori, G. and Rotolo, A. 2008. BIO logical agents: Norms, beliefs, intentions in defeasible logic. Autonomous Agents and Multi-Agent Systems 17(1), pp.36-69.
13. Harel, D., Kozen, D. and Tiuryn, J. 2000. Dynamic Logic. Foundations of Computing. Massachusetts: MIT Press.
14. Hintikka, J. 1962. Knowledge and belief: An introduction to the logic of the two notions. Ithaca,NY: Cornell University Press.
15. Jirakunkanok, P. et al. 2014. Analyzing Reliability Change in Legal Case In: Proceedings of the 8h International Workshop on Juris-Informatics.
16. Jones, S. P. (eds.). 2002. Haskell 98 Language and Libraries. `http://www.haskell.org/onlinereport/`
17. Katsuhiko, S. et al. 2012. Misconception in Legal Cases From Dynamic Logical Viewpoints. In: Proceedings of the 6h International Workshop on Juris-Informatics.
18. Mayer, M. E. 1923. Der allgemeine Teil des deutsch Strafrechts, 2. C. Winter, Heidelberg.
19. Nishida, N. Yamaguchi, A. Saeki, H. 2008. The 100 selected precedents of the Penal Code 1, 6th edit. Yuhikaku (in Japanese).
20. Oya, M. 2007. Lecture note on Criminal law: General part. Seibundo (in Japanese).
21. Plaza, J.A. 1989. Logics of public communications. In: Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems, pp.201-216.
22. Smith, C. et al. 2013. Reflex Responsibility of Agents. In: Proceedings of The 26th International Conference on Legal Knowledge and Information Systems (JURIX 2013), pp.135-144.
23. The Penal Code. `http://www.japaneselawtranslation.go.jp/law/detail/?id=1960`
24. Velázquez-Quesada, F.R. 2014.Dynamic Epistemic Logic for Implicit and Explicit Beliefs. Journal of Logic, Language and Information 23(2), pp.107-140.
25. Yamaguchi, A. 2006. Criminal law: General part. Yuhikaku (in Japanese).

A Method to Estimate Document Structure from Text Documnet and its Application to Law Articles

Yoichi Hatsutori and Katsumasa Yoshikawa

IBM Research Tokyo, IBM Japan Ltd.
19-21, Hakozakicho Nihonbashi, Chuo-ku, Tokyo, Japan
{yoichih, katsuy}@jp.ibm.com
http://www.research.ibm.com/labs/tokyo/index_j.shtml

Abstract. Active research on text analytics is accompanied by a growth in available electronic documents. Since text documents are stored as unstructured data, the growing diversity of documents is an issue in text analytics. Text documents have an internal document structure, like chapters, sections, subsections, and paragraphs. Since document structure is used to define the range of topics in a document, extracting document structure from diverse unstructured documents is an important task. However, unstructured documents do not have a common document structure, text documents are unique common information among unstructured documents. This paper proposes a rule-based approach to estimate document structure from text documents and demonstrates its application to legal documents.

Keywords. Document Structure, Article Extraction, Article Comparison, Text Analytics, Law Articles

1 Introdution

Electronic documents, e.g. office documents, web contents, articles, technical papers, and so on, are widely read and available online. Active research on text analytics, like natural language processing (NLP) and text mining, is accompanied by a growth in available text documents. However, text documents are often stored as unstructured data. The growing diversity of documents stored in unstructured data is an issue in text analytics.

Text documents have internal document structure, like chapters, sections, subsections, and paragraphs. Since hierarchical document structure is used to define the range of topics in a document, extracting document structure is an important task in text analytics. For example, structural information can be very useful in indexing and retrieving the information contained in the document [1]. In another example, estimated document structure can be used for eliminating needless words or characters in text analytics, e.g. section numbers, subsection numbers, and enumerations. Therefore, document structure has to be estimated in preprocessing of text analysis. However, unstructured documents do not have a common document structure. Moreover, the definition of document structure can differ even if text documents are stored in the

same file format, because ways to define document structure vary in accordance with the document's objective, its writers, and so on. For example, two ways to define list items, i.e. 1., 2., …, in Hyper Text Markup Language (HTML) format are shown in Figs. 1 and 2.

```
<ol>
  <li>list element</li>
  <li>list element</li>
</ol>
```

Fig. 1. One way to define list items in HTML

```
1. list element<br>
2. list element<br>
```

Fig. 2. Another way to define list items in HTML

In Fig. 1, HTML element <li> is used for specifying list items. On the other hand, plain text is used for writing list items in Fig. 2. Both outputs are the same on a display. Text documents are unique common information among unstructured documents. Thus, the document structure has to be estimated from text documents even if documents are stored in the same file format.

The contribution of this paper is to propose a method to estimate document structure from text documents. The proposed method is designed to support preprocessing of NLP and helps to add structure information to unstructured documents. Another contribution is to demonstrate an application of the proposed method to legal documents. Specifically, the proposed method is applied to article comparison in which it is used to extract articles from ordinances operating in some wards of Tokyo. Through calculating similarity scores among articles, similar articles are listed in a comparison table among wards.

Section 2 introduces related works on estimating document structure. Section 3 describes the processing flow of the proposed method. In Section 4, the proposed method is applied to analyze ordinances in some wards. Section 5 concludes this paper.

2 Related Works

In this section, previous works are listed and the scope of this paper is described. There are two types of approaches to estimate document structure. The first uses text description for estimating document structure. The second uses additional information attached to text description, e.g. tag information describing structural information. The latter approach is used for markup languages such as HTML.

2.1 Text-based approach

Fukui et al. [2] propose a text description based approach to extract document structure. In their research, pattern match is used for each line to extract document structure. Each line is assumed to conform to their knowledge based model called "Intra-line Structure Model", which consists of "heading", "delimiter", and "content" parts. They prepare about 300 patterns for the heading part, which consists of a "numeric part", "punctuation", and "reserved heading word". If a line has a heading part and the part is matched to a predefined pattern, then the line is extracted as a start point of document structure. The average accuracy of their structure extraction is 89.0%. However, their intra-line structure model assumes only the heading part can be the start point of a document structure and reserved heading words are required for detecting structure.

Déjean [3] proposes a robust method to detect long range numbered sequences applied to different types of documents. This paper specifically focuses on scanned, optical character recognition (OCR) and PDF documents. Only text description is used for estimating document structure. In Déjean's research, incremental types (e.g. digits, upper-case and lower-case letter, Roman letters, Chinese characters, or "Bis" sequences) are detected and extracted by predefined regular expressions. The document is scanned line by line, and numbered objects are extracted by pattern match. For every match, a pattern is associated with the line. A numbered object is recognized as a header part of the line, and the remaining part is recognized as the body part of the line. Document structure is estimated from extracted objects. The originality of this research is the notation of missing items (often due to OCR errors), and the notation of multi-increment items. This research activity also addresses intra-line structure, like Fukui et al.'s research.

Igari et al. [4] propose another method based on syntactic rule. When documents are described in a common format, it is possible to express the document structure by syntax rule for the document text. Because they focus on analyzing and parsing legal judgments which has a common format, their method has generality, extensibility and high accuracy. This approach is very useful and high accurate for documents which have common format. In our research, we do not assume that there is no common format in diverse documents. Thus syntactic approach is not applicable in our target documents.

2.2 Markup-based approach

If a document has meaningful tag information describing structural information, document structure can be extracted from it. Some open source libraries are available for extracting document structure from markup. For example, Apache Tika[1] is an open source library from which metadata and text can be extracted from over a thousand different file types.

[1] https://tika.apache.org/

However, these approaches assume that all structure information is defined in tags. When a file cannot use markup, e.g. plain-text and portable document file (PDF), markup-based approach cannot extract document structure.

2.3 The scope of this paper

Markup based approaches have a limit for applicable files. On the other hand, text description based approaches do not. In this paper, a text-based approach is used for processing diverse text documents. There are two ways for extracting document structure. One is a rule-based approach, and the other is a machine learning approach. In the machine learning approach, a large amount of training data is required to train the model for estimating document structure. However, preparing training data is a bottleneck for practical use. On the other hand, the rule-based approach does not need training data to be prepared. Therefore, the rule-based approach is applied for extracting document structure. Since this paper focuses on the method to estimate document structure, we assume that text is extracted from diverse documents preliminarily.

3 Proposed Method

This section describes a method to estimate document structure from a text document. Subsection 3.1 details two ways for defining document structure and explains the target structure in this paper. Subsection 3.2 describes the processing flow of proposed method.

3.1 Definition of document structure

There are two ways to define document structure in a document. One is to add additional text describing structural information. The other is to change the style of text.

In the first way, serial numbers are used for defining document structure. For example, numbered objects like 1., 2., 3., …, or Section 1, Section 2, … are added before each chapter, section, or subsection. For example, the document structure of this paper is also defined by additional text. Sections are defined by numeric numbers (1, 2, 3, …) and subsections are defined by multi-level numeric numbers (2.1, 2.2, 2.3, …). Defining by additional text is also used for enumerations in text. An example of this structural definition is shown in Fig. 3.In Fig. 3, additional texts are used for the structural definition, i.e. Section 1, Section 2, (A), (B), (1), (2), (a), and (b). Here, (a) and (b) are used for an enumeration in a sentence. The others are used for defining structure among sentences. Therefore, additional text also appears not only at the header part of a sentence but also in a sentence. In this study, not only the header part of each line but also enumerations in sentences is also considered in the structural estimation algorithm.

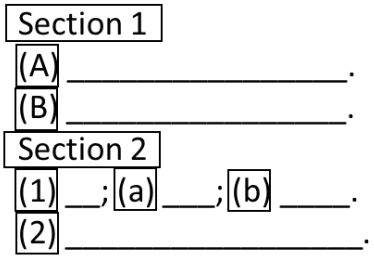


Fig. 3. Example of structural definition by additional texts

On the other hand, changing style is another way for defining structure. For example, font size, font style (e.g. bold, underline, and italics), alignment, indent, and so on can be changed. Since this paper focuses on a text-based approach, style information is not considered. However, even if text documents are stored in plain text format, indent or space characters can be used to define structure.

In this paper, we assumed that document structure is defined by additional texts. Estimating document structure from style information is out of this paper's scope. A method to estimate document structure is proposed in subsection 3.2.

3.2 Processing flow of document structure estimation

As mentioned in subsection 3.1, the proposed method assumes that document structure is defined by additional texts, e.g. Section 1, Section 2, or 1., 1.1, and so on. Note that document structure is a hierarchical structure. There are super-subrelations among the structure like relationships among sections-subsections. For example, there are two types of numbered objects, i.e. numeral numbers and alphabetic numbers in Fig. 4. Since text from "2." to "3." covers the text from "a." to "b.", we can estimate the following relationship: numeric numbers mean higher level structure than alphabetic numbers in this text document. This hierarchical structure has to be estimated from text documents.

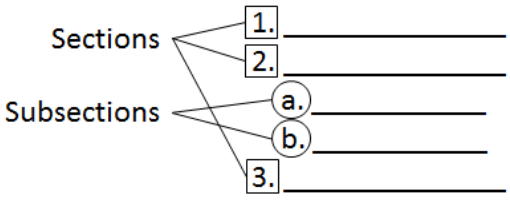


Fig. 4. Example of hierarchical document structure

The proposed method consists of four steps:

1. Extract candidate elements of document structure by using rules
2. Build candidate trees from extracted elements

3. Prune trees and identify unbranched trees
4. Identify super-subrelations among unbranched trees

The detailed process is described in the following sub-subsections.

Extract candidate elements of document structure by using rules.

In this step, candidate elements of document structure, e.g. chapter numbers, section numbers, numbered objects, are extracted by regular expressions. Fig. 5 shows parts of regular expressions used in this study. In this study, about 70 regular expressions are defined.

```
\s\([a-z]+\)\s
\s\([0-9]+\)\s
\s[0-9]+\.\s
[Ss][Ee][Cc][Tt][Ii][Oo][Nn]\s[0-9]+\.?\s
```

Fig. 5. Examples of regular expression for extracting candidate elements of document structure

For example, text shown in Fig. 6 is input to the proposed method. The candidate elements extracted by regular expressions are shown in Fig. 7.

```
Section 1.
(a) ----------.
(b) ---; (1) --, (2) --, and (3) --.
(c) ---; (1) ------, or (2) --------.
Section 2.
(a) ----------.
(b) ----------.
(c) ----------.
```

Fig. 6. Example of input text

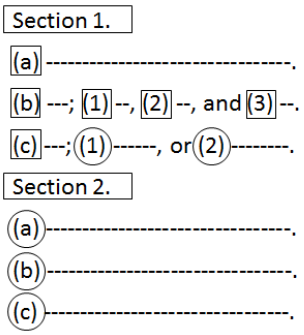


Fig. 7. Extracted elements (rectangle and circle parts are extracted by regular expressions)

After candidate elements are extracted by regular expressions, all extracted items are checked and filtered. For example, when a numbered object "3." is extracted by a rule and numbered objects "1." and "2." are not extracted before "3.", "3." is removed from candidate elements.

Build candidate trees from extracted elements.

To build a candidate tree of document structure, candidate elements extracted in the previous step are classified by the left-hand side character of each element. For example, extracted elements are classified into three groups by left-hand side character: linefeed code, punctuation, or alphabet. Note that if the left-hand word of candidates is a conjunction (i.e. "and", "or", or "and/or"), the conjunction is ignored and instead the next left-hand side character is used for classifying elements. For example, in Fig. 7, "Section 1", "Section 2", and alphabetic objects are classified into a "linefeed code" group and numeral objects are classified to a "punctuation" group.

Next, the candidate trees are built by predefined rules based on regular expressions and classifications. For example, a tree is built from elements extracted by a regular expression "\s\([a-z]+\)\s", and the left-hand side character is "linefeed code". In another example, a tree is built from elements extracted by a regular expression "\s\([0-9]+\)\s", and the left-hand side character is "punctuation". Here, the position of each element is also considered for building a candidate tree, i.e. in Fig. 7, the combination of the first "(a)" and the second "(b)" is a candidate. On the other hand, the combination of the first "(b)" and the second "(a)" is not a candidate because the first "(b)" appears before the second "(a)". Then, the candidate tree for the text in Fig. 7 is shown in Fig. 8. In Fig. 8, the words of sections, the alphabets such as "a" and "b", and the numerals such as "1" and "2" are separated and three groups of candidate trees are generated. The merits of grouping candidates and separating into small trees are; 1) reducing memory footprint to store candidate trees, 2) enabling us to estimate enumeration in one sentence in next step, and 3) isolating two problems, i.e. identification of a sequence of numbered objects such as "a-b-c" and identification of super-subrelationship among unbranched trees. In our method, a sequence of numbered

objects are identified first. Next, super-subrelationship among unbranched trees are identified.

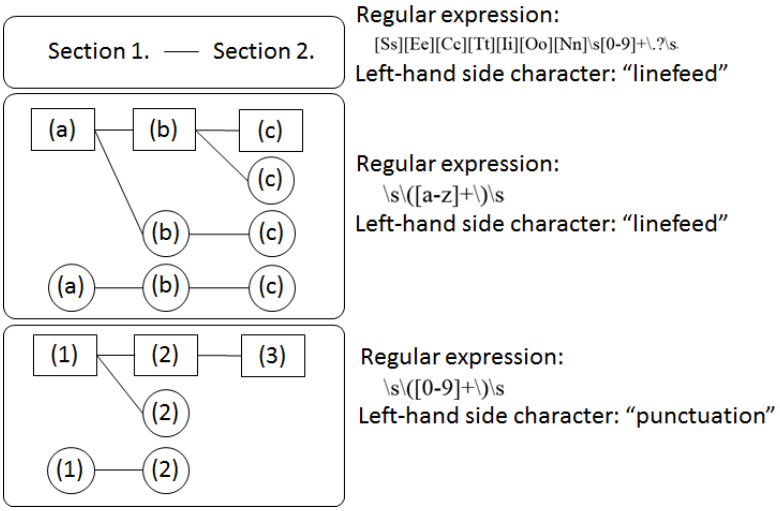


Fig. 8. Built candidate tree of document structure

Prune trees and identify unbranched trees.

As mentioned in subsection 3.1, the text document has two types of structure. One is a structure of enumerations in one sentence. The other is a structure among sentences. Then, pruning the tree is divided into two steps. The first focuses on estimating a structure of enumerations in one sentence. The second focuses on estimating a structure among sentences.

Step (1): estimate enumeration in one sentence.

Enumerations in one sentence are estimated in this step. Since target enumerations are in one sentence, a structure is estimated by checking a proposition ("Text from a root element to a leaf element is in one sentence") for each leaf by using an existing sentence splitter, and candidate trees are pruned on the basis of truth-value of the following proposition:

"True": prune the other branches

"False": check next leaf

For example, in Fig. 8, the combination of the first "(1)", "(2)", and "(3)" is in one sentence. On the other hand, the combination of the first "(1)" and the second "(2)" is not. As a result, the branch of the second "(2)" is pruned.

Through this step, enumerations in one sentence are estimated and the trees of enumerations become unbranched trees. The results of this pruning of the trees in Fig. 8 are shown in Fig. 9.

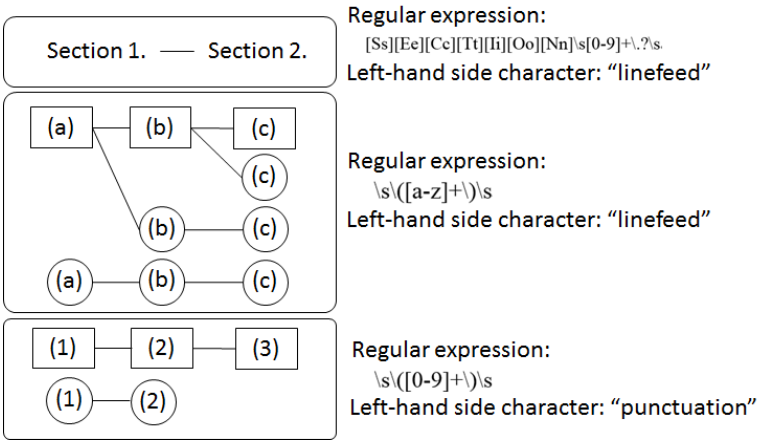


Fig. 9. Results of pruning in Step (1)

Step (2): estimate a structure among sentences.

To prune the remaining tree, unbranched trees are used for pruning the other trees. First, unbranched trees are identified. Second, if the elements in unbranched trees are used in the other branched trees, elements in branched trees are removed from them. For example, in Fig. 9, circled “(a)”, “(b)”, and “(c)” are in the unbranched tree. Then circled “(b)” and “(c)” in the other trees are removed. Third, if a branch crosses an unbranched tree, the crossed branch is pruned. Pruning in Step (2) is processed iteratively. If all candidates become an unbranched tree, the iteration is finished. The result of this pruning for the trees in Fig. 9 is shown in Fig. 10.

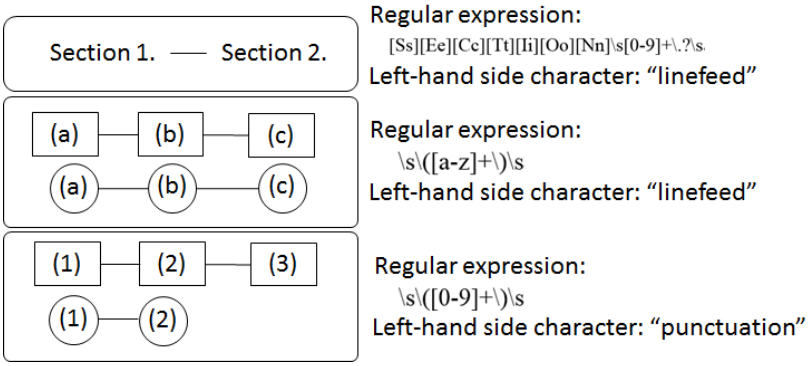


Fig. 10. Results of pruning in Step (2)

Here, if an iteration of the pruning in Step (2) cannot remove any elements and branches are still in candidates, predefined rules are used for pruning candidates. For example, the longest branch can be used as a correct branch.

Identify super-subrelations among unbranched trees.

Through pruning steps in the previous sub-subsection, all candidates become unbranched trees. Super-subrelations among unbranched trees are identified. The position of each element of unbranched trees is referred for identifying super-subrelations, because the pruning step ensures that there is no cross branch in candidates. For example, since text from "Section 1." to "Section 2." covers the text from rectangle-lined "(a)" to "(c)", we can estimate the following relationship: numeric numbers with a prefix "Section" mean higher level structure than rectangle-lined "(a)", "(b)", and "(c)" in this text document. The results of estimation for Fig. 10 are shown in Fig. 11.

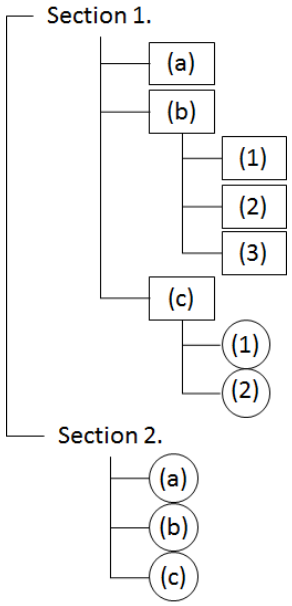


Fig. 11. Results of estimated document structure

4 Application in legal domain

In this section, the proposed method is applied to analyze legal documents. In this paper, we focus on analyzing some ordinances operating in wards of Tokyo. Articles are extracted by the proposed method (subsection 4.2), and a comparison table of articles is generated by calculating similarity among articles (subsection 4.3). Since ordinances are defined in every ward, ordinances need to be analyzed to establish or update ordinances, classify them, and identify issues of ordinances in operation. For example, Kakuta et al [5] describes a method and a real system which shows similar articles between every ordinances in all prefectures in Japan.

Since, an ordinance has an internal document structure (e.g. section, subsection, and articles), we demonstrate that the proposed method is applicable to extract articles from ordinances. Next, similarities among extracted articles are calculated and similar

articles among wards are aligned for creating a comparison table. Through this application, the effectiveness of the proposed method is discussed.

4.1 Sample Ordinances

In this paper, three wards in Tokyo (Minato, Shibuya, and Sumida) are targeted. The following three ordinances for each ward are picked out:

- Committee meeting
- Electoral management committee
- Establishment of secretariat of audit commission

Each ordinance has multiple articles, and some ordinances also have sections. Fig. 12 shows an example of an ordinance, which is a part of an electoral management committee regulation in Sumida Ward. This ordinance has a hierarchical document structure with sections and articles. Section or Article has a header part with numbered objects.

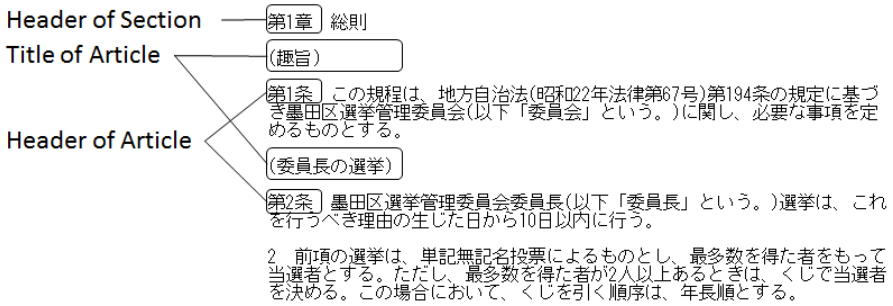


Fig. 12. Part of ordinance about establishment of secretariat of audit commission in Sumida Ward

4.2 Article Extraction

First, document structure is identified by the proposed method, and the text following after numbered objects is extracted as a corresponding body text of an article. The statistics of the result are shown in Table 1. As shown in Fig. 12, each article has its title with round brackets. Thus, titles are also extracted by pattern match with round brackets and linked to the corresponding article. An example of an extracted articles is shown in Fig. 13. In Table 1, the number of extracted articles are same to the number of the truth. In Fig. 13, starting and end points are correctly identified. Thus, the accuracy of the proposed method is confirmed.

Table 1. Results of article extraction

	Committee meeting		Electoral management committee		Establishment of secretariat of audit commission	
	Extracted	Truth	Extracted	Truth	Extracted	Truth
Minato Ward	28	28	12	12	13	13
Shibuya Ward	32	32	12	12	19	19
Sumida Ward	29	29	12	12	24	24

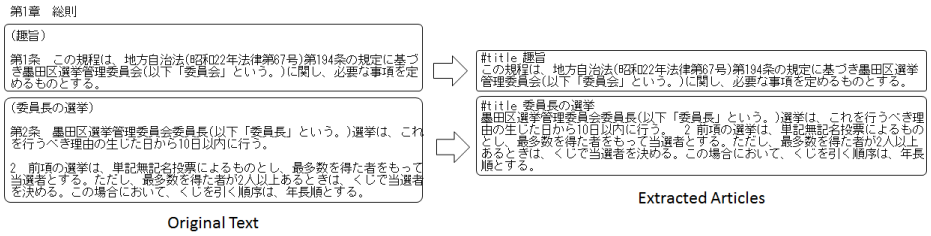


Fig. 13. Results of article extraction

4.3 Comparison table of articles

Extracted articles are compared among wards and summarized in a comparison table. Each article is represented by a Vector Space Model with TF-IDF term weighting [6, 7]. To find corresponding articles and generate a comparison table, we calculate the cosine similarities [8] between vectorised articles in different wards. Then, a comparison table is generated from this cosine similarity. The range of similarity score is 0.0 - 1.0, and higher scores are better. Each article is used as a query to search for similar articles in the other wards. The most similar articles are aligned and listed in a comparison table. Here, a threshold for deciding whether an article is similar or not is set to 0.5. This means if all similarities for an article are less than 0.5, no articles are aligned in the comparison table. A part of the result is shown in Table 2, in which two articles are analyzed among three wards. For the first article, an article operating in Shibuya Ward is used as a query, and similar articles operating in Minato and Sumida Wards are searched for. The highest scores are 0.79 and 0.77, respectively. The articles that have the highest scores are aligned next to the query article. On the other hand, for the second article, an article operating in Sumida Ward is used as a query, and similar articles operating in Shibuya and Minato Wards are searched for. However, the highest scores are 0.22 in Shibuya Ward and 0.14 in Minato Ward. These scores are less than the pre-defined threshold (i.e. 0.5). Thus, Shibuya and Minato Wards do not have corresponding articles in its ordinances, and no articles are aligned to the query article. The second query is defined in the ordinance about "establishment of secretariat of audit commission". As shown in Table 1, Sumida Ward has 24 articles in the ordinance, and the number of articles operating in Sumida Ward is larger than those in Minato and Shibuya Wards. This means Sumida Ward defines the ordinance in detail. Thus, this detailed article is not found in the other two wards.

Table 2. Comparison table among wards

		Shibuya Ward	Minato Ward	Sumida Ward
Article 1	Title	委員の選任	委員の選任	委員の選任
	Body Text	常任委員、議会運営委員及び特別委員(以下「委員」という。)は、議長が会議に諮って指名する。ただし、閉会中においては、議長が委員を選任することができる。（一部改正… 一九年三八号）２ 議長は、委員の選任事由が生じたとき、速やかに選任する。（改正… 二四年四九号）３ 議長は、常任委員の申出があるときは、会議に諮って当該委員の委員会の所属を変更することができる。ただし、閉会中においては、議長が所属を変更することができる。（一部改正… 一九年三八号）４ 前項の規定により所属を変更した常任委員の任期は、第三条(常任委員の任期)第二項の例による。	常任委員、議会運営委員及び特別委員(以下「委員」という。)の選任は、議長の指名による。２ 議長は、委員の選任事由が生じたときは、速やかに選任する。３ 議長は、常任委員の申出があるときは、当該委員の委員会の所属を変更することができる。４ 前項の規定により所属を変更した常任委員の任期は、第三条(常任委員の任期)第二項の例による。	常任委員、議会運営委員及び特別委員(以下「委員」という。)は、議長が指名し、会議に諮って選任する。ただし、閉会中においては、議長が指名し、選任することができる。２ 議長は、委員の選任事由が生じたときは、速やかに委員を選任する。３ 常任委員及び議会運営委員の任期満了による後任者の選任は、その任期満了前30日以内に行うことができる。４ 議長は、常任委員の申出があるときは、会議に諮って当該委員の委員会の所属を変更することができる。ただし、閉会中においては、議長が変更することができる。５ 第1項ただし書の規定により委員を選任したとき、及び前項ただし書の規定により委員の所属を変更したときは、議長は、その旨を次の議会に報告しなければならない。６ 第4項の規定により所属を変更した常任委員の任期は、第三条第2項の例による。（昭50条33・平3条20・平19条31・平25条1・一部改正）
	Score	Query	0.79	0.77
Article 2	Title			局長の専決事項
	Body Text			局長が専決できる事案は、次のとおりとする。 （1）所属職員の事務分担に関すること。 （2）所属職員の休暇、育児休業、欠勤、出張、職免、超過勤務、休日の勤務及び当該休日に代わる日の指定、週休日の勤務及び当該週休日の振替並びに研修の命令に関すること。 （3）諸証明及び公簿の閲覧に関すること。 （4）定例的又は軽易な通知、照会、報告等に関すること。 （5）前各号に掲げるもののほか、定例的又は軽易な事案の処理に関すること。 （平10選告10・一部改正）
	Score	0.22	0.14	Query

5 Conclusion

Actual context in processing unstructured and diverse documents raises challenges of estimating document structure from text documents. Our perspective in this paper is structural estimation from numbered objects. The proposed method is a rule-based approach and uses only text descriptions for estimating document structure from diverse documents. Thus, it can be used for supporting preprocessing of NLP and helping to add structure information to unstructured documents. For example, estimating document structure has the following merits as preprocessing of NLP: (1) the estimated range of chapters, sections, and subsections can be used for subsequent tasks like NLP; and (2) needless words for NLP, e.g. section numbers and subsection numbers, can be removed. For achieving these objectives, the proposed method consists of the following four steps: (1) extract candidate elements of document structure by using rules; (2) build candidate trees from extracted elements; (3) prune trees and identify

unbranched trees; and (4) identify super-subrelations among unbranched trees. The proposed method has the following characteristics: (1) a rule-based approach does not need training data required in a machine learning approach; (2) building structural tree by extraction rules of numbered objects and grouping rules by left-hand side characters reduce candidate trees; and (3) pruning by sentence splitter can identify enumerations in one sentence.

In addition, an application to article extraction was demonstrated to discuss the effectiveness of the proposed method in legal documents. Since the proposed method achieved 100% accuracy for nine sample ordinances with 181 articles, it is accurate enough to be applied for practical use. Through generating a comparison table from extracted articles, it is confirmed that the proposed method is applicable for analyzing legal documents.

Reference

1. Song Mao, Azriel Rosenfeld, and Tapas Kanungo: Document Structure Analysis Algorithms: A Literature Survey. Proc. SPIE 5010, Document Recognition and Retrieval X, 11 pages (2003)
2. Isamu Iwai, Miwako Doi, Koji Yamaguchi, Mika Fukui, and Yoichi Takebayashi: A Document Layout System Using Automatic Document Architecture Extraction. CHI'89 Proceedings, pages 369-374, (1989)
3. Hervé Déjean: Numbered Sequence Detection in Documents. Proc. SPIE 7534, Document Recognition and Retrieval XVII, 753405 (18 January 2010), 12 pages (2010)
4. Hirokazu Igari, Akira Shimazu, and Koichiro Ochimizu: Document Structure Analysis with Syntactic Model and Parsers: Application to Legal Judgments, Jurisin 2011 (2011)
5. Tokuyasu Kakuta, Aki Shima, Daichi Saito, and Tadashi Ohtani: Development of eLen Database for Regulations of the Local Governments and a Quantitative Survey of the Regulations [in Japanese], Information network law review, volume 13, Number 1, Pages 13-33 (2014)
6. Yiming Yang and Xin Liu: A reexamination of text categorization methods, SIGIR-99, 8 pages (1999)
7. Salton, G., Wong, A., and Yang, C. S.: A Vector Space Model for Automatic Indexing. Commun. ACM, Vol. 18, Num. 11, pp. 613-620 (1975)
8. Sandeep Tata and Jignesh M. Patel: Estimating the selectivity of tf-idf based cosine similarity predicates, ACM SIGMOD Record, Volume 36, Issue 2, June 2007, Pages 7-12 (2007)

A Legal Citation Retrieval System Using Topic Modeling and Semantic Similarity

Talia Shwartz[1], Michael Berger[2], Juan Hernandez[3]

[1] LL.B., LL.M., LL.M. Candidate, University of California, Berkeley, School of Law
taliashwartz@berkeley.edu
[2] B.Sc., J.D., M.I.M.S., University of California, Berkeley, School of Information
mjb@ischool.berkeley.edu
[3] B.A., M.Sc. Candidate, Northwestern University, McCormick School of Engineering
juanhernandez2016@u.northwestern.edu

Abstract. Topic models are statistical models that detect themes in text corpora. They can be used in information retrieval to find documents that are "similar" to a query, based on the similarity of the themes in the query to the documents in the retrieval database. Applying such models to the domain of legal research might help in improving the efficacy and accuracy of legal research and writing processes that currently rely, to a large extent, on specialized domain knowledge to conduct human-supervised information organization, query formulation, and document retrieval. In this paper we suggest a novel approach that incorporates automatic content analysis methods into the legal sphere and applies Latent Dirichlet Allocation (LDA) to assist in case retrieval when drafting legal documents. We develop a prototype, built for a popular word processor, that runs on a fixed corpus of sixty-four thousand United States Supreme Court cases. The tool is called while the user is developing a document, using the document itself to formulate a query. The tool detects the user's writing context to automatically formulate a query, and uses topic modeling methods to recommend relevant legal cases for citation and quotation within that context. The paper offers an initial evaluation of the method by testing performance using paragraphs from existing legal cases and their associated citations, showing that our algorithm provides an overall effective recommender system compared with the traditional manual-human legal querying method.

Keywords: Text analysis; Topic modeling; Latent Dirichlet Allocation (LDA); Legal citation; Legal Corpus; United States Supreme Court; Legal research; Information Retrieval; Document Similarity;

1 INTRODUCTION

This paper reports on a novel implementation of a recommendation algorithm that utilizes LDA in the legal context to provide legal citation recommendations. Using natural language processing while benefiting from the multi-topic nature of LDA, the proposed tool is well-suited to legal research and is capable of augmenting the abilities of domain-knowledgeable users.

In common law, precedent-based legal systems, case law is a primary source of law. Legal practitioners, academics, and students all routinely conduct legal research and turn to case law in the pursuit of the most relevant set of facts and applicable legal rules in a court decision or other legal document. Once a relevant document is found, it can be cited in support of an argument, or distinguished as inapplicable. The legal retrieval system reported herein, was thus motivated by the desire to use text analysis to complement traditional legal research with an effective (accurate and relevant) and efficient (low time and effort) search tool. The extensive amount of digitized legal text available for search presents the problem of an ineffective legal search, a problem with which scholars have been trying to address for several decades[8]. Another issue that the presented tool is intended to address is the human "tendency toward the familiar", as was also reported in the legal context[25]. Indeed, the incorporation of text analysis methods into Law and Artificial Intelligence has been recognized as a pressing need[2]. A study conducted in 2007 on the American case law network found a highly skewed distribution of authorities in the Web of Law; 2% of U.S. Supreme Court citation network comprise 56% of all citations[27]. The ultimate meaning of this "rich get richer phenomenon", results in legal practitioners analyzing the legal sphere via a highly limited lens[9].

Previous attempts at using automated means for legal research have sometimes been met with difficulty due to technological limitations in both hardware and software. But with recent advances in both hardware capabilities and new algorithmic techniques, the technology is ripe for further exploration of legal search tools. Such tools have the additional potential of normalizing the distribution of case law usage and authority in the Web of Law. The need for an efficient legal citation tool is complemented and reinforced by the existence of advanced data mining and text analysis techniques as well as a more open, digitized environment from which a legal corpus of data can be drawn. The LDA algorithm is of particular interest as it can produce higher-dimension topics/clusters than other topic modeling and clustering algorithms that have been applied to this problem in the past.

To meet this end, a system was developed which recommends a case that is relevant to a text query that has been written in natural language. The described system runs on a data set of the United States Supreme Court decisions, which consists of 63,547 documents.

The rest of the paper unfolds as follows. The second section provides an overview of previous work utilizing text analysis techniques in various fields with regard to legal information extraction and retrieval systems and methods. Next, the methods used in our proposed model are described, namely, the application of topic modeling /

Latent Dirichlet Allocation to a legal corpus. The fourth part provides our results, including results from an *in vivo* automated testing method devised by the authors to test the practical performance of the tool built. In the last part, conclusions and further research are discussed.

2 PREVIOUS WORK

The use of probabilistic topic models to identify systematic patterns and commonalities in data has applications in diverse fields, from medicine[1] to political science [12,17]; and is implemented for such purposes as text mining[19], social network analysis[6], customer purchasing behavior[14], language modeling[3], and others. The application of collaborative topic modeling to a retrieval system for scientific articles[31] is a utilization of LDA that shares some features with the approach taken in this paper. The application of text analysis methods in the legal sphere is limited, albeit not novel. Automated content analysis and semantic interpretation of legal text has been implemented for the purposes of case summary[20,21] legal outcome prediction[2]; identification of justices' ideology[16] or quantifying the "complexity" of a written opinion[22]; determining consensus among justices[23]; attributing authorship in unsigned court opinions[18]; and as a litigation support system in the field of Online Dispute Resolution (ODR)[10].

The presented retrieval tool involves several lines of research within Law and AI, namely, knowledge representation, legal retrieval systems, and information extraction using various text analysis methods. The SALOMON system detects relevant legal text by word-frequency techniques and performs retrieval based on similarity measures, while acknowledging the potential of learning topics by the grouping of text [20,21]. Other models extract semantic information and classify legal text by predefined concepts[5], or suggest retrieval based on the k-Nearest Neighbors approach[2,24]. Natural Language Processing techniques have been also implemented in automatic classification of legal case factors[2,32] and in extracting ontologies from legal text[15]. A recent work by El Jelali et al. introduces an information retrieval system that uses machine learning and natural language processing techniques to match disputant case descriptions with court decisions[10]. Text mining and analysis have been implemented outside the academic realm as well, in aiding legal work from a practitioner's perspective. There are several designated Information Retrieval systems in the field of Online Dispute Resolution (ODR)[1].

As has been done in more recent work[10], the tool reported here uses natural language inputs rather than pre-defined, constructed, template-based methods, which were more common in older systems. Critically, the model presented here differs from previous work in the application's methods and goals. To the best of our knowledge, there is no previous published work that utilizes LDA in the context of the legal domain in order to build a legal document retrieval system. The advantage of

[1] For an overview see Carneiro D, Novais P, Andrade F, Zeleznikow J, Neves J.,Online dispute resolution: an artificial intelligence perspective. Artif Intell Rev 41(2):211–240 (2014).

LDA over many other methods that have been attempted before lies in its modeling of documents as distributed over *multiple* topics, rather than a single topic, as is assumed in hard clustering approaches such as K-means and K-medoids. In modeling legal decisions as topic distributions, and topics as distributions over words, LDA accounts for the multi-topic nature of a court case (a single case, or even a single part of a case, may naturally address several legal topics.) Another advantage of LDA is that it learns the relevant item representation in an automated unsupervised fashion, eliminating the need for manually curated topics or human-enabled feature extraction. Moreover, a significant portion of former work on legal information processing and retrieval focused on semantic inferences that could be gleaned from the legal text. While the suggested topic model algorithm does embody semantic similarity measures[26] rather than a word-based approach[10], in its simplicity, it overlooks the complexity of judicial language. In other words, the proposed tool uses text-analysis techniques that take into account similarity measures while being "agnostic" to words' linguistic meanings, differently from other work[8,10,15,19,32]. Another feature that distinguishes the work reported here is a practical testing technique, which automatically analyzes 1000 queries extracted from a United States Supreme Court Case database.

3 METHODS

Our aim was to design a tool for one type of use-case that is common among legal researchers—i.e. the task of finding a case to cite that is relevant to a portion of text that has already been written in a legal document, a common and modifiable use-case for this legal research tool.

The methods we employed are directed toward this use-case, and can be summarized as follows: first, we build a retrieval system using LDA that suggests legal citations that are most relevant to a natural language query from a user; and second, we test the retrieval system by simulating the expected use-case. In this model, "relevance" is defined by topic similarity: the most relevant document is the document that is most similar in topics to the text query. In the first phase, we build several different models by varying certain parameters used in LDA, described below.

In this section we outline the data used, the steps used for processing data, the method for approximating LDA, and our testing framework.

3.1 Latent Dirichlet Allocation

LDA is a hierarchical Bayesian generative model, in which documents are represented as a mixture of a limited set of topics, where each topic is characterized by a distribution over words[3]. The model, which allows the representation of documents in a reduced feature space consisting of K dimensions (rather than a larger space that contains as many dimensions as the number of unique words in a given corpus), is thus used as a method of unsupervised machine learning to discover latent topics that are hidden in a text.

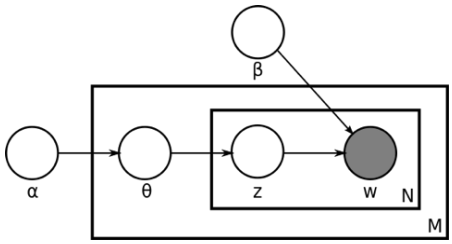


Fig. 1. Blei et al. 2003, page 997

LDA can be represented by plate notation to describe the generative process wherein each document in the corpus is generated by picking a multinomial topic distribution $\theta \sim \text{Dirichlet}(\alpha)$ for the document d. Each word of the document is assigned a topic from the topic distribution. Given the topic, a word is drawn from the multinomial word distribution $\beta k \sim \text{Dirichlet}(\eta)$ from the dictionary for that topic.

The method used in our model for approximating the LDA is the Online Learning algorithm, available in the Python library *Gensim*, described by Hoffman et al., a variation on the Expectation Maximization approach[13]. The Online Learning algorithm is streamed and runs in constant memory with regard to the number of documents and can make use of a distributed system, which allows it to be implemented on a much larger corpus.

3.2 Data

Data Source. Our data set is the entire corpus of United States Supreme Court decisions available on courtlistener.com,[2] which consists of 63,547 court decisions. The data span a period of over two centuries of deliberations by the United States Supreme Court, from 1754 to 2014. We created our models using a fifty percent random sample of the mentioned corpus, meaning, about 32,000 documents, to form a "modeling corpus." This modeling corpus was used due to the somewhat limited computational resources available to us – in this case, insufficient RAM. However, a 50% random sample is more than sufficient to form a model generalizable to the entire corpus, and in any event, the amount of RAM required to use the entire corpus would be well within the means of a commercial user, or could be parallelized across a cluster of inexpensive machines. Within the 32,000 document modeling corpus, we trained the model on 80% and evaluated perplexity on 20% (see section 3.3). Once a model was learned from the modeling corpus, topics were inferred on the entire cor-

[2] CourtListener, a legal research website sponsored by the non-profit Free Law Project, was established in 2010 by Brian W. Carver and Michael Lissner (University of California, Berkeley School of Information). Michael Lissner, CourtListener.com: A platform for researching and staying abreast of the latest in the law, Masters Thesis, University of California, Berkeley, School of Information, (May 7, 2010). For the Supreme Court database see https://www.courtlistener.com/?q=&court_scotus=on&order_by=score+desc (last visited April 16, 2015).

pus using the trained model, and these topic distributions were used to make the recommendations that we then tested, as described further below.

Data Preprocessing. As part of preprocessing, data cleaning for LDA consisted of removing punctuation and stop words, in which each document in the corpus was transformed into a "bag of words" representation. Moreover, we created two different corpora to test whether or not stemming helped to reduce noise and produce LDA models that represented topic distribution more accurately.

3.3 Creating and Evaluating Models

The modeling corpus, which consists of roughly 32,000 documents, is divided into a training set - consisting of 80% of the corpus selected randomly (approximately 26,000 documents) - and an evaluation set consisting of the remaining 20% of the corpus. To create each model, a set of K topics are first learned from the training set by applying Latent Dirichlet Allocation[3] to learn topics. Each model was then evaluated for perplexity based on the remaining evaluation set, selecting the model that minimizes perplexity as the best model, as described in section 4.2. The LDA algorithm uses two user-determined parameters, α and K, which alter the sparsity of the topic distribution and the number of topics, respectively. The best model for which we had the computational time and resources to implement on our single-processor machine was then chosen using the perplexity measurement described in the Analysis section below. The best model built on the modeling corpus is then applied among the entire corpus of documents to infer the topic distribution of each document therein, representing each document as a vector of length K. This model underlies the retrieval system - the topic vector of each document, as computed by the model, is stored in a database.

3.4 Retrieving Relevant Citations

In order to perform the information retrieval task, the engine receives a query consisting of a text input automatically extracted from the user's draft document. It then searches for relevant documents as follows: (1) the user's text is transformed into a K-dimensional vector representing that query's topic distribution using the same LDA model learned from the training corpus; (2) the similarity of the query topic vector to the topic vectors of each of the documents in the database is computed, and (3) the documents are retrieved as citation recommendations in descending order of their semantic similarity, as measured by a distance measure between the topic distributions.

We compare four different distance measures: Kullback-Leibler (KL) Divergence,

$$KL(p,q) = \sum_{i=1}^{T} p_i \log \frac{p_i}{q_i} \quad (1)$$

Information Radius (IR),

$$IR(p,q) = \sum_{i=1}^{T} p_i \log \frac{2\times p_i}{p_i+q_i} + \sum_{i=1}^{T} q_i \log \frac{2\times q_i}{p_i+q_i} \quad (2)$$

Hellinger Distance (HD)

$$HD(p,q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{T}(\sqrt{p_1}-\sqrt{q_i})^2} \quad (3)$$

and Manhattan Distance (MD).

$$MD(p,q) = 2\times(1 - \sum_{i=1}^{T} \min(p_i, q_i)) \quad (4)$$

In all the above formulae, p and q are the topic vectors to be compared, and p_i and q_i are each of the co-ordinate elements of the respective topic vectors. Each distance measure computes how different or similar two vectorised topic distributions are. The higher the distance, the more dissimilar the distributions. We tested more than one distance measure because KL, a very popular measure, is not symmetric, meaning that the similarity of document p to document q is not the same as the similarity of document q to document p. The three other approaches are symmetric.

These distance measures were used to create a document-to-document similarity coefficient following the example of Dagan et al. (1997) and Rus et. al (2013).

$$SIM(p,q) = 10^{-\delta IR(c,d)} \quad (5)$$

Another version of the retrieval algorithm weighs the similarity coefficient alongside a measure of the retrieved documents' popularity (i.e. the number of times each document has been cited before). The weighting is accomplished by multiplying the similarity by a coefficient of between 0.8 and 1.2, scaled to the total distribution of document citations. The coefficient boundaries were selected empirically to be conservative in order to prevent overweighting—*i.e.*, to prevent the popularity measure of a document from having an excessive effect on the ranking and overpowering the similarity measure.

3.5 Testing the Retrieval System

Besides the standard evaluation methods for topic models, we set up the following practical testing scenario as another method to determine the optimal parameters, as well as test the relevant performance of our two[3] corpora: We randomly selected 1000 paragraphs from cases in the US Supreme Court database, each of which contains some text and a citation to another Supreme Court decision. The citations were then separated from the paragraphs, and each of the paragraphs were used as query text to be input into the retrieval system. Using the above-described algorithm, we retrieve a sorted list of recommended citations for each test paragraph. Of course, we removed any citations that were not dated earlier than the document from which the test paragraph was taken. Our metric for performance was the position of the actual cited document in the recommendation ranking. Ideal performance would be when the number one recommendation in the ranking matches the actual citation that had been originally cited in the paragraph. We compared how well each of our models, similarity

[3] (1) Corpus with stemming, and (2) Corpus without stemming.

measures, and weighting with popularity ranks the citation found in the paragraph. The optimal combination of parameters will be used in the tool implementation.

4 RESULTS AND ANALYSIS

4.1 Model Parameter Choices

We developed two sets of LDA models with 15, 30, 50, 100, 200, 300, 400, 500, 600, 700, 800, and 900 topics, one set using stemming and another set without stemming words. In addition to comparing the number of topics, we tested the difference in results from selecting α as 50/k (a rule of thumb following Griffiths and Steyvers, 2004) and dynamically learning α from the data.

4.2 Model Evaluation

As described in the Methods section above, each model was evaluated in an initial evaluation stage. This was done by calculating the per-word likelihood and perplexity, using the 20% holdout of the modeling corpus as an evaluation set. Intuitively, perplexity is an indicator of the ability of a given model to predict the next word in a given document, and the lower the perplexity, the better the model is at doing so[30]. More formally, perplexity is derived from log-likelihood of unseen documents and is calculated as follows, where w is the set of unseen documents (and w_d refers to each document in the set), Φ is the topic matrix, and α is Dirichlet prior of the topic distribution:

$$\mathcal{L}(w) = \log p(w|\mathbf{\Phi}, \alpha) = \sum_d \log p(w_d|\mathbf{\Phi}, \alpha) \quad (6)$$

$$\text{perplexity}(\text{test set } w) = exp\left\{ {}^{\mathcal{L}(w)}/_{\text{count of tokens}} \right\} \quad (7)$$

The figure below shows the results of our evaluation of the perplexity of the models described above. Since automatic α parameter setting was computationally more expensive and yielded no relative benefit, we discontinued it. Figure 2 shows the perplexity of the different model on a log-scale, hence the negative values.

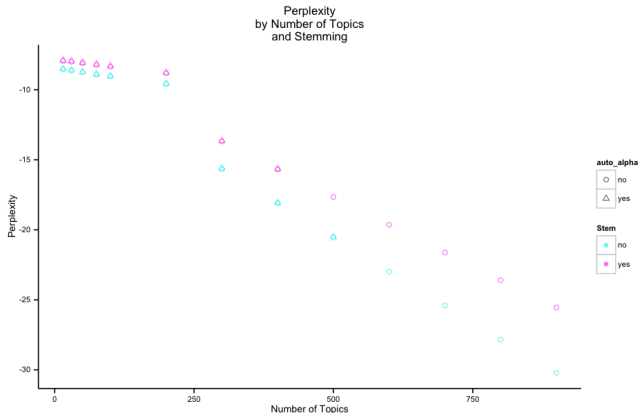


Fig. 2. Log-Perplexity by Number of Topics and Stemming

Although the apparently superior models contained over 500 topics, computing resource restrictions permitted us only to do the next phase of the testing with a 200-topic model by the publication deadline. The result is a less granular representation of the topic distribution of each document. With more topics, we would expect more accurate similarity metrics due to the higher resolution representation. Additional testing with the models of greater than 200 topics is currently being performed, with more results expected in the coming weeks.

4.3 Retrieval System Testing Results

Using the 200-topic model in the engine, we tested the recommendation performance of the engine using the procedure described in the Methods section above. This additional procedure measured the ability of the retrieval system to retrieve the citation that we knew was in some way relevant to a given text query paragraph (because it had been cited by the Supreme Court in relation to that paragraph). In the graph below, we show how the length of the query paragraph ("Document Length") affected the ranking of the "correct" citation returned by the retrieval system ("Minimum Ranking").

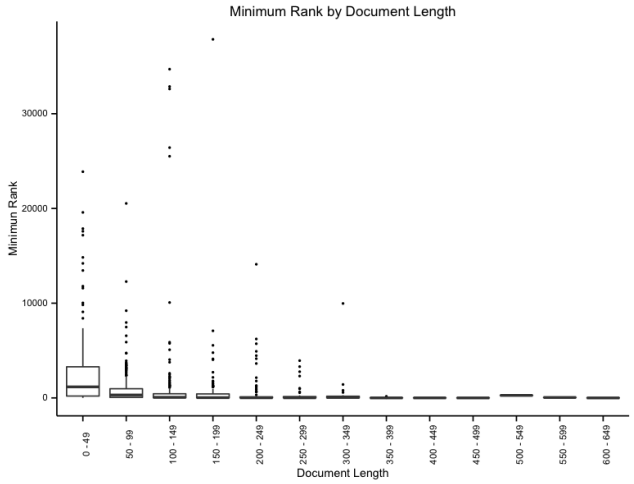


Fig. 3. Minimum Rank by Document Length

The graph above shows that performance stabilizes as the query length exceeds 250 words. These results have implications for recommendation tool design: using a 200-topic model, a recommendation tool built on this engine should extract 250 words of context from the user's document in order to retrieve better results.

We investigated the outlier points in the boxplot and found that they represented documents from the data set where the parser failed to parse a query paragraph correctly. Including query cases of every length, as well as outliers, our initial overall quantile results were as follows, where the top row shows the percentile and the bottom row shows the minimum ranking of the cited document:

Percentile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Minimum Rank	2	8	24	53.6	135.5	286.8	632.9	1223.8	2924.9	37854

Table 1. Initial Quantile Results

Focusing on query cases that were of length greater than 250 words, we report the quintile results below:

Percentile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Minimum Rank	0.6	2	3	6.4	13	27.2	59	174.6	703.2	9970

Table 2. Quantile Results for queries of length > 250 words

Thus, for queries of length greater than 250 words, the retrieval system built on the 200-topic model weighted with popularity returned the original citation associated with that query in its top 3 results 30% of the time, and in the top 59 results at least 70% of the time. These results are very encouraging for an initial test, and we are now

iterating on this design by engaging additional computing resources and parallelization techniques to test the models we created that had a larger number of topics, K.

4.4 Illustrative Query Results

To illustrate more intuitively how the retrieval system was able to find the matching document to many of the query paragraphs, we provide an example of both a matching and non-matching document to a single query.

The graph below shows the probability distribution of the top 20 topics in an example query (blue), and compares them with the distribution over the same 20 topics in the highest ranked document returned by the engine (pink). As can be seen, the topic distributions are fairly well matched:

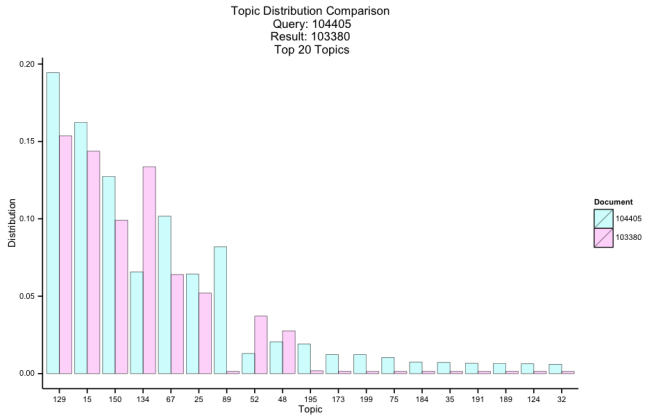


Fig. 4. Topic Distribution Comparison

Compare this with the next graph, which shows the same query compared with a low-ranked document returned from the engine. As can be seen, the top 20 topics in the distribution for the comparison document (pink) are barely found in the probability distribution for topics in the query (blue):

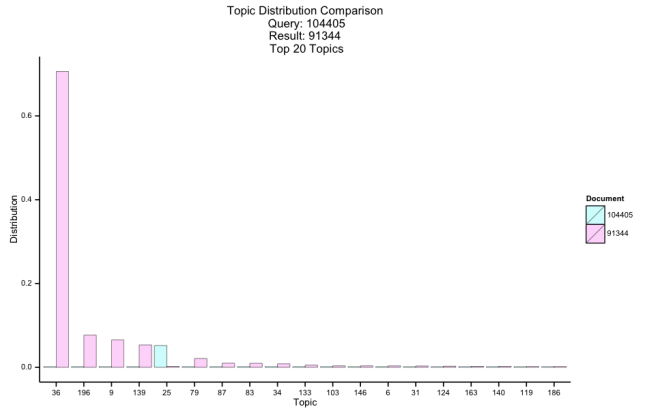


Fig. 5. Topic Distribution Comparison

5 CONCLUSIONS

To the best of our knowledge, this was the first attempt of its kind to create a reliable legal retrieval system based on LDA topic modeling, and the first attempt at testing such a system by automated means. As described above, initial results have been encouraging, and further research and testing is currently being undertaken.

The tool's admittedly circumscribed ability to detect the cited document among its top recommended results is nevertheless worthy of further exploration, given the limited amount of topics in the model we were able to test with our set of computing resources. Additional models with higher number of topics and lower perplexity are currently being tested to see if they improve performance on the automated test set. While results points to the need for more topics in the models, an optimal number of topics should be learned from additional tests.

Additional modifications to the data set in the pre-processing stage, as well as further changes to the ranking method, are also candidates to be considered for improving the system's overall performance. Moreover, for the purpose of this research, the data set was intentionally left unfiltered. Further research might consider filtering out some of the documents based on noise (documents with low to non-alphabetic content) and legal relevance (for example, the corpus consisted of a large number of decisions dismissing writs of certiorari with very little content.) Another route for further research and development could consider using existing documents and their citations to learn (using supervised machine learning) the optimal weights for popularity and other metadata available in our corpus to improve recommendations.

This work also raises additional interesting avenues of research exploration in the domain of testing. Whereas the automated testing techniques described rely on judicial language, additional testing could be performed to evaluate the tool's performance given text queries generated by laypeople, students, or legal practitioners with different linguistic styles. A user experience study could be conducted among legal practitioners to determine the tool's efficiency (low time and effort), by comparing the legal research user experience with the aid of the suggested tool versus traditional legal research and writing methods. Furthermore, testing is required among testers with domain specific knowledge to evaluate the effectiveness of the system, in terms of the recommendation accuracy and relevance. By collecting user data, this information could also be used to optimize the recommendation, using supervised learning, in a similar way to the calibration framework suggested above.

Yet another direction of future research could be the substitution of alternative and newer methods of topic modeling, such as Heirarchical Dirichlet Process (HDP) and Nested Chinese Restaurant Proces (nCRP)[4,28] in an attempt to contribute to the sophistication of the model and its accuracy, by implementing/learning topic hierarchies, rather than flat lists, within the legal texts.

Finally, from a normative and policy perspective, additional research should consider the implications of the discussed technology on the legal industry and profes-

sion, such as its potential narrowing effect on legal research and writing skills, or its potential in widening the distribution of case law networks.

Acknowledgments. The authors wish to thank Brian Carver and the Free Law Project for their generous provision of data. This resource was invaluable in conducting our analysis. We also acknowledge the UC Berkeley DLab, its director Justin McCrary, the Computational Text Analysis Working Group led by Nicholas Adams and Brooks Ambrose, for their excellent support and assistance. Finally, we thank the UC Berkeley School of Information and its Computing & Information Services Unit for their technical and application support, access, and hosting.

REFERENCES

1. Arnold, C. El-Saden, S. Bui, A. Taira, R. 2010. Clinical case-based retrieval using latent topic analysis. In: *Proc. AMIA* 26.
2. Ashley, K.D. Brüninghaus, S. 2009. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law* 17(2), 125.
3. Blei, D.M., Ng, A. and Jordan, M.I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research.* 3, 993.
4. Blei, D.M., Griffiths, T.L. and Jordan. M.I. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)* 57(2), 7.
5. Brüninghaus, S. Ashley, K.D. 2001. The role of information extraction for textual CBR. In: *Proceedings of the 4th international conference on case-based reasoning* (ICCBR-01, Vancouver, CA). Springer Lecture Notes in Artificial Intelligence, Springer, Berlin.
6. Cha, Y. and Cho, J. 2012. Social-network analysis using topic models. *SIGIR '12*.
7. Dagan, I. Lee, L. and Pereira, F. Similarity-based methods for word sense disambiguation. 1997. *Proceedings of the 35th ACL/8th EACL*, 56.
8. Daniels, J.J. Rissland, E.L. 1997. Finding legally relevant passages in case opinions. In: *Proceedings of the 6th International Conference on Artificial intelligence and Law*, ACM. New York, NY, USA 39.
9. Delgado R. and Stefancic, J. 2007. Why Do We Ask the Same Questions? The Triple Helix Dilemma Revisited, 99 *LAW LIBR. J*. 307.
10. El Jelali, S. Fersini, E. and Messina, E. 2015. Legal retrieval as support to eMediation: matching disputant's case and court decisions. *Artificial Intelligence and Law* 23(1), 1.
11. Griffiths, TL. Steyvers, M. 2004. Finding Scientific Topics. In Proceedings of the National Academy of Sciences of the United States of America, 101, 5228.
12. Grimmer, J. 2010. A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases. *Political Analysis* 18(1), 1.
13. Hoffman, M. Blei, D. and Bach, F. 2010. Online learning for Latent Dirichlet Allocation. *Neural Information Processing Systems*.
14. Iwata, T. Watanabe, S. Yamada and T. Ueda, N. 2009. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*.
15. Lame, G. 2004. Using NLP techniques to identify legal ontology components: Concepts and relations. *Artificial Intelligence and Law* 12(4), 379.
16. Lauderdale, B. and Clark, T. 2012. The Supreme Court's Many Median Justices. *The American Political Science Review* 106(4), 847.

17. Laver, M. Benoit, K. and Garry, J. 2003. Extracting Policy Positions from Political Texts Using Words as Data. *The American Political Science Review* 97 (2), 311.
18. Li, W. Azar, P. Larochelle, D. Hill, P. Cox, J. Berwick, R. C. and Lo, A. W. 2013. Using algorithmic attribution techniques to determine authorship in unsigned judicial opinions. 16 *Stanford Technology Law Review,* 503.
19. McCarty, L.T. 2007. Deep semantic interpretations of legal texts. In Proceedings of the eleventh international conference on artificial intelligence and law, 217.
20. Moens, M.F. Gebruers, R. and Uyttendaele, C. 1996. SALOMON: Final Report. Technical Report ICRI, K.U. Leuven.
21. Moens, M.F., Uyttendaele, C. and Dumortier, J. 1999. Abstracting of Legal Cases: The Potential of Clustering Based on the Selection of Representative Objects. *Journal of the American Society for Information Science* 50(2), 151.
22. Owens, R. and Wedeking, J. 2011. Justices and Legal Clarity: Analyzing the Complexity of U.S. Supreme Court Opinions. *Law & Society Review* 45(4), 1027.
23. Rice, D. and Zorn, C. J. 2014. The Evolution of Consensus in the U.S. Supreme Court. Available at SSRN: http://ssrn.com/abstract=2470029.
24. Riloff, E. 1996. An Empirical Study for Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence* 85, 101.
25. Ruhl, J. B. and Katz, D. M. 2015. Measuring, Monitoring, and Managing Legal Complexity. Iowa Law Review, 100, Vanderbilt Public Law Research Paper No. 15-1.
26. Rus, V., Niraula, N. and Banjade, R. 2013. Similarity Measures based on Latent Dirichlet Allocation. In: *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing 2013). Springer Berlin Heidelberg, Samos, Greece, 459.
27. Smith, T. A. 2007. The Web of the Law, *44 San Diego Law Review*, 309.
28. Teh,Y. W. Jordan, M. I. Beal, M. J. and Blei, D. M. 2006. *Journal of the American Statistical Association*, 101 (476), 1566.
29. Wallach, H. M. 2006. Topic modeling: beyond bag-ofwords. *ICML '06*. ACM.
30. Wallach, H.M. Murray, I. Salakhutdinov, R. and Mimno. D. 2009. Evaluation methods for topic models. In L. Bottou and M. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*.
31. Wang, C. and Blei, D.M., 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 448.
32. Wyner, A. and Peters, W. 2010. Lexical semantics and expert legal knowledge towards the identification of legal case factors. In Radboud Winkels, editor, *Proceedings of Legal Knowledge and Information Systems (JURIX 2010)*. IOS Press. 127.

An Implementation of Belief Re-revision and Reliability Change in Legal Case

Pimolluck Jirakunkanok, Katsuhiko Sano, and Satoshi Tojo

School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{pimolluck.jira,v-sano,tojo}@jaist.ac.jp

Abstract. Since belief change plays a significant role in agent communication, we need to consider two questions. The first question is how an agent decides to change his/her belief. If we consider the reliability of an information source, an agent may change his/her belief based on reliability change. For reliability change, an agent may change his/her reliability for the other agents based on his/her belief change. Both belief change and reliability change can be formalized by several dynamic operators in terms of dynamic epistemic logic. Since there are many operators, the second question is how an agent decides which operators to be applied for realizing his/her belief change and reliability change. In order to deal with these questions, we develop a practical implementation for analyzing belief change and reliability change of an agent. Moreover, we also demonstrate our implementation by a legal case.

Keywords: belief revision, reliability change, legal case, dynamic epistemic logic

1 Introduction

In agent communication, belief revision is a key problem, i.e., when an agent receives a piece of information, he/she has to decide if he/she will believe the received information or not. That is, whenever the agent considers that the received information entails inconsistencies in his/her belief, he/she has to revise his/her belief. On the other hand, the agent will accept and may believe the received information if he/she considers that it is consistent with his/her belief. Moreover, the reliability of the information source, which is one of the most important criteria in agent communication, can be used to decide which information the agent should believe. That is, an agent will believe the received information if a source of received information is considered to be reliable. Recently, belief revision have been extensively studied by some researchers [1, 2]. In addition, dynamic epistemic logic (DEL) [3] is a branch of modal logic for studying about knowledge change or belief revision.

In legal proceedings, a judge also needs a concept of belief revision. That is, when the judge receives new information, he/she has to reconsider his/her belief

in order to reach his/her own decision. A legal case is a good example to show how a judge changes his/her belief by the reasoning process. Nowadays, several studies [4, 5] have applied logical approaches in the legal systems because logic can be used to clarify the meaning and soundness of reasoning.

Thus far, the previous work [6] introduced two dynamic operators including commitment and permission in order to formalize belief re-revision of a judge in a legal case. An agent can remove some belief states by applying the commitment operator, while the permission operator is used for restoring the former belief. Then, [7] presented an influence of the reliability of information source on belief change, i.e., an agent may change his/her belief based on a consideration of the reliability. In order to formalize such reliability change, two dynamic operators consisting of upgrade and downgrade operators are introduced. Next, [8] proposed a logical formalization based on a combination of dynamic operators from [6] and [7] for analyzing belief re-revision based on reliability change of a judge in legal case. In addition, their work introduced a new kind of downgrading called joint downgrade operator in order to downgrade such agents in the specific group to be equally reliable and less reliable than the other agents. Nevertheless, when we have many dynamic operators, the question is how we decide which operators to be applied for analyzing his/her belief change and reliability change.

This study aims to develop an implementation of our logical formalization in a computer system in order to analyze belief re-revision and reliability change of an agent in a practical way. Furthermore, our implementation is demonstrated in an example of legal case from British Columbia.

This paper is organized as follows. Section 2 describes the target legal case. Then, Section 3 presents the static logic of agents' beliefs equipped with the notion of sign information. In order to capture belief change and reliability change of an agent, Section 4 provides dynamic operators in terms of dynamic epistemic logic. In Section 5, we propose an implementation of our logical formalization. Finally, Section 6 concludes this paper and states some future works.

2 Target Legal Case

Let us summarize a story of our target legal case that occurred on August 31, 1996 in Surrey, British Columbia[1] as follows:

> During some Indo-Canadian games at Royal Kwantlen Park in Surrey, a victim v with a collection of new friends including f_1, f_2 and the others gathered for social purposes, possibly including alcoholic beverages. At around 5:30 p.m., a van consisting of two respondents r_1, r_2 and the others slowly approached the group of v. As the van drew closer, a burst of gun fire swept the group of v, shooting on a low trajectory into the ground. As a result, v died and three others including $f2$ were wounded. Then, the van sped away.

[1] This legal case can be referred from `http://www.canlii.org/en/`.

In the inquiry stage, several witnesses who involved in the incident were interviewed by the police. However, there are only two witnesses, i.e., f_1 and f_2 gave the reasonable evidences as follows:

(I_1) f_1 stated that r_1 was a driver of the van and r_2 was the shooter. More details can be shown as follows:

> f_1 said that he saw a brown van coming from the north toward us. Then, he saw that r_2 lifted the barrel of the gun and then started shooting.

(I_2) f_2 only stated that r_1 was a driver of the van, but did not identify r_2 as the shooter. More details can be shown as follows:

> f_2 was asked: "Who was in the driver's seat? Can you describe him?" His answer was: "r_1 was driving. I'm one hundred percent sure. There was no passenger." "Who was in the back of the van?" His answer was: "I don't know."

After the interviews, both r_1 and r_2 were charged with the first degree murder of v, the attempted murder of three other persons, and the aggravated assault on the same three persons.

In the Crown Court, f_2 changed his statement that identified who is the shooter. Thus, there are two testimonies as follows:

(T_1) f_1 stated that r_1 was a driver of the van and r_2 was the shooter. More details can be shown as follows:

> f_1 testified that he saw r_1 driving the vehicle and saw r_2 sitting by the sliding door. He said he saw r_2 struggling with the gun. He also testified that he has known r_1 for 10 to 15 years and r_2 for 15 years.

(T_2) f_2 also stated that r_1 was a driver of the van and r_2 was the shooter. More details can be shown as follows:

> f_2 testified that he saw r_2 with the gun, firing. He was hit in the legs. He testified he could identify r_1. He had known him for the past 10 to 12 years. He recalled the incident of one week earlier at the Rotary Stadium that he has an altercation with r_2. He said he knew the van was r_1's van because he had seen him driving it before and he described r_2 as being in the middle seat. He also identified that he has known r_2 for the past 10 or 12 years.

From the above testimonies, the judge accepted that the statements of both f_1 and f_2 as sufficient identification evidence because of the following reasons.

- With respect to the statements of f_1 (I_1 and T_1), the judge considered that they was consistent and could be accepted as both credible and reliable. The judge was satisfied beyond a reasonable doubt that f_1 has properly identified both r_2 and r_1 as being the shooter and the driver involved in this crime.
- With respect to the statements of f_2, the judge may have some doubt about the identification by f_2 of r_2. Thus, the judge considered that only f_2's identification of r_1 as the driver to be truthful.

– Both f_1 and f_2 had an absolutely clear view on a clear day and, even in a brief span of time with a van moving very slowly past them, they were both very familiar with both of these defendants. Therefore, the judge considered that their identification is the truth.

In this verdict, the judge acquitted both r_1 and r_2 of first degree murder, but convicted them of manslaughter. The judge also convicted them of aggravated assault instead of attempted murder of the other three victims. This is because of the following reasons.

– Since the shooting was the random nature of the attack, there was no intent to cause death.
– Due to the fact that the shooting was directed downwards, rather than at the group generally, which would inevitably have caused much greater damage. This shows that r_2 did not intent to kill v.

For this reason, the Crown Court judged both r_1 and r_2 to be each sentenced to nine years in prison on the manslaughter count, and to terms of four years on each of the other three convictions, all to be served concurrently with the nine year sentence for manslaughter by Article 236 of Criminal Code.[2]

> *Article 236* (Manslaughter)
> Every person who commits manslaughter is guilty of an indictable offence and liable
> (1) where a firearm is used in the commission of the offence, to imprisonment for life and to a minimum punishment of imprisonment for a term of four years; and
> (2) in any other case, to imprisonment for life.

3 Static Logic of Agents' Beliefs for Signed Information

To formalize belief re-revision and reliability change of an agent from a logical point of view, we introduce a modal language, based on previous works [6, 7], which enables us to formalize each agent's belief, the reliability of information sources, and signed information.

Let G be a fixed *finite* set of agents. Our syntax $\mathcal{L}$ consists of the following vocabulary: (i) a countably infinite set $\mathsf{Prop} = \{ p, q, r, ... \}$ of propositional letters, (ii) Boolean connectives: $\neg$, $\wedge$, (iii) the belief operators $\mathsf{Bel}(a, \cdot)$ $(a \in G)$, (iv) the signature operators $\mathsf{Sign}(a, \cdot)$ $(a \in G)$, and (v) the constants for reliability ordering $b \leqslant_a c$ $(a, b, c \in G)$. A set of formulas of $\mathcal{L}$ is inductively defined as follows:

$$\varphi ::= p \,|\, \neg\varphi \,|\, \varphi \wedge \varphi \,|\, \mathsf{Bel}(a, \varphi) \,|\, \mathsf{Sign}(a, \varphi) \,|\, b \leqslant_a c,$$

where $p \in \mathsf{Prop}$ and $a, b, c \in G$. Note that $b <_a c$ stands for b is strictly more reliable than c, i.e., $(b \leqslant_a c) \wedge \neg(c \leqslant_a b)$, and $b \approx_a c$ which stands for b and c are

[2] An English translation of the article can be referred from `http://www.canlii.org/en/ca/laws/`.

equally reliable, defined as $(b \leqslant_a c) \wedge (c \leqslant_a b)$. We define $\vee$, $\rightarrow$, $\leftrightarrow$ as ordinary abbreviations. Next, let us provide Kripke semantics for our syntax. A *model* $\mathfrak{M}$ is a tuple $\mathfrak{M} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq_a)_{a\in G}, V)$, where W is a non-empty set of states, called the *domain*, $R_a \subseteq W \times W$ is an accessibility relation representing beliefs, $S_a \subseteq W \times W$ is an accessibility relation representing signatures, $\preccurlyeq_a$ is a function which maps from W to $\mathcal{P}(G \times G)$ representing agent a's reliability ordering between agents, and $V : \mathsf{Prop} \rightarrow \mathcal{P}(W)$ is a valuation. In what follows, we simply write $b \preccurlyeq_a^w c$ for $(b, c) \in \preccurlyeq_a (w)$. Following [9] and [7], $\preccurlyeq_a^w$ is always required to be a *total pre-ordering* between agents, i.e., $\preccurlyeq_a^w$ is reflexive ($b \preccurlyeq_a^w b$ for all b), transitive, and comparable (for any b and c, $b \preccurlyeq_a^w c$ or $c \preccurlyeq_a^w b$). For any binary relation X on W and any state $w \in W$, we write $X(w)$ to mean $\{ v \in W \mid (w, v) \in X \}$.

Given any model $\mathfrak{M}$, any state $w \in W$, and any formula φ, we define the *satisfaction relation* $\mathfrak{M}, w \models \varphi$ inductively as follows:

$$
\begin{array}{lll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p) \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \mathfrak{M}, w \not\models \varphi \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w \models b \leqslant_a c & \text{iff} & b \preccurlyeq_a^w c \\
\mathfrak{M}, w \models \mathsf{Sign}(a, \varphi) & \text{iff} & \mathfrak{M}, v \models \varphi \text{ for all } v \text{ such that } wS_av \\
\mathfrak{M}, w \models \mathsf{Bel}(a, \varphi) & \text{iff} & \mathfrak{M}, v \models \varphi \text{ for all } v \text{ such that } wR_av
\end{array}
$$

A formula φ is *valid* in a model $\mathfrak{M}$ if $\mathfrak{M}, w \models \varphi$ for all states w of $\mathfrak{M}$.

By using the total pre-ordering $\preccurlyeq_a^w$, the agents can be ranked by giving a partition $(\mathsf{C}_i^a)_{i\leq M}$ to G, where M is a natural number representing the maximum rank (such M always exists because G is finite). $c \in \mathsf{C}_i^a$ can be read as 'from a's viewpoint, the rank of c is i'. As a result, the agents who are equally reliable are categorized in the same group.

4 Dynamic Operators for Belief Re-revision and Reliability Change of An Agent

This section describes the dynamic operators for formalizing belief re-revision and reliability change. First, we describe a joint downgrade based on [8] for formalizing reliability change. Then, we employ a variant of the permission based on [6], and the private announcement from [7] for formalizing belief re-revision.

4.1 Joint Downgrade

Based on the previous work [8], a joint downgrade $[H \downdownarrows^a]$ is introduced in order to allow an agent to downgrade the agents in the specific group to be equally reliable and less reliable than the other agents. For $[H \downdownarrows^a]$, $H \subseteq G$ is a set of agents. The reading of $[H \downdownarrows^a]\psi$ is 'after such agents in H are downgraded jointly by the agent a, ψ holds'. Semantically speaking, $[H \downdownarrows^a]$ makes such agents in H to be equally reliable and less reliable than all the other agents. Note that the joint downgrade operator $[H \downdownarrows^a]$ is different from the downgrade

operator $[H \Downarrow^a_\varphi]$ in [7] as follows. After downgrading, $[H \downarrow\downarrow^a]$ makes the reliability ordering between agents in H to be equal, while $[H \Downarrow^a_\varphi]$ keeps the same reliability ordering between agents in H.

Next, let us fix a Kripke model $\mathfrak{M} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq_d)_{d\in G}, V)$, a semantic clause for $[H \downarrow\downarrow^a]$ on $\mathfrak{M}$ and $w \in W$ is defined by:

$$\mathfrak{M}, w \models [H \downarrow\downarrow^a]\psi \text{ iff } \mathfrak{M}^{H\downarrow\downarrow^a}, w \models \psi,$$

where $\mathfrak{M}^{H\downarrow\downarrow^a} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq'_d)_{d\in G}, V)$ and $\preccurlyeq'_d$ is defined as: for all $u \in W$:

- if $d \neq a$, we put $\preccurlyeq'^u_d = \preccurlyeq^u_d$.
- otherwise (if $d = a$), we define $b \preccurlyeq'^u_a c$ iff

$$\big(b, c \in H\big) \text{ or } \big(b, c \in (G \setminus H) \text{ and } b \preccurlyeq^u_a c\big) \text{ or } \big(b \in (G \setminus H) \text{ and } c \in H\big).$$

Note that this joint downgrade can preserve the property of total pre-ordering of $(\preccurlyeq_d)_{d\in G}$. The following are valid on all models.

$[H \downarrow\downarrow^a]p$	$\leftrightarrow p$	
$[H \downarrow\downarrow^a](b \leqslant_d c)$	$\leftrightarrow b \leqslant_d c$	$(d \neq a)$
$[H \downarrow\downarrow^a](b \leqslant_a c)$	$\leftrightarrow b \leqslant_a c$	$(b, c \in G \setminus H)$
$[H \downarrow\downarrow^a](b \leqslant_a c)$	$\leftrightarrow \top$	$(b, c \in H)$
$[H \downarrow\downarrow^a](b \leqslant_a c)$	$\leftrightarrow \bot$	$(b \in H, c \in G \setminus H)$
$[H \downarrow\downarrow^a](b \leqslant_a c)$	$\leftrightarrow \top$	$(c \in H, b \in G \setminus H)$
$[H \downarrow\downarrow^a]\neg\psi$	$\leftrightarrow \neg[H \downarrow\downarrow^a]\psi$	
$[H \downarrow\downarrow^a](\psi_1 \wedge \psi_2)$	$\leftrightarrow [H \downarrow\downarrow^a]\psi_1 \wedge [H \downarrow\downarrow^a]\psi_2$	
$[H \downarrow\downarrow^a]\mathsf{Sign}(b, \psi)$	$\leftrightarrow \mathsf{Sign}(b, [H \downarrow\downarrow^a]\psi)$	
$[H \downarrow\downarrow^a]\mathsf{Bel}(b, \psi)$	$\leftrightarrow \mathsf{Bel}(b, [H \downarrow\downarrow^a]\psi)$	

Moreover, if ψ is valid on all models, then $[H \downarrow\downarrow^a]\psi$ is also valid on all models.

4.2 Permission

The permission $[\mathsf{per}_j(\varphi)]$ is proposed in [6] in order to capture belief re-revision of an agent. This permission operator $[\mathsf{per}_j(\varphi)]$ allows an agent to restore the former possibilities to his/her belief. Semantically speaking, $[\mathsf{per}_j(\varphi)]$ is used for enlarging j's attention to cover all the φ's states. While the previous study [6] used the permission to revise agent's belief locally in one specified world, this work assumes that an effect of an agent's permission is applied globally for all $w \in W$. Given a Kripke model $\mathfrak{M} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq_a)_{a\in G}, V)$, a semantic clause for $[\mathsf{per}_j(\varphi)]\psi$ on $\mathfrak{M}$ and $w \in W$ is defined as follows:

$$\mathfrak{M}, w \models [\mathsf{per}_j(\varphi)]\psi \text{ iff } \mathfrak{M}^{\mathsf{per}_j(\varphi)}, w \models \psi,$$

where $\mathfrak{M}^{\mathsf{per}_j(\varphi)} = (W, (R'_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq_a)_{a\in G}, V)$ and $(R'_a)_{a\in G}$ is defined as:

- if $a = j$, R'_a is defined as follows: for all $x \in W$,

$$R'_a(x) := R_a(x) \cup [\![\varphi]\!]_{\mathfrak{M}}, \text{ where } [\![\varphi]\!]_{\mathfrak{M}} := \{v \in W \mid \mathfrak{M}, v \models \varphi\}.$$

– otherwise (if $a \neq j$), $R'_a := R_a$.

Note that this permission operator can preserve the properties of seriality, transitivity, and Euclideanness of S_a.[3] For R_a, the permission operator can preserve the property of seriality, but it cannot preserve the properties of transitivity and Euclideanness in general.

4.3 Private Announcement

From [7], the private announcement $[\varphi \rightsquigarrow a]$ is captured in terms of the *action model* in dynamic epistemic logic. Our intended reading of $[\varphi \rightsquigarrow a]$ is "after a private announcement of φ to agent a". The main idea is that belief change of a will not be noticed by the other agents than a,[4] and only a recipient is defined as agent a. Therefore, this operator can be used for self-decision. Firstly, $[\varphi \rightsquigarrow a]$ is defined by an action model, i.e., a tuple $(E, (D_c)_{c\in G}, (U_a)_{a\in G}, \mathrm{pre})$ such that E consists of two actions: φ-announcing action $!_\varphi$ to agent a and non-announcing action $\top$, $D_a = \{ (!_\varphi, !_\varphi), (\top, \top) \}$ and $D_c = \{ (!_\varphi, \top), (\top, \top) \}$ if $c \neq a$, $U_c = \{ (!_\varphi, \top), (\top, \top) \}$ for all $c \in G$, and pre assigns a precondition to each action by $\mathrm{pre}(!_\varphi) = \varphi$ and $\mathrm{pre}(\top) = \top$. Next, when a Kripke model $\mathfrak{M}$ is fixed as a tuple $(W, (R_c)_{c\in G}, (S_c)_{c\in G}, (\preccurlyeq_c)_{c\in G}, V)$, a semantic clause for $[\varphi \rightsquigarrow a]\psi$ on $\mathfrak{M}$ and $w \in W$ is defined as follows:

$$\mathfrak{M}, w \models [\varphi \rightsquigarrow a]\psi \;\text{ iff }\; \mathfrak{M}^{\varphi\rightsquigarrow a}, (w, !_\varphi) \models \psi,$$

where $(w, !_\varphi)$ is the updated state of $\mathfrak{M}^{\varphi\rightsquigarrow a}$ (defined just below) by the above action model and $\mathfrak{M}^{\varphi\rightsquigarrow a} = (W', (R'_c)_{c\in G}, (S'_c)_{c\in G}, (\preccurlyeq'_c)_{c\in G}, V')$ is the updated model by the above action model, i.e.,

– $W' := W \times E = W \times \{ !_\varphi, \top \}$.
– $(w, e)R'_c(v, f)$ iff wR_cv and $(e, f) \in D_c$ and $\mathfrak{M}, v \models \mathrm{pre}(f)$ (for all $c \in G$).
– $(w, e)S'_c(v, f)$ iff wS_cv and $(e, f) \in U_c$ (for all $c \in G$).
– $d \preccurlyeq'^{(w,e)}_c d'$ iff $d \preccurlyeq^w_c d'$.
– $(w, e) \in V'(p)$ iff $w \in V(p)$.

Note that, based on [7], this private announcement can preserve the properties of seriality, transitivity, and Euclideanness of S_c. For R_c, the properties of transitivity and Euclideanness are preserved by this private announcement.

In addition, [7] proposed to capture the *careful policy* from [9] based on this private announcement as follows: Based on [9], a careful policy which is one of aggregation policies is used to decide which pieces of information an agent should believe. A concept is to accept, as beliefs, the statements which are universally signed by a group of agents who are equally reliable. Firstly, the careful policy is

[3] For a binary relation S, S is *serial* if for any state w, there is some state v such that wSv. S is *transitive* if wSv and vSu jointly imply wSu for all states w, v, u, and S is *Euclidean* if wSv and wSu jointly imply vSu for all states w, v, u.

[4] Based on [7], the axiom $[\varphi \rightsquigarrow a]\mathsf{Bel}(c, \psi) \leftrightarrow \mathsf{Bel}(c, \psi)$ captures that the action of a's privately receiving message φ will not affect of the other agents' beliefs than a. Thus, this work considers only the case that the other agents than a do not know about such event.

defined in terms of dynamic operators by $[\mathsf{Careful}(a,\varphi)]$, whose reading is "agent a aggregates information about φ". Then, $\mathsf{Sign}(\mathsf{C}_i^a,\varphi)$ which stands for 'all agents who are in the set C_i^a sign statement φ' is defined as follows:

$$\mathsf{Sign}(\mathsf{C}_i^a,\varphi) := \bigwedge\nolimits_{c\in\mathsf{C}_i^a}\big(\mathsf{Sign}(c,\varphi)\big).$$

Next, the *UniSign*, whose reading is 'a believes that φ is universally signed by a group of agents who are equally reliable', is introduced as follows:

$$\mathsf{UniSign}(\varphi,a) := \bigvee\nolimits_{i\leq M}\left(\begin{array}{c}\mathsf{Bel}\big(a,\mathsf{Sign}(\mathsf{C}_i^a,\varphi)\big)\wedge\\ \mathsf{Bel}\big(a,\bigwedge_{1\leq j\leq i-1}\neg\mathsf{Sign}(\mathsf{C}_j^a,\neg\varphi)\big)\end{array}\right),$$

where M is the maximum natural number of $\{i\leq \#G \mid \mathsf{C}_i^a\neq\emptyset\}$. Finally, the careful policy is captured by the help of the private announcement as follows:

$$[\mathsf{Careful}(a,\varphi)]\psi := \mathsf{UniSign}(\varphi,a)\rightarrow[\varphi\rightsquigarrow a]\psi,$$

where $[\mathsf{Careful}(a,\varphi)]\psi$ can be read as 'after agent a aggregates information about φ by the careful policy, ψ holds.'

5 Implementation

This section introduces an implementation of our logical formalization. We have implemented a Windows-Based Application with Visual C# ^{TM} as shown in Fig. 1. This program outputs the truth value of propositions, together with world accessibility relations in `dot` format. Thus, we can visualize the `dot` file by Graphviz^{TM}.

The main features of our implementation are summarized as follows:

(1) We can edit the definition of an initial model including G which is a set of agents, W which is a non-empty set of states, $(R_a)_{a\in G}$ which is an accessibility relation representing beliefs, $(S_a)_{a\in G}$ which is an accessibility relation representing signatures, and V which is a valuation. For the reliability ordering, the system will automatically define that all witness are equally reliable, i.e., their rank is 1 at the initial stage. Note that this work fixes the rank of reliability to be 0, 1, 2. That is, we may regard that 0 represents unreliable, 1 represents neutral, and 2 represents reliable. [5]

(2) We can input any dynamic operators, i.e., the private announcement $[p\rightsquigarrow a]$ which is represented by `[p=>a]`, the permission operator $[\mathsf{per}_j(p)]$ which is represented by `[per(j,p)]`, and the joint downgrade $[H\downdownarrows^a]$ which is represented by `[downgrade(a,H)]`. Then, the system automatically calculates such operator and outputs the result on the screen such as in Fig. 4.

[5] In the real world, a judge cannot categorize the reliability of witnesses to be several group as [9]. Thus, this work proposes to simplify the rank of reliability in a real situation by fixing to be 0, 1, and 2.

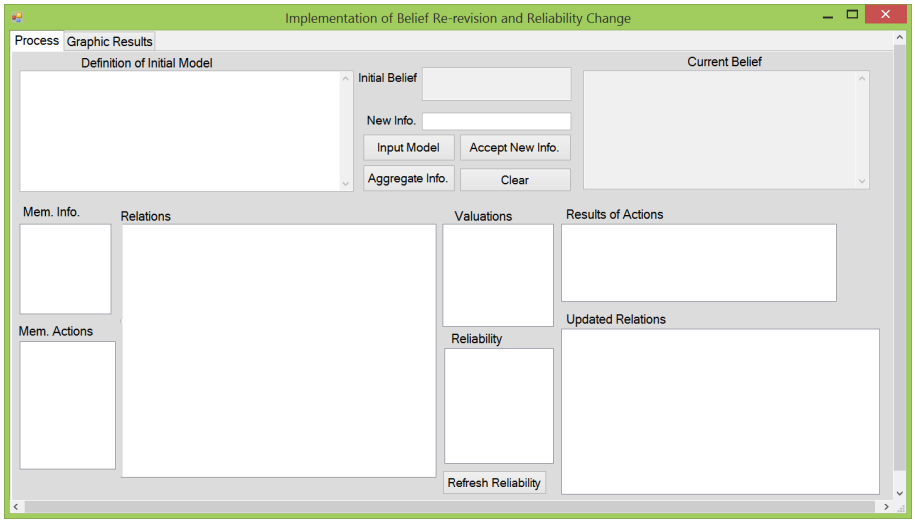


Fig. 1. Interface of our implementation

(3) The system can check if there is inconsistency between the existing belief and new information or not. If there is inconsistency, the system applies the joint downgrade and the permission operators. Then, the system calculates the operations of such operators. Finally, the system can automatically visualize the resultant states such as in Fig. 3.

(4) We can apply the careful policy by clicking the button "*Aggregate Info.*". Then, the system will calculate the careful policy according to Section 4.3.

Let us demonstrate our implementation by the target legal case in Section 2. In order to analyze the target legal case from the judge's perspective, we will only focus on the Crown Court. We will not consider the inquiry stage because there is no change of reliability.

In the Crown Court, the set G of agents is $\{j, f_1, f_2\}$, where we recall that f_1, f_2 are agents of two witnesses, and j is a judge of the Crown Court. For the statement involving the legal case, we consider only one propositional letter p whose reading is "r_2 is the shooter". We assume at first that all witnesses are equally reliable for j as follows:

$$\mathfrak{M}, w_0 \models \mathsf{Bel}(j,\ f_1 \approx_j f_2),$$

where we regard the state w_0 as j's viewpoint at the initial stage of the Crown Court. In the trial, we assume that f_1 first told statements to j that can be captured by $[\mathsf{Sign}(f_1, p) \rightsquigarrow j]$ and j will believe the following information.

$$\mathfrak{M}, w_0 \models [\mathsf{Sign}(f_1, p) \rightsquigarrow j]\mathsf{Bel}(j,\ \mathsf{Sign}(f_1, p))$$

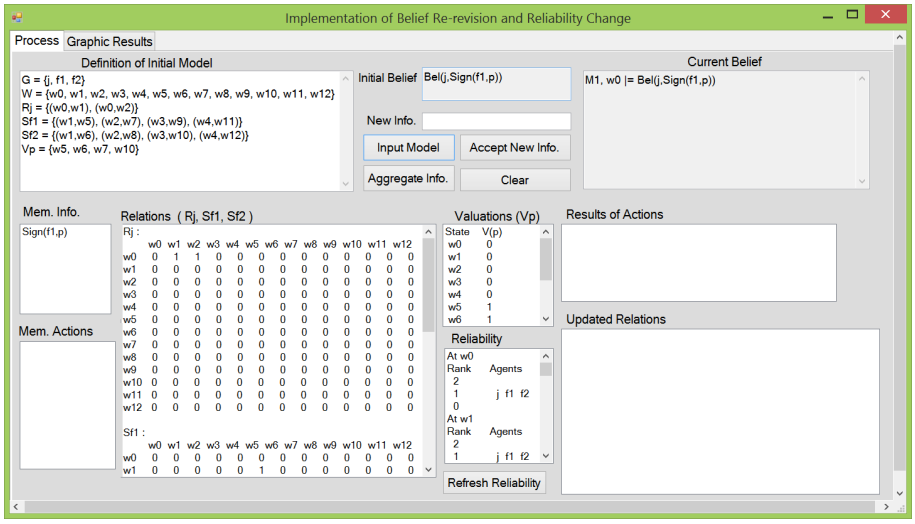


Fig. 2. Program after inputting a model $\mathfrak{M}_1$

We define $\mathfrak{M}_1$ by the updated model of $\mathfrak{M}$ after the above action. A model $\mathfrak{M}_1$ can be defined as follows:

$$
\begin{aligned}
G &= \{\, j, f_1, f_2 \,\} \\
W &= \{\, w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}, w_{11}, w_{12} \,\} \\
R_j &= \{\, (w_0, w_1), (w_0, w_2) \,\} \\
S_{f_1} &= \{\, (w_1, w_5), (w_2, w_7), (w_3, w_9), (w_4, w_{11}) \,\} \\
S_{f_2} &= \{\, (w_1, w_6), (w_2, w_8), (w_3, w_{10}), (w_4, w_{12}) \,\} \\
V(p) &= \{\, w_5, w_6, w_7, w_{10} \,\}
\end{aligned}
$$

Now, let us analyze belief change and reliability change of the judge in the Crown Court by our implementation. First, we input the above model $\mathfrak{M}_1$ and then the system initializes the values of R_j, S_{f_1}, S_{f_2}, $V(p)$, and the reliability of all agents in each state as in Fig. 2. After that, the system outputs the graphic result after inputting the model $\mathfrak{M}_1$ as in the left-hand side of Fig. 3.

Second, we input `[Sign(f2,p)=>j]` which represents that f_2 told the statement $\mathsf{Sign}(f_2, p)$ to j. Then, the system calculates such operator and shows the result as follows:

$$\mathfrak{M}_1, w_0 \models \mathsf{Bel}(j,\ \mathsf{Sign}(f_1, p)) \wedge [\mathsf{Sign}(f_2, p) \rightsquigarrow j]\mathsf{Bel}(j,\ \mathsf{Sign}(f_2, p)).$$

The graphic result of the above action can be shown in the right-hand side of Fig. 3.

Next, when j turns back to the inquiry stage, he/she finds that f_2 provides the statement $\mathsf{Sign}(f_2, \neg p)$. In this sense, j commits his/herself to the statement $\mathsf{Sign}(f_2, \neg p)$ by $[\mathsf{Sign}(f_2, \neg p) \rightsquigarrow j]$. This can be done by inputting an action `[Sign(f2,~p)=>j]` into the system. Then, the result of calculation by such operator can be shown in Fig. 4 and the graphic result can be shown in the left-hand

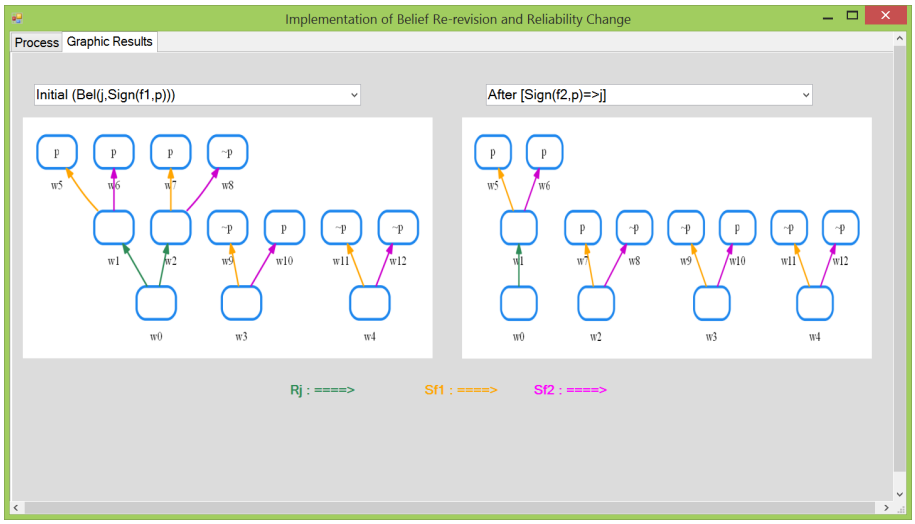


Fig. 3. The left-hand side is Kripke model of $\mathsf{Bel}(j, \mathsf{Sign}(f_1, p))$ and the right-hand side is Kripke model after $[\mathsf{Sign}(f_2, p) \rightsquigarrow j]$.

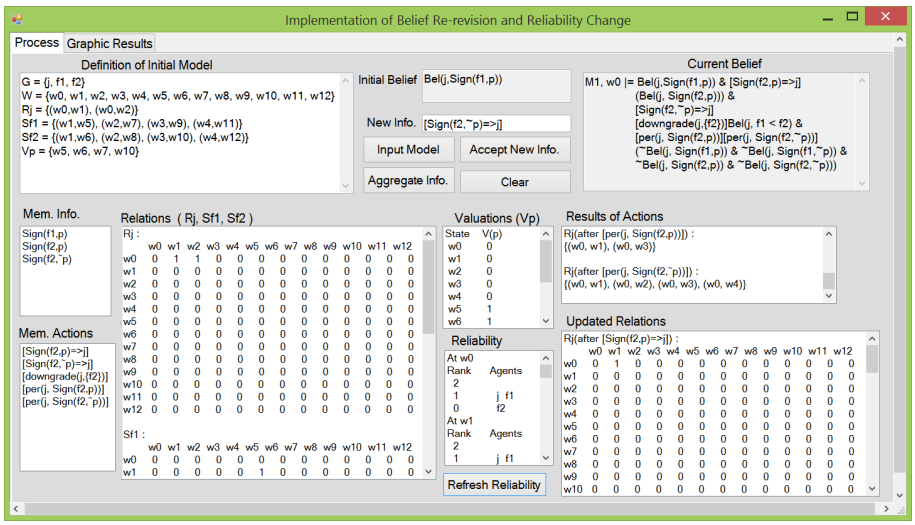


Fig. 4. Program after inputting $[\mathsf{Sign}(f_2, \neg p) \rightsquigarrow j]$

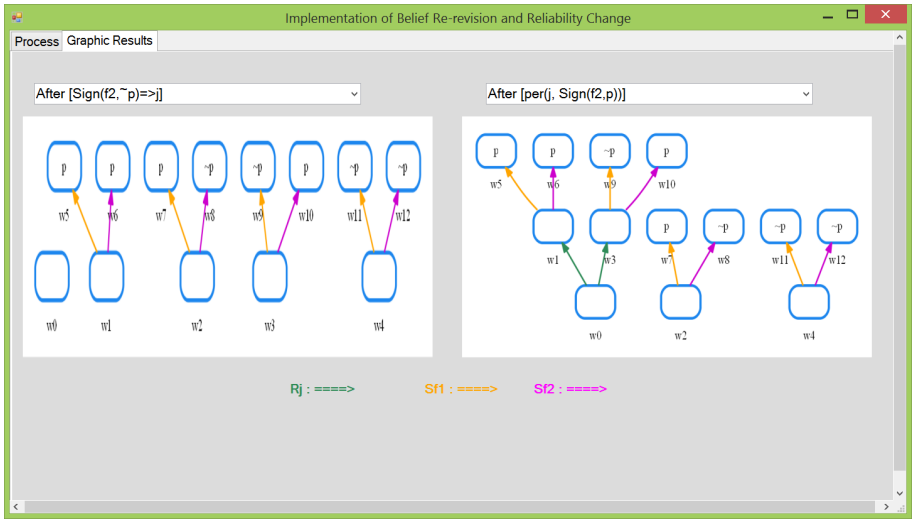


Fig. 5. The left-hand side is Kripke model after $[\mathsf{Sign}(f_2, \neg p) \rightsquigarrow j]$ and the right-hand side is Kripke model after $[\mathsf{per}_j(\mathsf{Sign}(f_2, p))]$.

side of Fig. 5. After that, the system will check if the received information is inconsistent with j's belief or not. If there is inconsistency, the system automatically performs three actions as the following sequences.

1) Based on j's decision in Section 2, we can regard that j considers that the statements of f_2 are inconsistent. In other words, f_2, who gives the inconsistent statements, can be regarded to be unreliable. So, we regard that j needs to downgrade the agent f_2 by $[\{f_2\} \Downarrow^j]$. In our system, an action `[downgrade(j,{f2})]` which represents $[\{f_2\} \Downarrow^j]$ is performed. Then, the system outputs the result after this downgrading as "`Bel(j,f1<f2)`" representing $\mathsf{Bel}(j, f_1 <_j f_2)$, i.e., the agent f_2 becomes less reliable than the agent f_1 from j's perspective.
2) By the update of $[\mathsf{Sign}(f_2, \neg p) \rightsquigarrow j]$, the system deleted all the links from w_0 into the states where $\mathsf{Sign}(f_2, \neg p)$ is false, and as a result, there is no links from w_0 (see the left-hand side of Fig. 5). So, we regard that j needs to permit the possibility of both $\mathsf{Sign}(f_2, p)$ and $\mathsf{Sign}(f_2, \neg p)$ by $[\mathsf{per}_j(\mathsf{Sign}(f_2, p))]$ and $[\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))]$, respectively. This can be done by performing an action `[per(j,Sign(f2,p))]` which represents $[\mathsf{per}_j(\mathsf{Sign}(f_2, p))]$ and an action `[per(j,Sign(f2,~p))]` which represents $[\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))]$ in our system. $[\mathsf{per}_j(\mathsf{Sign}(f_2, p))]$ and $[\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))]$ aim to restore the former links to the states where $\mathsf{Sign}(f_2, p)$ and $\mathsf{Sign}(f_2, \neg p)$ are true, respectively. After that, the system outputs the graphic results of $[\mathsf{per}_j(\mathsf{Sign}(f_2, p))]$ and $[\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))]$ as shown in the right-hand side of Fig. 5 and in the left-hand side of Fig. 6, respectively.

After the above actions, j believes the following information.

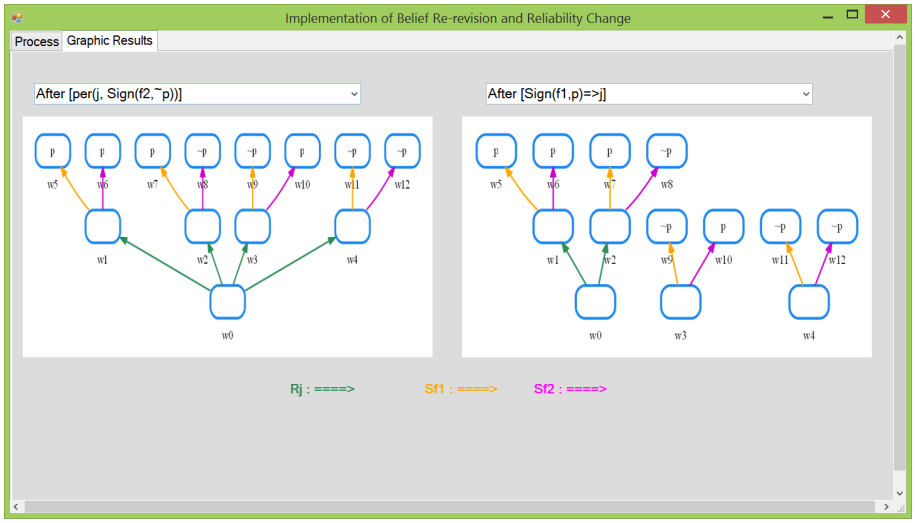


Fig. 6. The left-hand side is Kripke model after $[\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))]$ and the right-hand side is Kripke model after $[\mathsf{Sign}(f_1, p) \rightsquigarrow j]$.

$$\begin{aligned}\mathfrak{M}_1, w_0 \models & \mathsf{Bel}(j,\ \mathsf{Sign}(f_1, p)) \wedge [\mathsf{Sign}(f_2, p) \rightsquigarrow j]\mathsf{Bel}(j,\ \mathsf{Sign}(f_2, p)) \wedge \\ & [\mathsf{Sign}(f_2, \neg p)) \rightsquigarrow j][\{f_2\} \Downarrow^j]\mathsf{Bel}(j, f_1 <_j f_2) \wedge \\ & [\mathsf{per}_j(\mathsf{Sign}(f_2, p))][\mathsf{per}_j(\mathsf{Sign}(f_2, \neg p))] \\ & (\neg\mathsf{Bel}(j, \mathsf{Sign}(f_1, p)) \wedge \neg\mathsf{Bel}(j, \mathsf{Sign}(f_1, \neg p)) \wedge \\ & \neg\mathsf{Bel}(j, \mathsf{Sign}(f_2, p)) \wedge \neg\mathsf{Bel}(j, \mathsf{Sign}(f_2, \neg p)))\end{aligned}$$

We can denote $\mathfrak{M}_2$ by the updated model of $\mathfrak{M}_1$ after the above actions. From the decision in the Crown Court, we may regard that j admits the statements of f_1, i.e., $\mathsf{Sign}(f_1, p)$ by $[\mathsf{Sign}(f_1, p) \rightsquigarrow j]$, and the graphic result can be shown in the right-hand side of Fig. 6. Therefore,

$$\mathfrak{M}_2, w_0 \models [\mathsf{Sign}(f_1, p) \rightsquigarrow j]\mathsf{Bel}(j,\ \mathsf{Sign}(f_1, p)).$$

Let us denote $\mathfrak{M}_3$ by the updated model of $\mathfrak{M}_2$ after $[\mathsf{Sign}(f_1, p) \rightsquigarrow j]$. Finally, j aggregates information by the careful policy and will believe that r_2 is the shooter (p) as follows:

$$\mathfrak{M}_3, w_0 \models \mathsf{Bel}(j, p).$$

6 Conclusion

This study proposed an implementation of our logical formalization for analyzing an agent's belief re-revision and reliability change. Several dynamic operators,

i.e., private announcement, permission and joint downgrade operators were presented in terms of dynamic epistemic logic. With a combination of these dynamic operators, we demonstrated our implementation in an example of legal case from British Columbia. In the inquiry stage, the witnesses gave the statements to the police, but after then one of them changed their statements in the Crown Court. In order to analyze this process, our dynamic operators were employed in our implementation. The main contribution in this study is to implement our logical formalization in a computer system to show its adequacy.

For the future works, we aim to consider more sophisticated ways to construct a process of the accessibility restoration by the permission operator. In our logical formalization, we consider information about only one topic. In this sense, the permission operator can work well. Nevertheless, if there is more than one topics, it may cause a problem called bad side-effect. For example, an agent j first believes p and q ($\mathsf{Bel}(j,p) \wedge \mathsf{Bel}(j,q)$). Then, if j needs to permit the possibility of $\neg p$, $[\mathsf{per}_j(\neg p)]$ is employed. By the update of $[\mathsf{per}_j(\neg p)]$, j will not believe p ($\neg\mathsf{Bel}(j,p)$) and may not believe q ($\neg\mathsf{Bel}(j,q)$). In other words, $[\mathsf{per}_j(\neg p)]$ affects not only $\mathsf{Bel}(j,p)$ but also $\mathsf{Bel}(j,q)$. In fact, $[\mathsf{per}_j(\neg p)]$ should not affect the other propositions than p. Therefore, our next goal is to avoid this problem.

References

1. Dragoni, A., Giorgini, P.: Revising beliefs received from multiple sources. In Williams, M.A., Rott, H., eds.: Frontiers in Belief Revision. Volume 22 of Applied Logic Series. Springer Netherlands (2001) 429–442
2. Roorda, J.W., van der Hoek, W., Meyer, J.J.C.: Iterated belief change in multi-agent systems. In: AAMAS, ACM (2002) 889–896
3. Baltag, A., van Ditmarsch, H.P., Moss, L.S.: Epistemic logic and information update. In Adriaans, P., van Benthem, J., eds.: Handbook on the Philosophy of Information. Elsevier Science Publishers (2008) 361–456
4. Bench-Capon, T.J.M., Prakken, H.: Introducing the logic and law corner. J. Log. Comput. **18** (2008) 1–12
5. Grossi, D., Rotolo, A.: Logic in the law: A concise overview. Logic and Philosophy Today, Studies in Logic **30** (2011) 251–274
6. Jirakunkanok, P., Hirose, S., Sano, K., Tojo, S.: Belief re-revision in chivalry case. In: New Frontiers in Artificial Intelligence. (2013) 230–245
7. Jirakunkanok, P., Sano, K., Tojo, S.: Analyzing reliability change in legal case. In: Proceedings of the Eighth International Workshop of Juris-Informatics. (2014) 52–65
8. Jirakunkanok, P., Sano, K., Tojo, S.: Analyzing reliability re-revision by consideration of reliability change in legal case. In: Proceedings of the Seventh International Conference on Knowledge and System Engineering. (2015) 228–233
9. Lorini, E., Perrussel, L., Thévenin, J.: A modal framework for relating belief and signed information. In: Computational Logic in Multi-Agent Systems. Volume 6814. (2011) 58–73

A Study of Ontology for Civil Trial

Yuya Kiryu[1], Atsushi Ito[1], Takehiko Kasahara[2],
Hiroyuki Hatano[1], Masahiro Fujii[1], Yu Watanabe[1]

[1]Utsunomiya University, 7-1-2 Yoto, Utsunomiya, Tochigi, 321-8505, Japan
kiryu@degas.is.utsunomiya-u.ac.jp, {at.ito, hatano, fujii, yu}@is.utsunomiya-u.ac.jp
[2]Toin Yokohama University, 1614 Kuroganecyo, Aoba-ku, Yokohama-shi, Kanagawa, 225-8503, Japan, kasahara@toin.ac.jp

Abstract. Recently, ICT has been adopted for use by the judiciaries in many countries such as Singapore and a specialized paperless Cybercrime court in India, in response to the global changes in the economy and politics. It is urgently necessary for Japanese judiciary to adopt ICT to catch up with the global trend. This paper initially points out problems in the administration of the current judiciary system in Japan, especially problems of accessing and managing legal information. We next propose a system to solve such problems, recommending WEB based system as one that might be appropriate for solving the problems. Finally, we propose an ontology to be appropriate to such a system to determine what information and document metadata is produced in the course of a Japanese civil trial.

Keywords: Cyber court, High Tech court, Legal document, XML, Ontology, Semantic Web,

1 Introduction

Recently, the adoption of ICT by judiciaries is progressing in many countries to reflect global economic and political changes. In particular, a judicial system incorporating ICT is called a "Cyber Court" or a "High Tech Court". For example, the Supreme Court of Singapore introduced ICT in 1998 [1]. They have introduced a Centralized Display Management System, a Digital Transcription System, E-Signatures, Electronic Hearings, a JusticeOnLine System, etc. All documents are displayed on PC screens in the court [2]. Also, High Court of Delhi and District Court of Delhi introduced paperless E-Court [16]. From the standpoint of technological research, there are many studies about models combining law and knowledge structures of law [3], however it is very rare to find a model to create an experimental system for ICT based judiciaries.

We have decided that it is very important to develop a prototype of a Cyber Court to evaluate the effectiveness of such an idea. We developed the first prototype of a Cyber Court in 2004 at Toin University of Yokohama [4,15] and demonstrated its

adfa, p. 1, 2011.

effectiveness, especially for the recently adopted *Saiban-in* system (Japanese jury system) [5]. Our previous Cyber Court Prototype mainly focused on using the Internet, such as through Internet based teleconferences, recording all hearings on video stored in a cloud and digitizing all documents. However it did not include a function enabling participate the judiciary, such as being able to submit a complaint to a court by not only a lawyer, but also an ordinary citizen using the Internet and a accessing judicial records on line. Our next step in developing the Cyber Court is intended to improve access to the judiciary by the people. Online complaints and online reading of the docket and documents record of legal requires a system and environment that would allow anyone to submit and access the judicial record easily.

As described in the first paragraph of this section, the Singapore Supreme Court provides functions to convert legal documents into electric data for people who can't come to the Supreme Court, however, it has not led to increased ICT literacy and it is not easy to access and retrieve digitized documents through the Internet. Even in Japan, although ICT use has spread widely by now, not many people have personal scanners in the homes. To make a legal system useful and accessible, it is necessary to avoid use of special hardware, such as a scanner, and special software, such as commercial word processor applications.

In this paper, we propose a legal document access system that supports not only legal professionals but also ordinary citizens to allow them to access the judiciary system. Also we analyze the types of information in judicial documents in detail, especially those used in civil trials, to further develop a legal documents access system.

In section 2, we discuss the current status of access to judicial documents, and in section 3, we introduce the related works. In section 4, we discuss the problems encountered in realizing an advanced legal access system. Based on the discussing, we propose the design of an online legal data access system in section 5. In section 6, we describe the ontology that should be used in the proposed system in detail. Finally, we summarize our new Cyber Court System and propose further research is the conclusion.

2 Status of online access to judicial documents in Japan

2.1 Access Precedent

Some Japanese case records are now open access. We can access them online and search them [6, 17]. The search screen is shown in Figure 1, Figure 2. We think that the legibility and flexibility of the search function are not so well developed since the search system requires knowledge of the legal system and the court system, such as the court name, case number, trial date and time. In addition, at the bottom of the page in Figure 1, there are character string searches, however it allows the combination of up to three words connected by "OR" and the combination of three of those connected by "AND". In addition, it is not possible to use "NOT" to set the query, thus, it is difficult for an ordinary person to use it effectively.

One of the reasons for having such a simple search system is that the current judicial records are managed on paper or in PDF format. It usually entails substantial cost

to add keywords to the judicial record. If we want to build a database that allows a flexible search, we will need to add complex metadata instead of simple keywords.

Figure 1

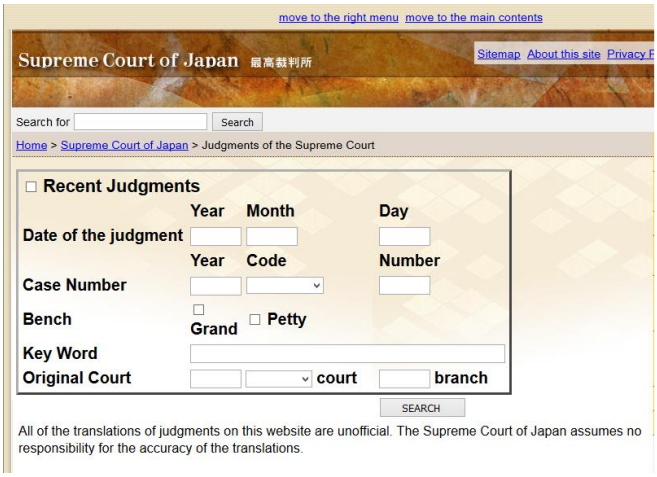


Figure 2

2.2 Online demand procedure

Another online system used to access judicial data in Japan is the “online demand process” [7]. This system requires installation of a dedicated program to use it. The

platform is also limited to Windows, so that it is now possible to access from a Mac or Smartphone. Though this system solves the problems of distance and time for the demand procedure, it creates another problem that requiring a computer environment and knowledge of ICT. The problem of using a computer means that, according to that form, specific platform, a JRE (Java Runtime Environment) [8], is required. Of course, this problem may be resolved as ICT progresses.

The problem of having knowledge of ICT is more serious. This is because JRE is necessary to install and use this program. It may be no problem for people who have enough knowledge of ICT to install it. But, it may be a problem for ordinary people who don't have enough knowledge of ICT. There seem to be quite a few people who can use this system even if we prepare them no matter how many polite commentary pages and videos we offer.

The technique introduced to enhance online access to the judiciary should not become a wall. It should be designed and implemented to enhance access to the judiciary, and besides, even serving that anyone can easily use it is important.

3 Related Work

3.1 Cyber Court

The pioneer study of a Cyber Court System is Courtroom 21 [9]. It started in 1993 at William & Mary University as a joint project of that university and the National Center for State Courts in the U.S.A. They implemented a cyber court using a teleconference system and performed many trials to determine the effectiveness of ICT in court. They also use such technology to support education at their Law School. At this time, they are mainly focusing on judicial and lawyer education and training, courtroom design and needs assessments.

3.2 Legal XML

OASIS (Organization for the Advancement of Structured Information Standards) has a project "LegalXML Electronic Court Filing TC" to define a standard to create and transmit legal documents among attorneys, courts, litigants, and others by using XML. OASIS defined Electronic Court Filing Version 4.01 as the OASIS Standard on May 23, 2013 [11]. This standard specified XML for court filing, however they do not define an ontology of legal documents.

3.3 Ontology for Legal Documents

There was a study to generate an ontology for a legal domain by using a special tool to describe ontology [12,14]. The paper describes the result of generating an ontology for a legal domain, however, the study was intended for generation of ontology. The result is not focusing on a specific procedure or real civil trial case.

4 Problems in Accessing Judicial Records

4.1 Outline of the problem

Here we will discuss the problems of technology used to access judicial documents that we described in section 2. The first one is the lack of flexibility in the search for judicial precedent information. The second one is complicated procedure required to prepare the environment to access judicial precedent information. For example, there was an online demand procedure system in the Sapporo District Court[1] was difficult to use and only one access attempt was succeeded during the trial period. In the following subsection, we would like to discuss these two problems in detail.

4.2 Problems in searching for precedent

With respect to the flexibility of the search, we think that the most important issue is about adding enough metadata to judicial record as mentioned in section 2.

Judicial records are handled and stocked as papers now. It is required to add metadata offline. Usually, judicial documents are described by using word processor, so that the original document is digital data. Because of the restriction about handling judicial documents, the electric data is printed on paper, and digitized (scanned) again and metadata added manually. This is an inefficient procedure and requires great expense. It is necessary to simplify this procedure.

In our opinion, the best method is to use the original digital data for searching documents, and instead of adding metadata manually, it is better to add metadata automatically and ask the authors of the documents to correct the automatically added metadata.

4.3 Problem in implementation of the system

The solution to the second problem, the complicated procedure, i.e., to prepare the environment for people without knowledge of ICT, is easy. If the system is implemented as a web-based application, users do not need to install a specific application.

In addition, the judicial records inputted via the web are unique data and no record exists locally except for scree print. By using a web-based input service, it is easy to verify authenticity of the data.

5 Proposed system

5.1 Architecture for the proposed system

To solve the problems in the current system involving judicial document description and retrieval, we would like to propose the system for civil trials described in

[1] This system has stopped on March 2009.

Figure 3. Figure 3 illustrates the architecture of our proposed system. Users input data to the server through a web browser and also access judicial records by using flexible query language such as Sparql [10] referring to the ontology of judicial record.

This system makes it possible to describe all documents related to the lawsuit, and to manage all legal starting from a complaint to judgment or settlement. Since it is the web based application, users can access it from the browser of their own PCs. We think that anyone can use it easily. For user management, an ID and passwords should be used in this system. One-time passwords may be useful because of the widespread adoption of smartphones. This technique, however, may become a barrier to people without knowledge of ICT, so we have to be cautious about that. In addition, as well to a current online demand system, an e-signature should be required. It is necessary to solve the problems relating to e-signatures, since there is not much awareness of e-signatures and the procedure of issuing e-signatures is complicated [13].

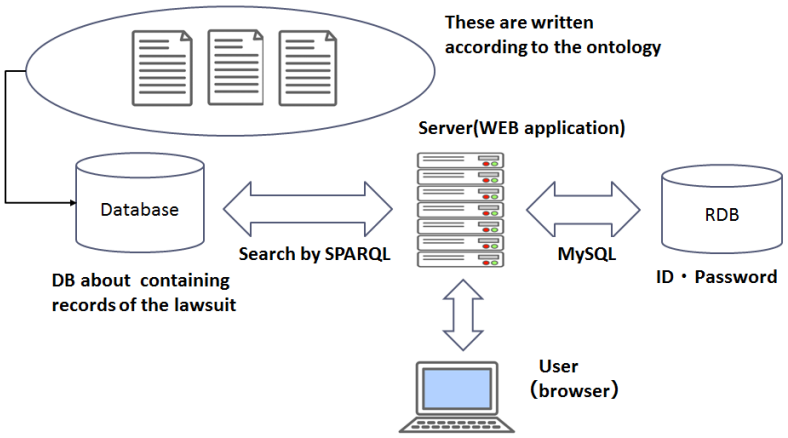


Figure 3

5.2 Metadata of the trial record

We would like to explain the left side of Figure 3. This part is the key function of this system. The purpose of this system is "to let users add metadata to trial records and support them by describing and managing case records without special effort by the users". The users can write documents visually by using an input form with a web application.

If we use a web based input form, the characteristics of the information are clearly defined (e.g., including the name of the plaintiff, an address, the postal code). Therefore, the system can add appropriate metadata for each form inputted on the server side. Even if the input data is complicated sentence, we can add metadata appropriately after the input data is parsed on the server.

The system should add appropriate metadata for an appropriate data such as words, sentences. The quality and quantity of the metadata in the electronic version of records of the lawsuit cannot exceed that of the paper edition. In other words, if the system simply recognizes the contents of the case records, the results may be similar to those in the paper version. Of course, this looks better than the metadata founded in the current precedent database, but there is a problem.

As described in Figure 4, currently case records are printed on paper as the summary of a case. If digital data is developed from this summary, the quality of the metadata is not better than that of previous one.

To solve this problem and to add more useful metadata, we suggest a "case records ontology", as shown in Figure 5. The case record is generated from original data of the case by using an ontology. Then a summary is generated from the original data.

To carry out this system, we have to define the civil case record ontology to provide enough information for both the user and the system.

In the following section, we will explain the ontology for a case record in a civil trial.

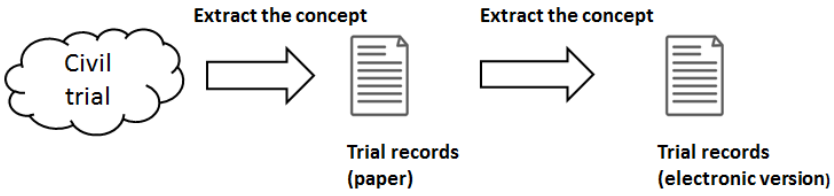


Figure 4

Figure 5

6 Ontology for civil trial

6.1 Basic design

In this section, we will describe the basic design of the ontology for a civil case. This ontology begins from the petition to the judgment or settlement. The ontology consists of two parts, as follows:

1. Information about the plaintiff and defendant
2. Various presentation procedures

In the following sub-sections, they are explained them in detail.

6.2 Ontology describing plaintiff and defendant

Figure 6 displays the relationship between the plaintiff and defendant. In Figure 6, the details of the defendant are not specified since the defendant has the same structure as the plaintiff.

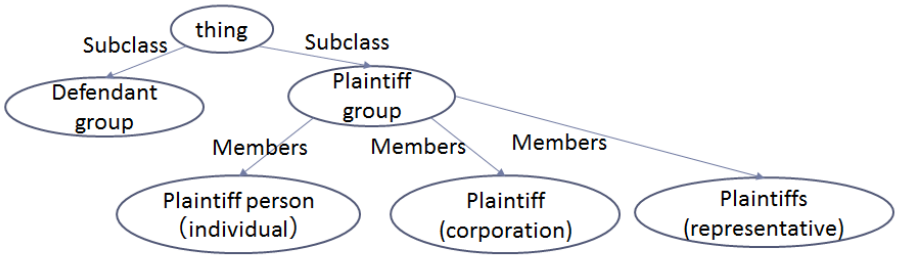


Figure 6

The above Figure starts with explanation added of the details of the plaintiff. A plaintiff person (individual), a plaintiff (corporation) and a plaintiff (representative) are members of the above plaintiff group. Each member has 0 or more elements.

Then, we explain the ontology about a plaintiff person (individual) in detail. A plaintiff person is a subclass of individuals as shown in Figure 7. Similarly, a defendant person is a subclass of individuals. Individual is subclass of person.

A plaintiff person has a name and a contact. A contact has a postal code, mailing address, phone number and e-mail address. FAX numbers are required in the current paper documents, but it is not entered because this system does not need it. These four items seem to be enough to provide contact information, however we should update this part of the ontology to include sufficient information.

The details of a plaintiff (corporation) are shown in Figure 8. The corporation has a corporation name, contact information and a representative of corporation. A representative of corporation is a subclass of person and gives the name of representative

and the job title of representative. The contact in Figure 8 is the same as the contact for the plaintiff person (individual) described in Figure 7, so the details are described in Figure 8.

The details of plaintiff (representative) are shown in Figure 9. A lawyer is a subclass of the person. A plaintiff (representative) has the contact and information of representative of the plaintiff group (including the name of law firm). Since the name of the plaintiff representative is an option, it is permitted a plaintiff group representative without name. In addition, a plaintiff representative can have multiple lawyers.

That is the complete explanation about the information related to plaintiffs and defendants. We indicated that the subclass of person, such as a plaintiff, a defendant, a corporate representative and a lawyer in the ontology. Strictly speaking, they are merely roles. However, the scope of this ontology is very limited as described in 6.1, so that we can consider these sub-classes as a person who has such roles in nature.

In the case of a counterclaim, there seems to be a problem since the human has two roles: plaintiff and defendant [14], however, there is no contradiction since each stage of civil trial is managed independently. In addition, a person may be defined twice if a plaintiff person and the representative of a corporation are the same. In this respect, what a person as individual can do differs very much from what a person as a representative of a corporation can do. Based on this consideration, we should treat each person as a totally different person.

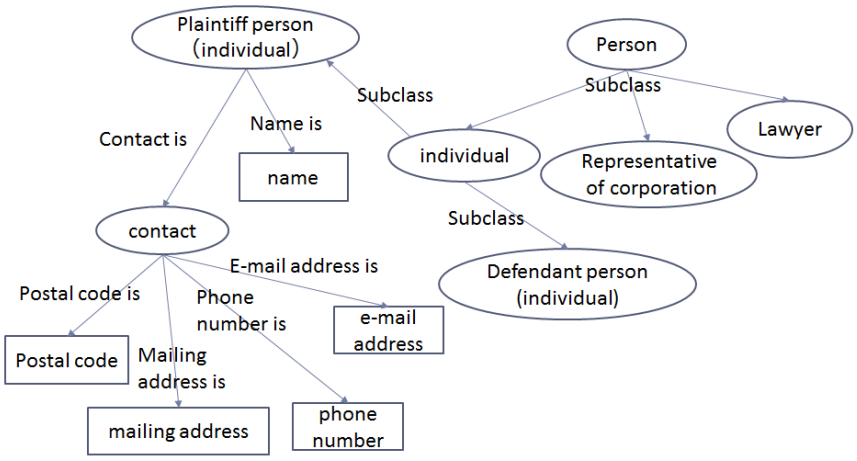


Figure 7

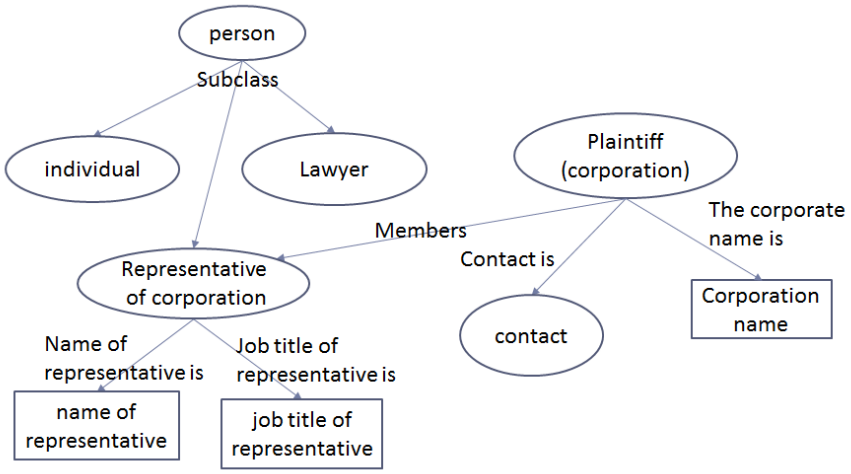


Figure 8

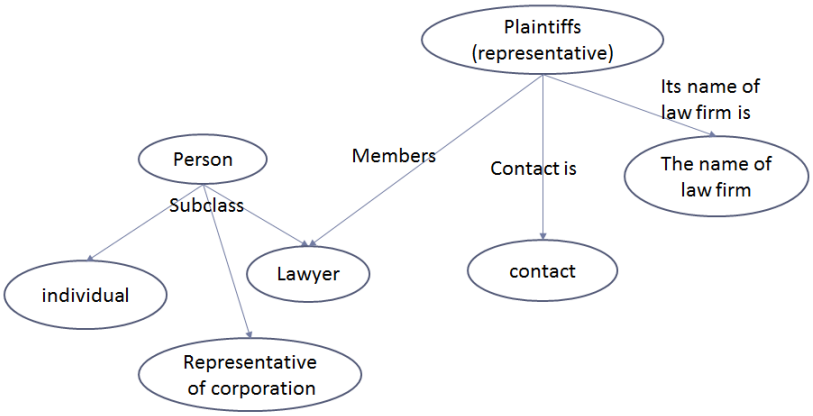


Figure 9

6.3 Ontology for various presentation procedures

In this sub-section, we explain the ontology of "Various presentation procedures" shown in6.1. As described in Figure 10, this ontology focuses on the plaintiff group and the defendant group in Figure 6. There are many documents in a trial, however, only a petition is given as an example in Figure 10. We use part of the petition shown in Figure 11 and Figure 12 for reference [13].

Each document is a subclass of all of the documents. The petition has a body, date, attached document and data. The e-signature is included in the attached data.

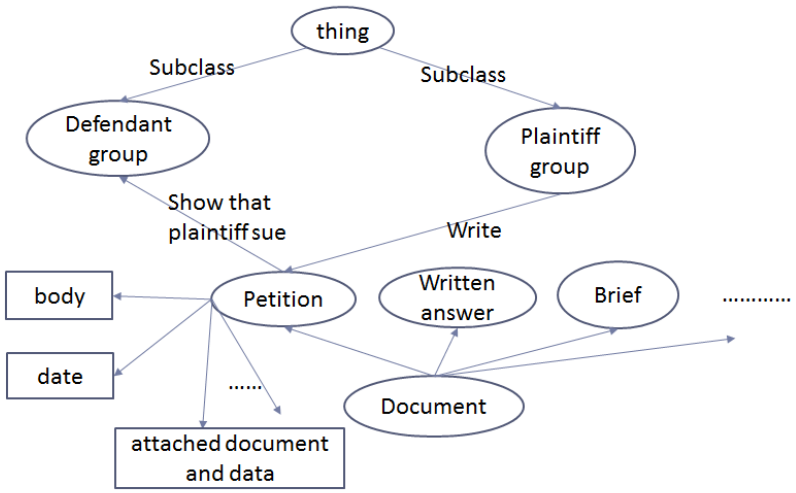

thing
Subclass
Subclass
Defendant group
Plaintiff group
Show that plaintiff sue
Write
body
Petition
Written answer
Brief
.....
date
.....
Document
attached document and data

Figure 10

訴状

平成２１年６月２日

福岡地方裁判所　御中

原告ら訴訟代理人弁護士　甲野太郎
同　乙野花子

〒８１０－００７４　福岡県福岡市中央区大手門１－３００
原　告　秋葉　隆
〒８１０－００７４　福岡県福岡市中央区大手門１－３００
原　告　大手門不動産
上記代表者代表取締役　秋葉　隆
〒８１０－００４２　福岡県福岡市中央区赤坂８００

松竹法律事務所（送達場所）
原告ら訴訟代理人弁護士　甲野太郎
同　乙野花子
電話　０９２－２２２－１１１１
FAX　０９２－２２２－２２２２

〒８１３－００２５　福岡県福岡市東区青葉１０００
被　告　井上　徹

土地建物明渡等請求事件

訴訟物の価額　２７８万４５４１円
貼用印紙額　１５９０００円

第１　請求の趣旨
１　井上徹は、秋葉隆らに対し、別紙物件目録１記載の土地及び建物並びに別紙物件目録２記載の土地を明け渡せ
２　井上徹らは、秋葉隆ら各自に対し、連帯して６２９万０００円並びに本訴状送達の日の翌日から別紙物件目録１記載の土地、建物及び別紙物件目録２記載の土地の明渡済みまで１か月２３万０００円の割合による金員を支払え
３　訴訟費用は被告らの負担とする

Figure 11

Complaint

June 2, 2009

Dear Hukuoka District Court

Representative of the plaintiff Taro Kono
Same as above Hanako Otsuno

Postal code 810-0074 1-300 Otemon, Chuo Ward, Hukuoka City, Hukuoka Perf.
Plaintiff Takashi Akiba
Postal code 810-0074 1-300 Otemon, Chuo Ward, Hukuoka City, Hukuoka Perf.
Plaintiff Otemon real estate
Rpresentative director Takashi Akiba
Postal code 810-0042 800 Akasaka, Chuo Ward, Hukuoka City, Hukuoka Perf.

Shochiku Legal Information Center (Place of Service)
Representative of the plaintiff Taro Kono
Same as above Hanako Otsuno
TEL 092-222-1111
FAX 092-222-2222

Postal code 813-0025 1000 Aoba, Higashi Ward, Hukuoka City, Hukuoka Perf.
Defendant Toru Inoue

Case about claiming land and buildings

Amount in controversy 2,784,541 yen
Amount of revenue 159,000 yen

First Object of claim
1 Toru Inoue should vacate the land and the building written in the property list 1 and the land written in the property list 2 to Takashi Akiba.
2 Defendant should pay 6,290,000 yen and the cost each in proportion to 230,000 yen a month from the next day this complaint was sent to the day vacating the properties written in the property list 1 and 2.
3 Defendant should have court Costs

Figure 12

7 Conclusion

In this paper, we have described a system that makes it possible to access all documents in a lawsuit, and to manage all judicial records from a the trial to the judgement or settlement. We also define ontology for civil trial.

We will update this ontology what should be the reflection of the requirements to describe legal documents relating to civil case. We have not yet defined the detailed metadata in the body of each document. We must study how to add metadata automatically to the body of the documents by parsing them.

Our next step will be to implement a prototype of a legal document access system based on the ontology that we have defined. We would like to discuss with legal professionals to further refine the ontology and prototype.

In addition, it is necessary to confirm its compatibility with a search for the judicial precedent information. Specifically, it is necessary to control the part that is not open to the public because of privacy to define a search form that can utilize the flexibility of SPARQL effectively and assess the search speed.

Acknowledgments.
We express special thanks to all people who joined the Cyber Court Project.

References

1. http://www.taf.or.jp/report/23/index-1/page/p069.pdf [Sep, 3, 2015]
2. https://www.supremecourt.gov.sg/default.aspx?pgID=361#10 [Sept, 3, 2015]
3. Proceedings of JURISIN 2013 and 2014
4. Takehiko Kasahara, “Cybercourt - Court Technology”, Journal of the Japanese Society for Artificial Intelligence vol. 23 Nr. 4, pp 513- [Jul. 2008]
5. “Can you judge?”, NHK Special, 2005.2.13, PM9:00- (http://www6.nhk.or.jp/special/detail/index.html?aid=20050213) [Sept, 4, 2015]
6. http://www.courts.go.jp/app/hanrei_jp/search1 [Sept, 4, 2015]
7. http://www.tokuon.courts.go.jp [Sept, 4, 2015]
8. http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html [Sept, 4, 2015]
9. http://law.wm.edu/academics/intellectuallife/researchcenters/clct/ [Sept, 4, 2015]
10. http://www.w3.org/TR/rdf-sparql-query/ [Sept, 4, 2015]
11. http://docs.oasis-open.org/legalxml-courtfiling/specs/ecf/v4.01/ecf-v4.01-spec/os/ecf-v4.01-spec-os.html [Oct, 12, 2015]
12. Yamaguchi Takahira, Kurematsu Masaki, Aoki Chizuru, Sekiuchi Rieko, Kagayama Shieru and Yoshino Hajime, "A Domain Ontology Development Environment Using a Machine Readable Dictionary ("Ontology : Foundations and Applications")", Journal of the Japanese Society for Artificial Intelligence, Vol.14, No.6, pp.1080-1087 (Nov. 1999)
13. "A study of applying ICT for legal service", Report of a research funded by the Ministry of Ministry of Internal Affairs and Communications in Japan in 2009, (March, 2010)
14. Riichiro Mizoguchi, “*Ontology engineering*”, Ohmsha, [Jan, 2005]
15. Takehiko Kasahara, “Court Technology for Civil Procedure”, Festschrift for seventieth birthday of Prof. Takeshi Kojima Ⅱ, pp. 961- (Sep. 2009)
16. http://www.indiaegov.org/documents/Inauguration%20of%20E-Court.pdf [Oct, 12, 2015]
17. http://www.courts.go.jp/app/hanrei_en/search? [Oct, 13, 2015]

Argumentation Support Tool with Reliability-Based Argumentation Framework

Kei Nishina, Yuki Katsura, Shogo Okada, and Katsumi Nitta

Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa, Japan
{nishina,katsura_y,okada,nitta}@ntt.dis.titech.ac.jp

Abstract. In legal debates, it is a matter of importance whether one's own argument is accepted or not.For this, we propose evaluation method for calculating the acceptability of arguments, and a tool developed based on the measures. This method is called reliability-based argumentation framework(RAFs), extended from argumentation framework, seeking for multivalued dialectical validities of arguments reliable to some extent. The modular reliability-based argumentation framework(MRAF) based on RAFs is able to integrate the RAF semantics in every module. This leads to an over-all valuation of the acceptability of argumentations including several local arguments. The argumentation-support tool can represent the utterance logs of those who join an debate, the argumentation diagram its users made, and the argumentation framework converted from this, contributing to the intuitionistic comprehension of the logical structures of arguments and their acceptability.This tool also enables represented argumentation framework to be converted into modular structures of local AFs, leading to an overall valuation of the acceptability of arguments.

Key words: Argumentation theory, semantics, argumentation framework, reliability, modular structure

1 Introduction

A legal debate is a discussion between lawyers, legal academics and others. Arguments are series of statements typically used in such a debate to persuade someone to do something or to present reasons to let him or her accept a conclusion. It is an important problem for participants to know which ones of the all the arguments come to be finally accepted. To solve this problem, argument diagrams, visual representation of the structure of arguments made in the discussion, can be seen being often used. There have been several researches done about such argument diagrams, such as the Araucaria system [1] which provides an interface which supports the Toulmin's diagramming [2] process and then saves the result, and the theory of Argumentation Framework (AF) [3] which is one of the fundamental theories of the field of argumentation theory, and has been extended in various ways to accommodate different kinds of abstraction about argumentation [4–6] and etc.

To teach legal debate skills to students of the law school, several argumentation support tools have been developed. Some tools have functions to visualize the structure of

argumentation in the form of diagram, and efficiency of using diagrams is studied by Pinkwart, et. al [7]. We have also developed an online argumentation support tool based on diagramming. This tool is installed on each tablet PC of each participant of the legal debate. Each participant inputs his arguments through the Toulmin's diagramming-like interface of his tablet PC, and then the input arguments are transformed into AF and its semantics are calculated. By this information, to win the argumentation the participants know which argument he should attack next. However, the AF diagram tends to become bigger as the discussion gets longer or more complicated. To cope with this problem, we extended functions of the argumentation support tool. The extended tool freely build AFs displayed on the screen into the module structures consisting of multiple local AFs unless none of the information of each component is lost, and the comprehensive acceptability of the each argument in the debate can be derived by combining each local acceptability of the each argument in the local AF is necessary to address these issues. To extend the argumentation support tool, we also have to extend the AF theory by structuring it into distinct modules in hierarchical structure. There has already been several theoretical studies of AFs whose structure is of a modular structure which resembles our view about the support tool, such as Modular Assumption-Based Argumentation(MABA)[8], Argumentation Context System(ACS)[9], and hierarchical Extended Argumentation Framework(hEAF)[10].

However, the arguments, as are seen in legal, social and political debates, in which some or other correctness is questioned should be compatible with the facts with which they are concerned, the professional knowledge presupposed by them, and the conclusion of their related arguments. Furthermore, the validity of each argument in such a discussion as these depends on the status of its acceptability, if it is found in other discussions as well. According to this view, based only on the dialectical acceptability derived from AF semantics, it is impossible to ascertain the compatibility with the background knowledge supporting each argument in such discussions. This is because not all of the facts related to their themes are stated in the arguments. It is impossible either to represent this compatibility as the relative ordering in PAF[11] or VAF[12].

Therefore, to analyze the validity of each argument in the discussion closely related to the circumstantial facts, knowledge and opinions by keeping an eye on the compatibility with their information and other discussions, including the meta-information of the discussion, we believe that one needs to grasp the status of the compatibility or acceptability of each argument in the discussion. For example, assume a discussion on a social issue and the case in which the argument made in that discussion is taken as valid arguments in AF semantics, which valuates the dialectical acceptability. If, then, the content of the argument is turned out to be in accord with that of a gratuitous rumor, from the meta-information of the discussion, it should be excluded from the result of the discussion considering the meta-information. Or else, if the context of this argument turns out not contradictory to the context of the hypothesis made by an expert well-acquainted with its related fields, from the meta-information of this argument, it should be regarded as a credulous result of the discussion considering the meta-information, which is roughly valid.

Furthermore, if the content of the argument turned out to be in accord with the results, the facts, or the expert knowledge of its related arguments, it should be regarded as

the skeptical result of the discussion considering the meta-information, which is strictly valid. To sum up, we believe that the acceptability of each argument in the discussion relating to the background meta-information depends on both the consistency between its context, called its reliability, and the background knowledge on one hand, and its result of assessment in AF semantics on the other hand.

We, then, propose Reliability-based Argumentation Framework (RAF) whose semantics consider the reliability of each argument. It is important to note that the reliability of an argument is different from the "trust" of trust-extended argumentation graph[13], which comes from the relationship of trust between the participants of discussion, and the appearance frequency of the argument of PrAF[14]. In a sense, the concept of the semantics of RAF is similar to the one of Authority Degree-Based Evaluation Strategy's concept[15] to the effect that the argument made by the expert well-acquainted with relative fields of its content has more significance than the one made by the outsiders. But, in RAF semantics, the argument made by the outsiders which agrees with the context of the expert of the relative domain of the context of it is regarded has major significance than the gratuitous one, and the number of the attackers and defenders of each argument in discussion is not taken into account because we follow the part of the concept of the acceptability of AF semantics.

Our research objectives are focused on the following two points concerning the analysis of real discussions: The first point is the theoretical approach extending AF semantics so as to take the reliability of each argument into account in the way which is different from the semantics of Assumption-based Argumentation framework (ABAF) [16] and Probabilistic Argumentation Framework (PrAF)[14]. And, the second point is the analytical approach developing a dynamic argumentation support tool for real time analysis of real discussions whose function is visualization of argument diagram, to derive the evaluation results of the acceptance of each argument, and to provide strategic direction. The paper is structured as follows: In section 2, we review the background of argumentation theory. In section 3 and 4, we define RAF, MRAF, and their semantics. Furthermore, we outline the architecture, and the functions of the support tool in section 5, and conclude this paper in section 6.

2 Theoretical Background

An argumentation framework [3] is a tuple $AF = (Ar, attacks)$, where Ar is a set of arguments, and attacks $\subseteq Ar \times Ar$ is a binary attack relation on Ar. An attack from an argument $a \in Ar$ to an argument $b \in Ar$ is expressed as $(a,b) \in attacks$.

- The set $S \subseteq Ar$ is *conflict-free* iff $^{\forall}x,y \in S, (x,y) \notin attacks$.
- For any $x \in Ar$, x is *acceptable* with respect to some $S \subseteq Ar$ iff $^{\forall}y \in Ar$ s.t. $(y,x) \in attacks$ implies $^{\exists}z \in S$ s.t. $(z,y) \in attacks$.
- $F_{AF} : 2^{Ar} \rightarrow 2^{Ar}$, and $F_{AF}(S) = \{a \in Ar \mid a \text{ is acceptable w.r.t. } S\}$.
- $S \subseteq Ar$ is complete extension iff S is *conflict-free* and $F_{AF}(S) = S$.

Caminada's theory of reinstatement labelling [17] is more expressive than the extensions defined in Dung's theory. AF-labelling is a total function $L : Ar \rightarrow \{\mathbf{in}, \mathbf{out}, \mathbf{undec}\}$. The label **in** indicates that the argument is explicitly accepted, the label **out** indicates

that the argument is explicitly rejected, and the label **undec** indicates that the status of the argument is undecided. This theory can convert every extension of Dung's AF into the set of in-labelled arguments by each reinstatement labelling.

3 Reliability-Based Argumentation Framework

In this section, we formally define Reliability-Based Argumentation Frameworks (RAFs) and their semantics by extending the AFs defined by Dung. An RAF is a directed graph consisting of nodes which have one degree of reliability and links between nodes. The nodes represent argument and the links represent attack relation on arguments, like the AFs defined by Dung. Furthermore, each argument has one degree of reliability.

Definition 1. *A reliability-based argumentation framework (RAF) is a tuple* $(Ar, attacks, ST)$*, where Ar is a set of argument,* $attacks \subseteq Ar \times Ar$*, and ST is a function mapping from each argument in Ar to any one element in the set of reliabilities, Re, defined as* $\{\mathrm{sk}, \mathrm{cr}, \mathrm{def}, \mathrm{unc}\}$.

The set *Re* consists of *sk*, *cr*, *def*, and *unc*, representing statuses of reliability. Note that there is no decision rule determining which one element of Re should be mapped in the theory of RAF. The elements of Re represent the statues of reliability of the argument in Ar, and they are derived by seeing detail of the meta-information of the content of the discussion which represent a RAF. For example, they are derived by the following interpret:

The argument in Ar whose content agrees with any fact, established theory, or the result of the result of the discussion which considered to be closed is assigned to the status of reliability, sk (skeptical).

The argument in Ar whose content agrees with any issue which is possibly true or opinion which isn't one-sidedly rejected is assigned to the status of reliability, cr(credulous).

The argument in Ar whose content agrees with any false rumor or rejected opinion is assigned to the status of reliability, def(defeated).

The argument in Ar whose content is collided with any meta-information of the content of the discussion represented by RAF is assigned to the status of reliability, unc(uncertain).

Extensional semantics for RAF define the detailed acceptability of arguments in real discussions and considers the reliability of each argument and the attack relations between those arguments. We propose three kinds of semantics based on three kinds of acceptability of arguments in *Ar*. They came from different intuitive perspectives on the persuasiveness of arguments in real discussions.

First, we adopt an intuitive perspective that unfounded arguments and the attack relations concerning them should be judged to be invalid in persuasiveness of arguments. Then, we define RAF-Non-Def semantics in which no argument whose reliability is *def* in *Ar*, (*defeated argument*) is included in any set in RAF-Non-Def complete extension, and every attack relation concerning any *defeated argument* is invalid.

Secondly, we adopt the optimistic perspective that the argument which isn't rejected in the discussion and has the status of reliability is *cr* or *unc* should be included in

the result of this discussion considering the content of the meta- information because there is no evidence in the meta- information which shows it is false . Then, we define RAF Optimistic semantics in which every argument in each set in Non-Def complete extension, and every argument which is not a *defeated argument* and is attacked only by the arguments whose status of reliability is *cr*, *unc*, or *def* are included by RAF Optimistic complete extension.

Finally, we adopt the pessimistic perspective that the argument which isn't rejected in the discussion and has the status of reliability is *cr* or *unc*, shouldn't be included in this result of the discussion considering the content of the meta- information because there is no evidence in the meta- information which shows it is absolutely true. Then, we define RAF Pessimistic semantics in which each set in every extension consists of the arguments whose reliability is *sk*.

Hence, the three semantics are formally defined based on the labelling extending Caminada's reinstatement labelling [17]:

We first define RAF-labelling and RAF-*conflict-free* labelling, following AF-labelling and AF-*conflict-free* labelling:

Definition 2. *A RAF-labelling in Δ is a total function $L : Ar \rightarrow \{\textbf{\textit{in}}, \textbf{\textit{out}}, \textbf{\textit{undec}}\}$. The set of in-labelled arguments by L, that of out-labelled arguments by L, and that of undec-labelled arguments by L are expressed as $in(L)$, $out(L)$, and $undec(L)$.*

Definition 3. *A RAF-labelling L is a RAF-*conflict-free *labelling iff no in-labelled argument by L attacks any other in-labelled argument by L and itself.*

Secondly, we define each complete labelling in RAF-Non-Def semantics, RAF-Optimistic semantics, and RAF-Pessimistic semantics.

Definition 4. *Let Δ be a RAF, and let L be a RAF-*conflict-free *labelling in Δ. Then, L is a RAF-Non-Def complete labelling iff it satisfies the following:*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{in}} \equiv \{ST(a) \neq def\} \wedge \{^{\forall}b \in Ar : [(b,a) \in attacks \supset L(b) = \textbf{\textit{out}}]\})$ *and*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{out}} \equiv \{ST(a) \neq def\} \vee \{^{\exists}b \in Ar : [(b,a) \in attacks \wedge L(b) = \textbf{\textit{in}}]\})$.

Definition 5. *Let Δ be a RAF, and let L be a RAF-*conflict-free *labelling of Δ. Then, L is a RAF-Optimistic complete labelling iff it satisfies the following:*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{in}} \equiv \{ST(a) \neq def\} \wedge \{^{\forall}b \in Ar : [(b,a) \in attacks \supset L(b) = \textbf{\textit{out}}]\})$ *and*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{out}} \equiv \{ST(a) = def\} \vee \{^{\exists}b \in Ar : [(b,a) \in attacks \wedge L(b) = \textbf{\textit{in}}]\}$
$\vee \{(ST(a) = cr \text{ or } unc) \wedge (^{\forall}b \in Ar : [(b,a) \in attacks \supset L(b) = \textbf{\textit{out}}])\})$.

Any RAF-Non-Def complete labelling of Δ is a RAF-Optimistic complete labelling of Δ, too. Any RAF-Non-Def complete labelling of Δ is a always RAF-Optimistic complete labelling of Δ, because every RAF-labelling in that meets the definition of RAF-Non-Def complete labelling meets the definition of RAF-Optimistic complete labelling.

Definition 6. *Let Δ be a RAF, and let L be a RAF-*conflict-free *labelling of Δ. Then, L is a RAF-Pessimistic complete labelling iff it satisfies the following:*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{in}} \equiv \{ST(a) = sk\} \wedge \{^{\forall}b \in Ar : [(b,a) \in attacks \supset L(b) = \textbf{\textit{out}}]\})$ *and*

$^{\forall}a \in Ar : (L(a) = \textbf{\textit{out}} \equiv \{ST(a) = def\} \vee \{^{\exists}b \in Ar : [(b,a) \in attacks \wedge L(b) = \textbf{\textit{in}}]\})$.

Definition 7. *Let Δ be a RAF, and let L be a RAF-σ-complete labelling of Δ, where $\sigma \in \{$Non-Def, Optimistic, Pessimistic$\}$).*

- *L is a RAF-σ-stable labelling of Δ iff it is a RAF-Non-Def complete labelling of Δ and $undec(L) = \emptyset$.*
- *L is a RAF-σ-preferred labelling of Δ iff $in(L)$ is maximal (w.r.t. set inclusion) among all RAF-σ- complete labelling of Δ.*
- *L is a RAF-σ-grounded labelling of Δ iff it is a RAF-Non-Def complete labelling of Δ and $in(L)$ is minimal (w.r.t. set inclusion) among all RAF-σ-complete labelling of Δ.*

Finally, let Δ be a RAF, σ-ε-extension of Δ, where $\varepsilon \in \{$complete, stable, preferred, grounded$\}$), is defined as the follow: $\{in(L) \mid L$ is a σ-ε-labelling of $\Delta\}$

4 Modular Reliability-Based Argumentation Framework

Each module in a modular reliability-based argumentation framework (MRAF) is a RAF representing the information of a local discussion. Two modules are in the hierarchical relationship when the argument expressed in one module is referred to as the argument in the other module. We call this the hierarchical relationship between two modules *reference relation*. One of the two modules in the *reference relation* is a higher module that refers to the argument in the other lower module. Note that the meta-information of the discussion which are represented in higher module is the content which are represented in the lower module by the form of RAF. Hence, MRAF represent the discussions which are end targets of analysis as the highest modules and the meta-information which is relative to them as lower modules.

Definition 8. *A Modular Reliability-Based Argumentation Framework is a tuple (MO,* refers*), where MO is a set of RAFs.* Refers *in the tuple is defined as* refers$\subseteq MO \times MO$, *which is a set of* reference relations *between modules in MO, and whose power set contains any element constructing the circular relationship of the elements of MO. The module $M \in MO$ is a higher module of the module $M' \in MO$, where M' is a lower module of M when $(M, M') \in$* refers *holds.*

We introduce the way of defining the reliability of the referred argument in the higher module derived by the function ST in the following, definition 9.

Note that the reliability of an argument in each RAF is defined uniquely in MRAF. Then, We express as an RAF taken as the module M as $\Delta_M = (Ar_M, attacks_M, ST_M)$.

Definition 9. *In every module, each argument's acceptability is calculated by only RAF-σ-semantics. The ranked reliability of the argument, a in module M, $ST_M(a)$ is defined as the following 1,2,3, and 4:*

1. *$ST_M(a) =$ sk holds iff the argument, a satisfies either of the (i) and (ii) in each lower module of M.*
   (i) a is an element of RAF-σ-grounded extension.
   (ii) a doesn't exist.

2. $ST_M(a) = def$ *holds iff the the argument, a satisfies the following (i) and (ii) in at least one lower module of M.*
   (i) a is not an element of RAF-σ-complete extension.
   (ii) a is labelled ***out*** *in at least one RAF-σ-complete labelling.*
3. $ST_M(a) = unc$ *holds iff the argument, a satisfies the following (i) and (ii) in at each lower module of M.*
   (i) a is not an element of RAF-σ-complete extension.
   (ii) a is labelled ***undec*** *in every RAF-σ-complete labelling.*
4. $ST_M(a) = cr$ *holds the argument a satisfies none of (1) or (2) or (3).*

– The Interrelation between MRAF semantics and AF semantics

Let a MRAF be Γ = (MO, *refers*), where $MO = \{M_i, M_j\}$ and *refers* = $\{(M_i, M_j)\}$. Let M_i be Δ_i = (Ar_i, $attacks_i$, ST_i), and let M_j be Δ_j = (Ar_j, $attacks_j$, ST_j). We express $(Ar_i \cap Ar_j)$ as Ar_{ij}. Let an AF be A_0 = (Ar_0, $attacks_0$), where $Ar_0 = (Ar_i \cup Ar_j)$ and $attacks_0 = (attacks_i \cup attacks_j)$ holds. Then, we can detect one interrelation between the MRAF semantics in M_i and the AF semantics in A_0 as described in the following proporsion1 iff the followings holds:

$$attacks_i \cap attacks_j = \emptyset, \text{ and } {}^{\forall}(x,y) \in attacks_0 : [x \notin (Ar_i \cap Ar_j) \text{ or } y \notin (Ar_i \cap Ar_j)]$$

Proposition 1. *If the set S which satisfying $Ar_{ij} \subseteq S \subseteq Ar_0$ is complete extension of A_0, then, the set $(S \cap Ar_i)$ is RAF-Non-Def complete extension of Δ_i.*

The RAF-Non-Def complete extension of Δ_0 (Δ_0 = $(Ar_0, attacks_0, ST_0)$), where every argument is ranked as *sk* ($ST_0(x) = sk$ holds for $\forall x \in Ar_0$) is the complete extension of A_0, too. Then, proposition 1 and the following (a) are equivalent. Furthermore, the contraposition of proposition 1, (b) is valid, too.

(a) If the set S satisfying $Ar_{ij} \subseteq S \subseteq Ar_0$ is RAF-Non-Def complete extension of Δ_0, then, the set $(S \cap Ar_i)$ is RAF-Non-Def complete extension of Δ_i.
(b) Let $S' \in Ar_i$ be a set which isn't RAF-Non-Def complete extension of Δ_i, and let $S'' \in Ar_i$ be a superset of S'. Then, S' and S'' are not complete extension of A_0.

Note that the above interrelation (b) is useful, when A_0 is too huge to calculate complete extension. We can convert it into a MRAF consisting of two modules. Hence, the interrelation (b) can give us the perspective which argument in A_0 is not complete extension by calculating RAF-Non-Def complete extension of Δ_i which is partially alternative to the semantics of A_0. In short, we can decrease the number of the possible candidates of the elements of complete extension of A_0. Furthermore, in proposition 1, (a), and (b), the words of "RAF-Non-Def complete extension" can be interpreted as the words of "RAF-Optimistic complete extension" because every element of RAF-Non-Def complete extensions is included in RAF-Optimistic complete extension.

5 Argumentation Support Tool Based on RAF

5.1 Overview of The Argumentation Support Tool

Our argumentation support tool is installed to the user's tablet PCs. These PCs are connected to the argumentation server, and users exchange messages (utterances) through it. When the user inputs an utterance, the server converts it into logical formulas and furthermore to AF, and calculates the semantics of the AF. By visualizing AF on the screen of PC, this tool helps users to understand logical structures of whole argumentation.

The functions of our tool are as follows.

- Value assignment to each argument
  This tool can assign values to arguments of the participants in several ways. For example, during discussion, participants may vote to one of conflicting arguments. By regarding the number of voting as value, this tool calculates VAF semantics[12].
- Visualization of argument trees
  This tool displays argument trees within nodes. Connecting the attack relations for the utterances in the argument tree can enhance readability when you want to look the entire argumentation. In addition, this can reduce the number of nodes produced due to sub arguments.
- Visualization of focused parts of AF
  This tool can focus on particular nodes(Fig. 3). AF becomes more complicated by utterance increases. With this tool, tapping the node can only display the relation from or to this node, so that the entire argumentation is made more understandable.
- Looking-ahead analysis for selecting next utterance
  Prior to entering utterances, this tool can obtain the results in progress by calculating the argumentation frameworks on a trial basis after utterances are entered. This makes it easier to formulate a strategy without making the argumentation disadvantageous due to my utterances.

We showed these functions are useful for actual discussions[18]. However, in real complex discussions which consisting of a lot of utterances, AF becomes complicated and huge, and users cannot understand whole structure well. To solve this problem, we extended our tool by attending the function to separate an AF into modules. To integrate the semantics of each module, we employed the semantics of MRAF.

Fig. 1 shows the snapshot of the main screen. The argument diagram is at the upper left, the argumentation framework is at the lower left, and the utterances log is at the right side of the screen.

5.2 System Operational Procedures

Fig. 2 shows the system operating procedures. When a user inputs an utterance in the form of argument diagram, it is immediately reflected on the argument diagram, and it is sent to the server, where it is converted into logical expressions, and then they are converted into AF. After the semantics of AF are calculated, AF and its semantics is returned to the iPad.

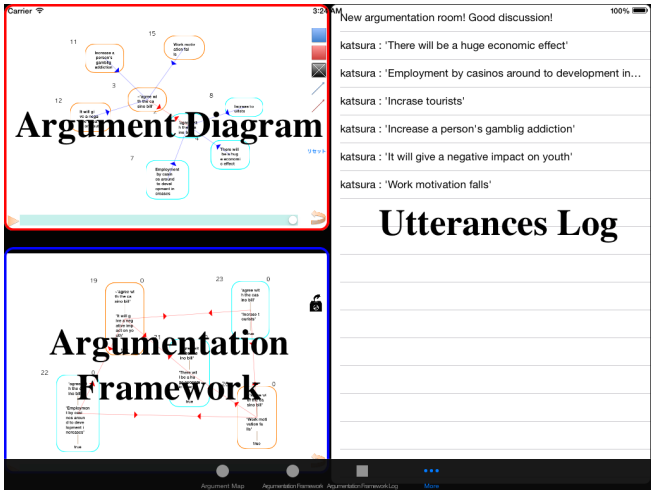


Fig. 1. Snapshot of the screen of the argumantation support tool

(A)Input Utterance in The Form of The Argument Diagram The argument diagram shows the Toulmin-like logical relationships between utterance components. This diagram saves time-series information of the argument, while accepting utterances input of user. To input the user's utterance in the form of the argument diagram, the user creates a node on the screen, makes a link from the node to the other node, and inputs his utterance in the node.

Definition 10. *The argument diagram is a tuple,* $(A, attacks, supports)$ *, where, A indicates a set of utterances, while attacks and supports indicates binary relations between utterances in A. A corresponds to nodes on the screen, while attacks and supports indicates two kinds of links between nodes.*

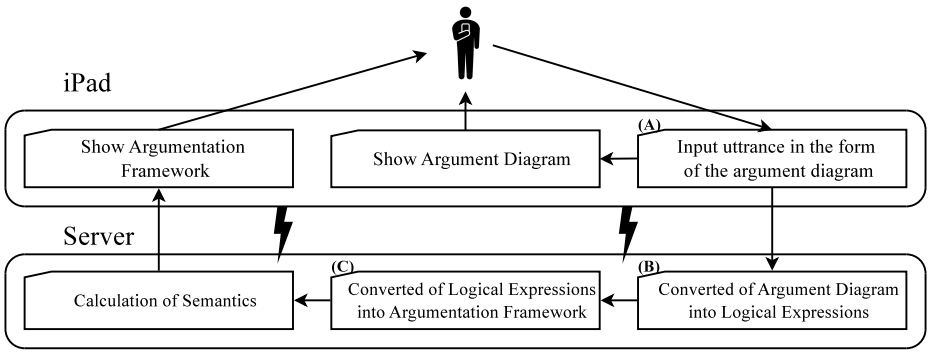


Fig. 2. Operating diagram of this system

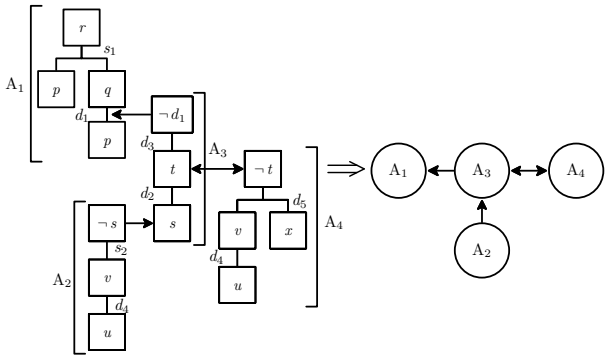


Fig. 3. Conversion of Logical Expressions into Argumentation Frameworks

(B)Conversion of Argument Diagram into Logical Expressions The argument diagram is sent to the argumentation server, where it is converted into the formula. The formula is classified to R_s(A set of defeasible inference rules), R_d(A set of defeasible inference rules), K_p(A set of the ordinary premises) and K_n(A set of the axioms).

Let an argument diagram be $(A, attacks, supports)$. For any member of attacks and supports, R_d and K_p are obtained as follows.

- $\forall a,b \in A,\ (b,a) \in attacks$ $\Longrightarrow \{\neg a \Leftarrow b\} \in R_d,\ \{b\} \in K_p$
- $\forall a,b \in A,\ (b,a) \in supports$ $\Longrightarrow \{a \Leftarrow b\} \in R_d,\ \{b\} \in K_p$

(C)Conversion of Logical Expressions into Argumentation Framework Conversion is carried out according to the conversion process of Aspic+[19]. Example (1) shows a conversion example.
Example 1.

$K_p = \{s,u,x\}$ $K_n = \{p\}$ $R_s = \{s_1,s_2\}$ $R_d = \{d_1,d_2,d_3,d_4,d_5\}$
$d_1 : p \Rightarrow q$ $d_2 : s \Rightarrow t$ $d_3 : t \Rightarrow \neg d_1$ $d_4 : u \Rightarrow v$
$d_5 : v,x \Rightarrow \neg t$ $s_1 : p,q \rightarrow r$ $s_2 : v \rightarrow \neg s$

Fig. 3 shows a conversion into an a argumentation framework. First of all, create argument trees in accordance with the rules. For example, check the top rule of A_1, s_1. As for p, p is $p \in K_n$. There is no need to examine this furthermore. Next, q is Conc(d_1) = q, check d_1. As is the case with above, p is $p \in K_n$, while there is no need to examine this any further. As a result, argument A_1 is structured as is shown by the argument diagram on the left side of Fig. 3. Preparing A_1 through A_4 similarly, create the attack relations while focusing on negation symbols of each argumentation tree. Finally, the arguments and the attack relations consist of AF as is shown in the right part of Fig. 3. This is the converted AF.

5.3 Modularization of Argumentation Framework

When there are a lot of utterances and AF becomes huge, the user can separate some part of AF from original AF and makes a new lower modules.

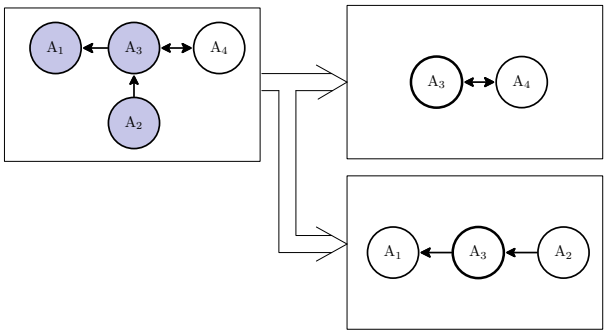


Fig. 4. Modularization of Argumentation Framework

To separate the argumentation, at first, a user selects the separation mode on the screen, then a lower module is created. After that, he selects target nodes which he wants to separate. The first selected target node is called a pivot node. The target nodes are moved to the lower module, and only the copy of the pivot node remains in the higher module. In Fig. 4, the left side shows the initial AF which consists of 4 nodes (arguments). The highlighted nodes $\{A_1, A_2, A_3\}$ are selected as target nodes, and A_3 is a pivot node. The right side of Fig. 4 shows two AFs after separating process. Target nodes $\{A_1, A_2, A_3\}$ are transferred to the lower module. The pivot node, A_3, appears in both module.

Argumentation of the lower module is continued and its semantics is sent to the higher module.

5.4 Example

We show the advantage of modularization of AF by siting an AF about the restart of nuclear power stations in Japan which has complex background. Such an AF tends to be large and complicated as illustrated in Fig. 5. In this case, we can reconstruct the graph structure of AF as the module structure of AF which is composed of the modules representing the parts of the original AF as illustrated in Fig. 6. Modules illustrated in Fig. 6 are AFs about the following subtopics of the restart of nuclear power stations in Japan.

Module 0: Main module
Module 1: Power supply
Module 2: Security
Module 3: Economy
Module 4: Stable acquisition of uranium
Module 5: Cost of energy power

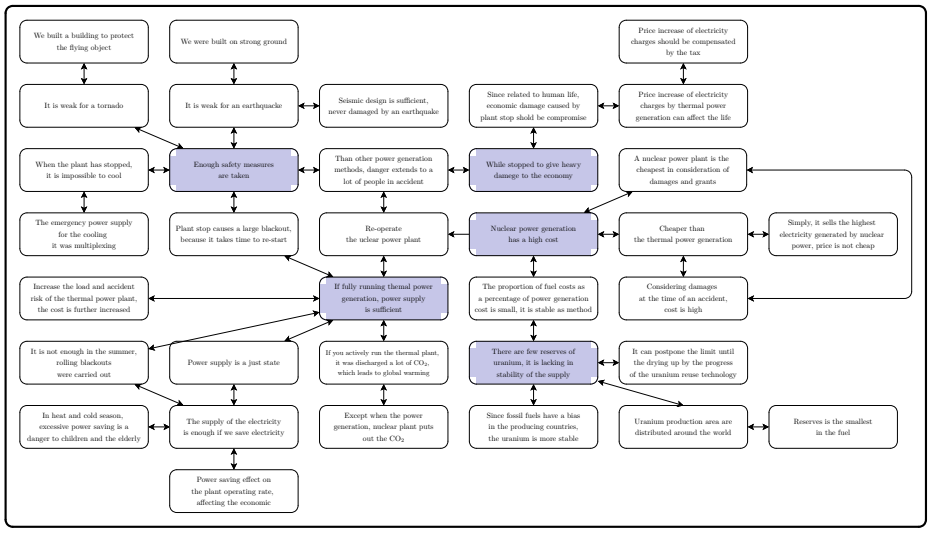


Fig. 5. AF about reopen of nuclear facility problem

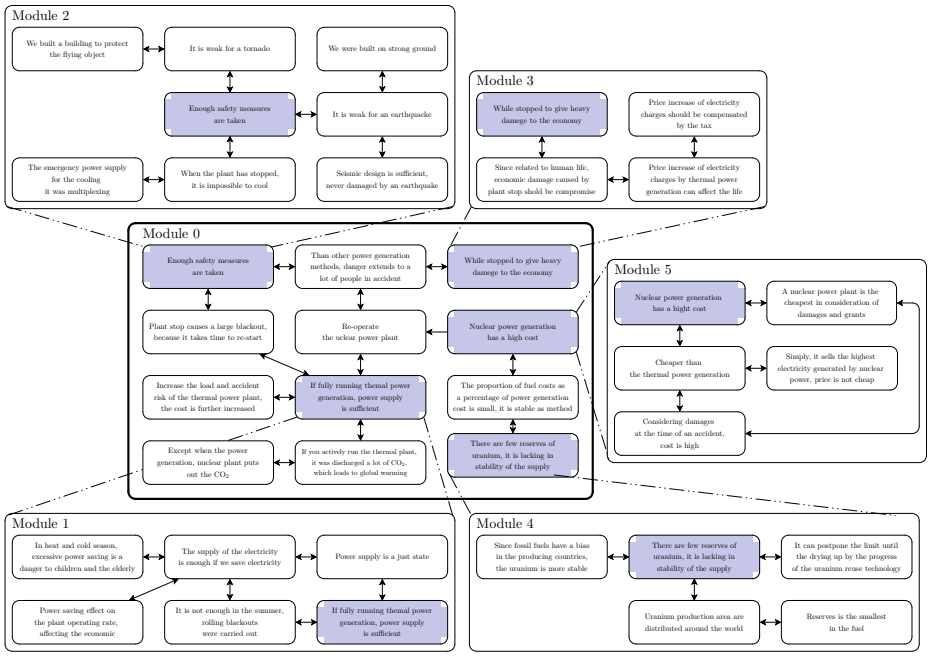


Fig. 6. Module structure of AF about reopen of nuclear facility problem

6 Conclusion

We introduced an argumentation support tool based on MRAF proposing this paper. This tool helps users by supplying environment of online argumentation, by visualizing

logical structure of arguments, and by calculating semantics of arguments based on AF or MRAF. MRAF is an extended theory of AF defined by Dung, and has a semantics which integrates more than one RAF semantics which depends on the reliability of each argument and gives theoretical foundation of MRAF. Based on MRAF theory, our argumentation support tool has a function to separate some part of AF from the original AF. This function is useful for users to grasp a logical structure of a huge argumentation and the acceptability of each argument.

References

1. Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979, 2004.
2. Toulmin Stephen. The uses of argument. *Cambridge University Press, Cam bridge*, 1958.
3. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.
4. Katie Atkinson, Trevor Bench-Capon, and Paul E DUNNE. Uniform argumentation frameworks. *Computational Models of Argument: Proceedings of COMMA 2012*, 245:165, 2012.
5. Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.
6. Leendert VANDER TORRE and Serena Villata. An aspic-based legal argumentation framework for deontic reasoning. *Computational Models of Argument: Proceedings of COMMA 2014*, 266:421, 2014.
7. Niels Pinkwart, Collin Lynch, Kevin Ashley, and Vincent Aleven. Re-evaluating largo in the classroom: Are diagrams better than text for teaching argumentation skills? *Intelligent Tutoring Systems*, pages 90–100, 2008.
8. Phan Minh Dung, Phan Minh Thang, and Nguyen Duy Hung. Modular argumentation for modelling legal doctrines of performance relief. *Argument and Computation*, 1(1):47–69, 2010.
9. Gerhard Brewka and Thomas Eiter. Argumentation context systems: A framework for abstract group argumentation. In *Logic Programming and Nonmonotonic Reasoning*, pages 44–57. Springer, 2009.
10. Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9):901–934, 2009.
11. Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 1–7. Morgan Kaufmann Publishers Inc., 1998.
12. Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
13. Yuqing Tang, Kai Cai, Peter McBurney, Elizabeth Sklar, and Simon Parsons. Using argumentation to reason about trust and belief. *Journal of Logic and Computation*, page exr038, 2011.
14. Hengfei Li, Nir Oren, and Timothy J Norman. Probabilistic argumentation frameworks. In *Theorie and Applications of Formal Argumentation*, pages 1–16. Springer, 2012.
15. Andrea Pazienza, Floriana Esposito, and Stefano Ferilli. An authority degree-based evaluation strategy for abstract argumentation frameworks.
16. Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218. Springer, 2009.

17. Martin Caminada. On the issue of reinstatement in argumentation. *Logics in artificial intelligence*, pages 111–123, 2006.
18. Yuki Katsura, Shogo Okada, and Katsumi Nitta. Dynamic argumentaion support tool using argument diagram. *Annual Conference of the Japanese Society for Artificial Intelligence*, 29:1–4, 2015. in Japanese.
19. H. Prakken. An overview of formal models of argumentation and their application in philosophy. *Studies in logic*, 4(1):65–86, 2011.

The ProLeMAS project: representing natural language norms in Input/Output logic

Livio Robaldo*, Llio Humphreys*, Xin Sun*, Loredana Cupi+, Cristiana Teixeira Santos#, and Robert Muthuri#

*University of Luxembourg
{livio.robaldo, llio.humphreys, xin.sun}@uni.lu
+University of Turin, loredana.cupi@unito.it
#Joint International Doctoral (Ph.D.) Degree in Law, Science and Technology
muthuri.r@gmail.com, cristiana.teixeirasantos@gmail.com ⋆

Abstract. This paper represents the first step of the Marie Sklodowska-Curie ProLeMAS research project, which aims to (1) fill the gap between the current logical formalizations of legal text, mostly propositional, and the richness of natural language semantics, and (2) implement a concrete NLP pipeline for automatically building formulae from existing legal documents. In this paper, drawing from insights of Input/Output logic and the Reification-based approach of Jerry R. Hobbs, we propose a first-order logic formalization of real obligations.

1 Introduction

Ontologies of legal norms such as those used in Eunomos [4] [2] and Legal URN [7] present norms in a structured, searchable and intuitive format for representing who may or must do what and when, which can be used by legal professionals for regulatory compliance and legal requirement engineering.

Automated or semi-automated methods for ontology building are necessary if we are to over-come the so-called 'resource consumption bottleneck' [8] of creating ontologies manually, which has contributed to the current lack of practical non-toy ontologies available (cf. [11]).

Legal text is an under-researched area in Information Extraction, and there is a lack of suitable annotated data. The idiosyncratic nature of legislative text poses particular challenges and advantages for the task of extracting such information using NLP, in order to associate norms with suitable semantic representations on which to perform automatic reasoning.

⋆ Livio Robaldo is supported by the Marie Sklodowska-Curie Individual project "ProLeMAS: PROcessing LEgal language in normative Multi-Agent Systems", University of Luxembourg. Llio Humphreys is supported by the National Research Fund, Luxembourg. Loredana Cupi is supported by the ITxLaw project funded by Compagnia di San Paolo. Cristiana Teixeira Santos and Robert Muthuri are supported by the Joint International Doctoral (Ph.D.) Degree in Law, Science and Technology (Last-JD). We would like to thank prof. Leon van der Torre for fruitful discussions.

Our ongoing work on extraction of legal information to populate legal ontologies addresses the following research questions:

(1) a. How to design suitable knowledge representations for legal language?
b. How to build an NLP ontology population tool for transforming legal language into such a representation?

The Marie Sklodowska-Curie ProLeMAS project[1] specifically addresses research questions (1.a-b). Drawing from the results of the ITxLaw project, ProLeMAS (Processing Legal Language for Normative Multi-Agent Systems) proposes to fill the gap between current logical frameworks designed to represent legal knowledge and the richness of NL semantics by using Jerry R. Hobbs's logic [10] for the semantic representation of legal text (research question (1.a)) and to implement a concrete NLP pipeline for automatically building formulae from existing legal documents (research question (1.b)).

The present paper represents the first attempt in the context of the ProLeMAS project, and it specifically addresses research question (1.a).

In Section 2 we present the Eunomos system and the corpus of European Directives where we are going to develop our research. Section 3 and Section 4 introduce the formal instruments at the base of our logical formalization: Hobbs's logic and Input/Output logic. The latter appears as one of the new achievements in deontic logic in recent years. Section 5 contains the central part of the paper; it defines a new logic that we call "The ProLeMAS logic", which intermediates between Hobbs's logic and Input/Output logic, and shows how it can be used to represent the meaning of obligations. Section 6 concludes the paper.

2 Background - The Eunomos system

The Eunomos legal Knowledge Management System [4], developed by the University of Turin as part of the ICT4LAW[2] project, enables customers to find and understand the law in their area of interest by searching a database of national and regional legislation, and using a multi-level ontology to find the precise definition for terms within a particular context. The ontology has been extended to include a particular structure for organising information around individual prescriptions. The user-focused ontology contains explanations in natural language, as well as key points from relevant cases, for which the input and interpretation of a specialist legal expert remains crucial. But, as it is well-known since the nineties, see, for instance, [21], there are other elements within the legal text that could be extracted by automated means, such as *Deontic clause* (i.e., the type of the prescription: obligation, prohibition, permission, exception, etc.), *Active role* (i.e., the addressee of the norm, such as the citizen, the director, etc.), *Passive role* (i.e., the beneficiary of the norm.), *Sanction* (i.e., the sanction resulting from the violation), etc. (see also [3]).

[1] http://www.liviorobaldo.com/ProLeMAS.htm
[2] http://www.ict4law.org

To capture the full richness of legal language in the way it expresses norms, we started by studying a corpus of EU legislation in English on a variety of topics. We identified which sections contained norms and definitions, and which sections contained legislative procedural details which are not relevant to our considerations. We identified norms and norm elements, refining concepts from legal theory and linking them to the linguistic variety using the Eunomos ontology framework as our starting point but with a view to extending the ontology to reflect the realities of the corpus - norm types such as powers, and norm elements such as the purpose of a norm, conditions, timeframe, etc.

Our preprocessing module uses pattern-matching and the Python NLTK Brill part-of-speech tagger to perform the following tasks:

- Sentence segmentation (identification of abbreviations).
- Identify titles (uppercase or lacking verb).
- Identify and transform references into processable units.
- Transform certain words to enable Semantic Role Labeling handling.
- Identify lists and transform the lists into proper grammatical sentences, by adding common beginnings and endings to each list item.

We use for our experiments a corpus of twenty European Directives from 1998 to 2011, covering a range of subjects: the profession of lawyer, passenger ships, biotechnological inventions, feedingstuffs from third countries, electronic signatures, seafarers' hours of work, etc.

Our initial experiments of norm extraction and representation in ProLeMAS logic is conducted on the English version of *Directive 98/5/EC of the European Parliament and of the Council of 16 February 1998 to facilitate practice of the profession of lawyer on a permanent basis in a Member State other than that in which the qualification was obtained.* We have excluded the preamble and annexes, and discarded 3 sentences with corrupted output from preprocessing the text to enable easier NLP handling. There are 36 obligations, 13 powers, 10 legal effects, 8 definitions, 6 permissions, 6 applicability types, 6 rationales, 2 rights, 1 exception, and 1 hierarchy. In this paper, we use the following example of obligation for explanatory purposes:

(2) *A lawyer who wishes to practise in a Member State other than that in which he obtained his professional qualification shall register with the competent authority in that State.*

We examined all obligations in the above-mentioned directive and we did not find relevant differences from the perspective of the logical architecture of their meaning. There are complex issues to solve from the point of view of the NLP pipeline in order to properly automatically build the formulae from text, e.g. anaphora resolution. But those issues are outside the scope of the present paper.

3 Hobbs' logical framework

Jerry R. Hobbs defines a wide-coverage logic for Natural Language semantics centered on the notion of Reification (see [10] and several other earlier publi-

cations by the same author[3]). Reification is a concept originally introduced by Donald Davidson in [5]. Modern logical frameworks based on Reification are known in the literature as "neo-Davidsonian" approaches.

Reification allows a wide variety of complex natural language (NL) statements to be expressed in First Order Logic (FOL). NL statements are formalized such that events, states, etc., correspond to constants or quantifiable variables of the logic. In other words, the states and events denoted by these constants as well as the variables are things in the world. Hobbs uses the term 'eventuality' to denote the reification of both a state or an event.

Hobbs' implements a fairly large set of linguistic and semantic concepts including sets, composite entities, scales, change, causality, time, event structure, etc., into an integrated first order logical formalism.

Hobbs distinguishes two parallel sets of predicates: primed and unprimed. The unprimed predicates are standard first order predicates commonly used in logical representations. For example, $(give\ a\ b\ c)$ asserts that a gives b to c in the real world. The primed predicate represents the reification of the corresponding unprimed relation. The expression $(give'\ e\ a\ b\ c)$ says that e is a giving event by a of b to c. Eventualities may be *possible* or *actual*. In Hobbs', this distinction is represented via a unary predicate *Rexist* that holds for eventualities really existing in the world. To give an example cited in Hobbs, if I want to fly, my wanting really exists, but my flying does not. This is represented as:

$$\exists_e[\ (Rexist\ e) \wedge (want'\ e\ \texttt{I}\ e_1) \wedge (fly'\ e_1\ \texttt{I})\ ]$$

Eventualities can be treated as the objects of human thoughts. Reified eventualities are inserted as parameters of such predicates as *believe*, *think*, *want*, etc. Reification can be applied recursively. The fact that *John believes that Jack wants to eat an ice cream* is represented as an eventuality e such that it holds:

$$\exists_e\exists_{e_1}\exists_{e_2}\exists_{e_3}[\ (Rexist\ e) \wedge (believe'\ e\ \texttt{John}\ e_1) \wedge (want'\ e_1\ \texttt{Jack}\ e_2) \wedge (eat'\ e_2\ \texttt{Jack Ic}) \wedge (iceCream'\ e_3\ \texttt{Ic})\ ]$$

Every relation on eventualities, including logical operators, causal and temporal relations, and even tense and aspect, may be reified into another eventuality. For instance, by asserting $(imply'\ e\ e_1\ e_2)$, we reify the implication from e_1 to e_2 into an eventuality e and e is, then, thought of as "the state holding between e_1 and e_2 such that whenever e_1 really exists, e_2 really exists too". Negation is represented as $(not'\ e_1\ e_2)$: e_1 is the eventuality of e_2's not existing.

The predicates *imply'* and *not'* are defined to model the concept of 'inconsistency'. In the next subsection, we show how this concept can be used to construct a uniform account of 'Direct' and 'Indirect' Contrast.

Two eventualities e_1 and e_2 are said to be inconsistent if and only if they (respectively) imply two other eventualities e_3 and e_4 such that e_3 is the negation of e_4. The definition is as follows:

[3] See http://www.isi.edu/~hobbs/csknowledge-references/csknowledge-references.html and http://www.isi.edu/~hobbs/csk.html.

(3) $(forall\ (e_1\ e_2)$
$(iff\ (inconsistent\ e_1\ e_2)$
$(and\ (eventuality\ e_1)\ (eventuality\ e_2)$
$(exists\ (e_3\ e_4)\ (and\ (imply\ e_1\ e_3)$
$(imply\ e_2\ e_4)(not'\ e_3\ e_4))))))$

(3) is an example of 'axiom schema'. In this logic, an 'axiom *schema*' provides one or more different axioms for each predicate p. The axiom schemas of the predicates used in formulae are generally stored into a separate ontology.

Higher order operators, such as modal and temporal operators, are modelled by introducing new predicates and by defining axiom schemas to restrict their meaning. However, it is important to note that thanks to reification, the formulae representing natural language utterances *never* feature any kind of embedding of predicates within other operators. In other words, formalae are *always* conjunction of atomic FOL predicates applied to FOL terms. See for instance [18], which propose a formalization in Hobbs' of concessive relations, one of the most trickiest relations occurring in natural language semantics.

It should be clear that the main peculiarity of Reification-based logical frameworks is their formal simplicity. This allow to easily deal with several natural language phenomena. Hobbs' past research is particular to address a proper treatment of anaphora (cf. in particular [10]). For instance, (4.a) may be represented via formula (4.b). For simplicity, in (4.b) the semantic relation between the two clauses is simply represented as a material implication (predicate $imply'$).

(4) a. If John goes to Mary's house, he tells her before.

b. $(Rexist\ e) \wedge (imply'\ e\ e_1\ e_2) \wedge (goTo'\ e_1\ \texttt{J}\ \texttt{M}) \wedge$
$(tell'\ e_2\ \texttt{J}\ e_1\ \texttt{M}) \wedge (happenBefore\ e_2\ e_1)$

e_1 is the event of John's going to Mary, while e_2 is the event of John's telling to Mary *the fact that he will come to her*. The predicate $(happenBefore\ e_2\ e_1)$ states that e_2 must occur before e_1. Note that e_1 and e_2 are only hypothetical eventualies: (4.b) does not assert they exist in the real world. In (4.a), the (hidden) pronoun "it", which is the object of the verb "tell", is straightforwardly represented: the eventuality e_1 is directly inserted as the second parameter of the $tell'$ predicate in (4.b). Without Reification, it would be necessary to introduce in the logic some 2-order operators in order to get the intended meaning of (4.a).

3.1 ProLeMAS object logic

Hobbs' formula are formulae in first order logic (FOL, henceforth) with a very restricted syntax. They are basically *conjunctions* of atomic predicates instantiated on FOL terms. From a formal point of view, eventualities are FOL terms exactly as "`Jack`" in example (4.b), the only difference being that they refers to facts and actions occurring in the world. Facts and actions are taken to be individuals of the domain like persons, dogs, etc.

The logic we are going to use as the object logic of I/O systems - which we will call "ProLeMAS object logic" - is a further simplification of Hobbs' logic. The formulae will be more verbose, but, in our view, the simpler syntax will enhance readability and it will facilitate the definition of a reference ontology storing the available predicates and the axiom schemas modelling their meaning.

In fact, it is easy to see that the structure of the formulae strictly resemble the technique of rewriting relations of arbitrary arity as binary relations, used in AI as entity-attribute-value (EAV) triples in the last decades, which is at the basis of the subject-predicate-object of the RDF/OWL data model[4].

In ProLeMAS object logic, there is a single type of predicates, i.e. there is not distinction between primed and unprimed predicates. Predicates will be always unary or binary predicates. Thus, for instance, "$(give\ a\ b\ c)$" is not an acceptable predicate in our logic. N-ary relations are modelled by making thematic roles explicit. This is done by introducing other FOL predicates referring each to an available thematic role. For example, "$(give\ a\ b\ c)$" is translated into:

$$(give\ e_1) \wedge (agent\ e_1\ a) \wedge (patient\ e_1\ b) \wedge (recipient\ e_1\ c) \tag{5}$$

The meaning of (5) is obvious: e_1 is a giving event whose agent is a, whose patient is b, and whose recipient is c. "Agent", "patient", and "recipient" are thematic roles. The ontology specifies, for each kind of eventuality, the available thematic roles and - via further axioms - the restrictions on these thematic roles.

For instance, if we want to impose that agents of giving eventualities can be only human beings, we add the following axiom to the ontology:

$$(forall\ (e\ a)\ (if\ (and\ (give\ e)\ (agent\ e\ a))\ (humanBeing\ a))) \tag{6}$$

Of course, the computational ontology has to be designed and developed with respect to the application and the domain where we want to concretely use the formulae. In our future research, we aim at specifically designing and implementing a legal ontology for modelling the meaning of norms expressed in natural language [1].

We now formally defines[5] the syntax of ProLeMAS object logic. For reasons that will be clear below, ProLeMAS object logic includes existential quantifiers but not universal ones. And, free variables are allowed. Those will be bound by an (external) universal quantifier.

[4] http://www.w3.org/TR/owl-ref

[5] Definition 1 only includes the boolean connective "$\wedge$". Other boolean connectives are modelled by introducing special predicates as $imply'$ and not' in (3).

Definition 1 (Syntax of ProLeMAS object logic). *ProLeMAS object logic is a fragment of First Order Logic (FOL). Where:*

- *The vocabulary includes FOL terms (constants, variables, and functions), FOL unary or binary predicates, the boolean connective "∧" and the existential quantifier "∃".*
- *If "p" and "q" are, respectively, a unary and a binary predicate, while "a" and "a" are terms, "p(a)" and "q(a,b)" are atomic formulae.*
- *If $\Phi_1 \dots \Phi_n$ are atomic formulas, "$\Phi_1 \wedge \dots \wedge \Phi_n$" and "$\exists_{x_1} \dots \exists_{x_m} [\Phi_1 \wedge \dots \wedge \Phi_n]$", where $x_1 \dots x_m$ occurring in $\Phi_1 \dots \Phi_n$, are non-atomic formulas, possibly containing free variables.*

4 Input/Output logic

Input/Output logic (I/O logic) has been originally introduced by Makinson and van der Torre in [14]. For a comprehensive survey and a techinqual introduction of I/O logic, see [17] and [19] respectively. Strictly speaking, I/O logic is not a single logic but a family of logics, just like modal logic is a family of logics containing systems K, KD, S4, S5, etc. In the first volume of the handbook of deontic logic and normative systems [6], I/O logic appears as one of the new achievements in deontic logic in recent years.

I/O logic takes its origin in the study of conditional norms. Unlike modal logic, which usually uses possible world semantics, I/O logic mainly adopts operational semantics: an I/O system is conceived as a deductive machine, like a black box which produces deontic statements as output, when we feed it factual statements as input. In the original paper of Makinson and van der Torre, i.e. [14], four I/O logics are defined: $\mathtt{out}_1$, $\mathtt{out}_2$, $\mathtt{out}_3$, and $\mathtt{out}_4$. They vary on the axioms used to constrain the deductive machine that produces the output against a valid input.

Let $\mathbb{P} = \{p_0, p_1, \dots\}$ be a countable set of propositional letters and $\mathtt{L}$ be the propositional language built upon $\mathbb{P}$. Let $\mathtt{G} \subseteq \mathtt{L} \times \mathtt{L}$ be a set of ordered pairs of formulas of $\mathtt{L}$. $\mathtt{G}$ represents the deduction machine of the I/O logic: whenever one of the heads is given in input, the corresponding tails are given in output. Each pair (a, b) in $\mathtt{G}$ is called a "generator" and it is read as "given a, it ought to be x". In this paper, "a" and "b" are respectively termed as the *head* and the *tail* of the generator (a, b). Formally, $\mathtt{G}$ is a function from $2^{\mathtt{L}}$ to $2^{\mathtt{L}}$ such that for a set $\mathtt{A}$ of formulas, $\mathtt{G}(\mathtt{A}) = \{x \in \mathtt{L} : (a, x) \in \mathtt{G} \text{ for some } a \in \mathtt{A}\}$. Makison and van der Torre [14] define the semantics of I/O logics from $\mathtt{out}_1$ to $\mathtt{out}_4$ as follows:

(7)
- $\mathtt{out}_1(\mathtt{G},\mathtt{A}) = Cn(\mathtt{G}(Cn(A)))$.
- $\mathtt{out}_2(\mathtt{G},\mathtt{A}) = \bigcap\{Cn(\mathtt{G}(\mathtt{V})) : \mathtt{A} \subseteq \mathtt{V}, \mathtt{V} \text{ is complete}\}$.
- $\mathtt{out}_3(\mathtt{G},\mathtt{A}) = \bigcap\{Cn(\mathtt{G}(\mathtt{B})) : \mathtt{A} \subseteq \mathtt{B} = Cn(\mathtt{B}) \supseteq \mathtt{G}(\mathtt{B})\}$.
- $\mathtt{out}_4(\mathtt{G},\mathtt{A}) = \bigcap\{Cn(\mathtt{G}(\mathtt{V}) : \mathtt{A} \subseteq \mathtt{V} \supseteq \mathtt{G}(\mathtt{V})), \mathtt{V} \text{ is complete}\}$.

Here Cn is the classical consequence operator of propositional logic, and a set of formulas is *complete* if it is either *maximal consistent* or equal to $\mathtt{L}$.

These four logics are called *simple-minded output*, *basic output*, *simple-minded reusable output* and *basic reusable output* respectively. For each of these four logics, a *throughput* version that allows inputs to reappear as outputs, defined as $\mathtt{out}_i^+(\mathtt{G},\mathtt{A}) = \mathtt{out}_i(\mathtt{G}_{id},\mathtt{A})$, where $\mathtt{G}_{id} = \mathtt{G} \cup \{(a,a) \mid a \in \mathtt{L}\}$. When A is a singleton, we write $\mathtt{out}_i(\mathtt{G},a)$ for $\mathtt{out}_i(\mathtt{G},\{a\})$.

I/O logics are given a proof theoretic characterization. We say that an ordered pair of formulas is derivable from a set G iff (a,x) is in the least set that extends $\mathtt{G} \cup \{(\top,\top)\}$ and is closed under a number of derivation rules. The following are the rules we need to define $\mathtt{out}_1$ to ${\mathtt{out}_4}^+$:

(8)
- SI (strengthening the input): from (a,x) to (b,x) whenever[6] $b \vdash a$.
- OR (disjunction of input): from (a,x) and (b,x) to $(a \vee b,x)$.
- WO (weakening the output): from (a,x) to (a,y) whenever $x \vdash y$.
- AND (conjunction of output): from (a,x) and (a,y) to $(a,x \wedge y)$.
- CT (cumulative transitivity): from (a,x) and $(a \wedge x,y)$ to (a,y).
- ID (identity): from nothing to (a,a).

The derivation system based on the rules SI, WO and AND is called $\mathtt{deriv}_1$. Adding OR to $\mathtt{deriv}_1$ gives $\mathtt{deriv}_2$. Adding CT to $\mathtt{deriv}_1$ gives $\mathtt{deriv}_3$. The five rules together give $\mathtt{deriv}_4$. Adding ID to $\mathtt{deriv}_i$ gives $\mathtt{deriv}_i^+$ for $i \in \{1,2,3,4\}$. $(a,x) \in \mathtt{deriv}_i(\mathtt{G})$ is used to denote the norms (a,x) derivable from G using rules of derivation system $\mathtt{deriv}_i$. In [14], it is proven that each $\mathtt{deriv}_i^{(+)}$ is sound and complete with respect to $\mathtt{out}_i^{(+)}$.

4.1 I/O logics as a general framework for normative reasoning

I/O logic is a general framework for normative reasoning, used to formalize and reason about the detachment of obligations, permissions and institutional facts from conditional norms. I/O logic is not defined in terms of a *truth-conditional* semantics. Rather, as pointed out above, I/O logic adopts *operational* semantics. As explained in [12] and [13], "directives *do not* carry truth-values. Only declarative statements may bear truth-values, but norms are items of another kind. They may be respected (or not), and may also be assessed from the standpoint of other norms, for example when a legal norm is judged from a moral point of view (or viceversa). But it makes no sense to describe norms as true or as false."

Thus, I/O logic allows to determine which obligations are operative in a situation that already violates some of them. To achieve this, we must look at the family of all maximal subsets G' such that $\mathtt{G'} \subseteq \mathtt{G}$ and $\mathtt{out}_i(\mathtt{G'},\mathtt{A})$ is consistent with A. The family of such $\mathtt{out}_i(\mathtt{G'},\mathtt{A})$ is called the outfamily of (G, A).

To understand this concept, consider the following example. Suppose we have the following two norms: "The cottage should not have a fence or a dog" and "if it has a dog it must have both a fence and a warning sign." that we may formalize as the I/O logic generators "$(\top, \neg(f \vee d))$" and "$(d,\ f \wedge w)$" respectively.

[6] Here $\vdash$ is the classical entailment relation of propositional logic.

Suppose further that we are in the situation that the cottage has a dog. In this context, the first norm is violated. And, the `outfamily` of (`G`, `A`) determines[7] that, still, we are obliged to build a fence with a warning sign around the cottage:

$$\texttt{G} \equiv \{(\top, \neg(f \vee d)), (d, f \wedge w)\}, \texttt{A} \equiv \{d\}, \texttt{outfamily}(\texttt{G}, \texttt{A}) \equiv \{Cn(f \wedge w)\}$$

It is not possible to achieve such a characterization of norms in terms of a truth-conditional semantics: a violation would correspond to an inconsistency, which will make inconsistent the whole knowledge base.

Although I/O logic is an adequate framework for representing and reasoning on norms, so far the objects logic used for asserting the generators and the input have been only *propositional*. The reason are issues related to the complexity of the framework. The complexity of input/output logic is at least as difficucut as the objective logic. By the complexity of input/output logic, we mean the complexity of the following fulfillment problem:

$$\text{Given finite } G, A \text{ and } x, \text{ is } x \in out_i(G, A)?$$

[20] shows that the complexity of the fulfillment problem for $\texttt{out}_1$,$\texttt{out}_2$,$\texttt{out}_4$ is `coNP` complete, for $\texttt{out}_3$ the lower bound is `coNP` while the upper bound is $\texttt{P}^{\texttt{NP}}$.

No efforts have been done yet to represent norms coming from *existing* legal texts as the one of the corpus described above in section 2. For representing concrete existing norms, the expressivity of propositional logic is not sufficient.

First order (object) logics are needed in order to fill the gap between the I/O logic and the richness of NL semantics. To this end, in the next section, we propose to merge the ProLeMAS Object Logic as defined in section 3.1 with I/O logic. There is no precedent in the literature of a first-order I/O logic. However, it is worth noticing that this proposal is not the first deontic logic employing first-order relational variables.

A very close approach is McCarty's Language for Legal Discourse (LLD) [15] and [16]. The main difference with respect to our approach is that LLD allows for *embedded* sub-formulae (i.e., embedded Horn clauses, in McCarthy's formalization), while the key point of Hobbs' proposal, on which the ProLeMAS Object Logic is drawn, is the *total* avoidance of embeddings. As explained in section 3, all formulae are conjoined atomic predications.

5 The ProLeMAS logic

The present section merges together I/O logic and the ProLeMAS object logic, whose syntax has been defined above in definition 1. The resulting logic will be termed as "the ProLeMAS logic".

We are not interested here in proposing first order versions of *all* definitions shown in the previous section, but we will restrict our attention to only those needed for the aims of the ProLeMAS project. In particular, the axiom `OR` in (8) does not appear to be suitable for legal reasoning. To see why consider the following obligations: "If someone kills a dog, s/he has to spend two years in

[7] In this example, $\texttt{outfamily}(\texttt{G}, \texttt{A}) \equiv \{Cn(f \wedge w)\}$ for all I/O logic $\texttt{out}_i^{(+)}$, with i=1,2,3,4.

prison" and "If someone robs a bank s/he has to spend two years in prison". And, suppose John did one of the two, but there is no way to understand *which* one, i.e. if either he killed a dog or he robbed a bank. Logically, John must spend two years in prison. But on the perspective of legal reasoning, he must not: only if concrete evidence of what he did is found, obligations apply. Thus, in the rest of the paper we will no longer consider the `OR` axiom and, consequently, the I/O logic $\texttt{out}_2$ and $\texttt{out}_4$. On the other hand, we will focus in particular on the `CT` (cumulative transitivity) axiom, used to define the *simple-minded reusable output* logic $\texttt{out}_3$. It will be easy to apply the considerations below also to the remaining axioms `SI`, `WO`, `AND`, and `ID`, so that we will skip formal definitions about them.

From a formal point of view, recalling that the syntax of the ProLeMAS object logic admits free variables, the last ingredient needed to merge together ProLeMAS object logic and $\texttt{out}_3$ are quantifiers bounding each a free variable. We impose these free variables to occur both in the head and the tail of a generator, and we bound them via universal quantifiers. This establishes a "brigde" between the head and the tail, needed to "carry" individuals from the input to the output. Consider these simple (toy) examples:

(9) a. Each lawyer *must* run.
b. A lawyer who runs *must* wear a pair of shoes.
c. If John goes to Mary's house, he'll *have to* tell her before.

We propose to represent (9.a-c) via the following generators:

(10) a. $\forall_x(\ lawyer(x),\ \exists_{e_r}[(Rexist\ e_r) \wedge run(e_r) \wedge agent(e_r,\ x)]\ )$

b. $\forall_x \forall_{e_r}(\ lawyer(x) \wedge (Rexist\ e_r) \wedge run(e_r) \wedge agent(e_r,\ x),$
$\exists_{e_w} \exists_y [(Rexist\ e_w) \wedge wear(e_w) \wedge agent(e_w,\ x) \wedge$
$patient(e_w,\ y) \wedge shoes(y)]\ )$

c. $\forall_{e_g}(\ (Rexist\ e_g) \wedge go(e_g) \wedge agent(e_g,\ \texttt{J}) \wedge to(e_g,\ \texttt{M}),$
$\exists_{e_t}[(Rexist\ e_t) \wedge tell(e_t) \wedge agent(e_t,\ \texttt{J}) \wedge receiver(e_t,\ \texttt{M}) \wedge$
$theme(e_t,\ e_g) \wedge (happenBefore\ e_t\ e_g)]\ )$

Note that, in (10.b), x occurs in *both* the head *and* the tail of the generator, while e_r *only* occurs in the head. On the other hand, y and e_w are existentially quantified variables that occur *only* in the tail: every time a lawyer runs, there is a different "wearing" eventuality and (possibly) a different pair of shoes.

On the other hand, in (10.c), the eventuality e_g occurs in both the head and the tail. That's because the sentence means: "If John goes to Mary's house, he'll have to tell *Mary* before *that he goes to Mary*".

In our solution, free variables occurring in the heads (and possibly *also* in the tails) are outscoped by universal quantifiers. Free variables occurring *only* in the tails are outscoped by the existential quantifiers of the ProLeMAS object logic syntax (cf. definition 1). Formally:

Definition 2 (ProLeMAS logic Generators).
A generator in ProLeMAS logic is a construct in the form:

$$\forall_{x_1,\ldots,x_n,y_1,\ldots,y_m}(\ \Phi(x_1,\ldots,x_n,\ y_1,\ldots,y_m),\ \Psi(y_1,\ldots,y_m)\ ) \in \mathtt{G}$$

where $x_1,\ldots,x_n$ are free variables occurring only in Φ while $y_1,\ldots,y_n$ occur both in Φ and in Ψ. Φ and Ψ do not contain any other free variable. Furthermore, Φ does not contain existential quantifiers.

Representing sentence 2 in ProLeMAS logic is straightforward: we simply increase the *size* of the formula, but not its *complexity*. The formula is:

$$\forall_x \forall_y \forall_{e_w} \forall_{e_p}(\ cond(x,\ y,\ e_w,\ e_p),\ action(x,\ y)\ )$$

where:

$$cond(x,\ y,\ e_w,\ e_p) \Leftrightarrow lawyer(x) \wedge memberState(y) \wedge different(y,\ f_w(x)) \wedge Rexist(e_w) \wedge want(e_w) \wedge practise(e_p) \wedge agent(e_w,\ x) \wedge patient(e_w,\ e_p) \wedge agent(e_p,\ x) \wedge at(e_p,\ y)$$

$$action(x,\ y) \Leftrightarrow \exists_{e_r}[\ Rexist(e_r) \wedge register(e_r) \wedge agent(e_r,\ x) \wedge patient(e_r,\ x) \wedge with(e_r,\ f_c(y))\ ]$$

where x, y, e_w, e_p, and e_r are variables denoting a lawyer, a Member State, and the eventualities of "wanting", "practising" and "registering". *agent*, *patient*, *at*, and *with* are thematic roles of the two eventualities. $f_w(x)$ is a function referring to the Member State where x obtained his professional qualification while $f_c(y)$ is a function that given a member state y returns the competent authority of y. The meaning of the formula is obvious: for every tuple of lawyer, Member State, and "wanting", and "practising" actions that satisfy together the predicates in *cond*, the predicates in *action* are instantiated on x and y.

5.1 Working with ProLeMAS formulae

The main peculiarity of Reification-based logical framework is their formal simplicity. By instantiating FOL predicates on non-variable FOL terms, we obtain again propositional formulae. Thus, it is easy to see that ProLeMAS's generators do not increase the complexity of the I/O logic originally defined in [14], provided that two requirements are met: (1) the domain is finite; and (2) the input formulae are only atomic formulae in ProLeMAS object logic, i.e, FOL predicates instantiated on non-variable FOL terms, i.e., propositional formulae.

The aim of the ProLeMAS project is to build concrete NLP-based applications to be used in practical applications, where both requirements (1) and (2) are met. The domains of individuals will be always finite (e.g., the set of all lawyers in the EU). And, we are interested in performing normative reasoning on specific[8] individuals only, e.g., deriving all obligations a specific lawyer must obey, according to a specific normative code.

[8] It could be the case that such applications will have to reason on *sets*, e.g., a sets of ten laywers. To properly deal with sets, Hobbs introduces in his framework the notion of *typical element* (cf. [9] and [10]).

Universal quantifiers are just a compact way to refer to all individuals in the domain. We obtain equivalent formulae by simply substituing the universally quantified variables with all constants referring each to an individual of the domain. For instance, assume G contains generator (10.a) only:

$$\mathtt{G} = \{\forall_x(\ lawyer(x),\ \exists_{e_r}[(Rexist\ e_r) \wedge run(e_r) \wedge agent(e_r,\ x)]\ )\}$$

And, suppose the domain is made up of the individuals John and Jack, the former being a lawyer, the latter being not. G is equivalent to the following G':

$$\begin{aligned}\mathtt{G'}=\{\ &(\ lawyer(\mathtt{John}),\ \exists_{e_r}[(Rexist\ e_r) \wedge run(e_r) \wedge agent(e_r,\ \mathtt{John})]\ )\\ &(\ lawyer(\mathtt{Jack}),\ \exists_{e_r}[(Rexist\ e_r) \wedge run(e_r) \wedge agent(e_r,\ \mathtt{Jack})]\ )\ \}\end{aligned}$$

Since John is a laywer while Jack is not, the *propositional* symbol "$lawyer(\mathtt{John})$" belongs to our initial facts while "$lawyer(\mathtt{Jack})$" does not, i.e. $lawyer(\mathtt{John}) \in \mathtt{A}$. In $\mathtt{out}_1$, we infer that John is obliged to run, i.e.

$$\mathtt{out}_1(\mathtt{G'},\ \mathtt{A}) = \{\ run(f_1(\mathtt{John})) \wedge agent(f_1(\mathtt{John}),\ \mathtt{John})\ \}$$

Where we substituted[9] the existential quantifier on e_r with a Skolem function f_1, so that $\mathtt{out}_1(\mathtt{G'},\ \mathtt{A})$ again contains propositional symbols only. Thus, it satisfies requirement (2) and, in $\mathtt{out}_3$, it could be reused to trigger other obligations, e.g., being applied to an obligation such as "every runner must wear a pair of shoes".

We stress again that, thanks to Reification, we are not increasing the complexity of the original I/O logic. On finite domains, the formulae we are going to use easily come out to be propositional. Original I/O logic definitions and proofs of soundness and completeness still hold, modulo generalizations via universal and existential quantifiers. For instance, CT is modified as follows:

CT (cumulative transitivity):

from

$$\forall_{x_1..x_n}(\ \Phi(x_1,\ldots,x_n),\ \exists_{z_1..z_k}[\Psi(y_1,\ldots,y_m,z_1,\ldots,z_k)]\ ),$$
$$\text{with } \{y_1,\ldots,y_m\} \subseteq \{x_1,\ldots,x_n\}$$

and

$$\forall_{x_1..x_n w_1..w_r}(\ \Phi(x_1,\ldots,x_n) \wedge \Psi(w_1,\ldots,w_r),\ \exists_{k_1..k_i}[\Upsilon(t_1,\ldots,t_l,k_1,\ldots,k_i)]\ ),$$
$$\text{with } \{t_1,\ldots,t_l\} \subseteq \{x_1,\ldots,x_n,w_1,\ldots,w_r\}$$

to

$$\forall_{x_1..x_n}(\ \Phi(x_1,\ldots,x_n),\ \exists_{k_1..k_i m_1..m_s}[\Upsilon(p_1,\ldots,p_c,k_1,\ldots,k_i,m_1,\ldots,m_s)]\ ),$$
$$\text{with } \{m_1,\ldots,m_s\} \subseteq \{w_1,\ldots,w_r\} \text{ and } \{m_1,\ldots,m_s\} \cup \{p_1,\ldots,p_c\} \equiv \{t_1,\ldots,t_l\}$$

[9] Skolemization is merely a formality to meet requirement (2). Alternatively, we could allow existential quantifiers on inputs and define a different pattern-matching rule between the input and the heads of the generators.

6 Future Work and Conclusions

This paper represents the first step of the ProLeMAS research project, which aims to (1) fill the gap between the current logical formalizations of legal text, mostly propositional, and the richness of natural language semantics, and (2) implement a concrete NLP pipeline for automatically building formulae from existing legal documents. To this end, the first step is to move beyond the propositional level towards first-order logical frameworks, in order to enhance the expressivity fit to formalize the meaning of the phrases constituting the sentences. ProLeMAS proposes to achieve such a result by using the Reification-based constructs from Hobbs.

This paper shows how it is possible to merge Hobbs's account with I/O logic by adding universal and existential quantifiers to the latter. It also discusses how the complexity of the resulting framework, provided the domain is finite, does not increase with respect to that of propositional I/O logic. We consider this a great result, due to the complexity issues related to I/O logic.

Our next steps will involve the following future work:

(11) a. Studying how the ProLeMAS logic could deal with other kinds of norms, such as permissions, powers, etc. And, eventually, extending the account to provide a proper representation of their meaning.
b. Designing suitable legal ontologies to represent and restrict the meaning of relevant predicates, in order to trigger automatic reasoning on the individuals in the Abox. We are particularly interested in developing legal ontologies in the data protection domain. Under the pressure from technological developments during the last few years, the EU legislation on data protection has shown its weaknesses, and is currently undergoing a long and complex reform that is finally approaching completion.
c. Building a concrete pipeline to populate the Abox of the ontology. In ProLeMAS, the pipeline will firstly process the documents via dependency parsing, then it will define a syntax-semantic interface from the dependency trees to the final formulae.

References

1. Cesare Bartolini and Robert Muthuri. Reconciling data protection rights and obligations: An ontology of the forthcoming eu regulation. In *(To appear in) Proceedings of the Workshop on Language and Semantic Technology for Legal Domain (LST4LD)*, 2015.
2. Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, and Leon van der Torre. Nlp challenges for eunomos, a tool to build and manage legal knowledge. In *Language Resources and Evaluation (LREC)*, pages 3672–3678. European Language Resources Association (ELRA), 2012.
3. Guido Boella, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. Learning from syntax generalizations for automatic semantic annotation. *Journal of Intelligent Information Systems*, 43(2):231–246, 2014.

4. Guido Boella, Llio Humphreys, Marco Martin, Piercarlo Rossi, and Leendert van der Torre. Eunomos, a legal document and knowledge management system to build legal services. In *AI Approaches to the Complexity of Legal Systems. Models and Ethical Challenges for Legal Systems, Legal Language and Legal Ontologies, Argumentation and Software Agents*, pages 131–146. Springer, 2012.
5. D. Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, 1967.
6. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. (eds.) van der Torre. *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.
7. Sepideh Ghanavati. *Legal-URN framework for legal compliance of business processes*. University of Ottawa, 2013.
8. M. Hepp. Ontologies: State of the art, business potential, and grand challenges. In M. Hepp, P. De Leenheer, A. de Moor, and Y. Sure, editors, *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, pages 3–22. Springer, 2007.
9. J.R. Hobbs. Monotone decreasing quantifiers in a scope-free logical form. In *in Semantic Ambiguity and Underspecification*, pages 55–76. 1995.
10. J.R. Hobbs. The logical notation: Ontological promiscuity. In *Chapter 2 of Discourse and Inference*. 1998. Available at http://www.isi.edu/∼hobbs/disinf-tc.html.
11. Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli, and Giulia Venturi. Ontology learning from italian legal texts. In *Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood*, pages 75–94, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.
12. David Makinson and Leendert van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.
13. David Makinson and Leendert van der Torre. What is input/output logic? In Benedikt Lwe, Wolfgang Malzkom, and Thoralf Rsch, editors, *Foundations of the Formal Sciences II*, volume 17 of *Trends in Logic*, pages 163–174. Springer Netherlands, 2003.
14. David Makinson and Leendert W. N. van der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.
15. L. T. McCarthy. A language for legal discourse i. basic features. In *Proc. of the 2nd International Conference on Artificial Intelligence and Law (ICAIL '89)*, ACM Press, 1989.
16. L. T. McCarty. Deep semantic interpretations of legal texts. In *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA*, pages 217–224, 2007.
17. X. Parent and L. van der Torre. Input/output logic. In J. Horty, D. Gabbay, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, London, 2013.
18. L. Robaldo and E. Miltsakaki. Corpus-driven semantics of concession: Where do expectations come from? *Dialogue&Discourse*, 5(1), 2014.
19. Xin Sun. How to build input/output logic. In *15th International Workshop on Computational Logic in Multi-Agent Systems*, pages 123–137, 2014.
20. Xin Sun and Diego Agustin Ambrossio. On the complexity of input/output logic. In *to appear in the Proceedings of The Fifth International Conference on Logic, Rationality and Interaction*, 2015.
21. A. Valente. *Legal Knowledge Engineering: A Modeling Approach*. PhD thesis, Amsterdam University, The Netherlands, 1995.

Utilization of Multi-Word Expressions to Improve Statistical Machine Translation of Statutory Sentences

SAKAMOTO Satomi[1], OGAWA Yasuhiro[2,1], NAKAMURA Makoto[3], OHNO Tomohiro[2,1], and TOYAMA Katsuhiko[2,1]

[1] Graduate School of Information Science, Nagoya University
[2] Information Technology Center, Nagoya University
[3] Japan Legal Information Institute, Graduate School of Law, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
Email: {satomi,yasuhiro}@kl.i.is.nagoya-u.ac.jp

Abstract. Statutory sentences are generally difficult to read because of their complicated expressions and length. Such difficulty is one reason for the low quality of statistical machine translation (SMT). Multi-word expressions (MWEs) also complicate statutory sentences and extend their length. Therefore, we proposed a method that utilizes MWEs to improve the SMT system of statutory sentences. In our method, we extracted the monolingual MWEs from a parallel corpus, automatically acquired these translations based on the Dice coefficient, and integrated the extracted bilingual MWEs into an SMT system by the single-tokenization strategy. The experiment results with our SMT system using the proposed method significantly improved the translation quality. Although automatic translation equivalent acquisition using the Dice coefficient is not perfect, the best system's score was close to a system that used bilingual MWEs whose equivalents are translated by hand.

Keywords: multi-word expressions, statistical machine translation, legal information sharing

1 Introduction

As human globalization continues, the translation of Japanese statues into foreign languages is required to meet such demands as promoting investment and facilitating international transactions. In 2009, the Japanese Ministry of Justice released the Japanese Law Translation Database System (JLT) [7], which translates Japanese statues into English. However, JLT's translation speed and volume are inadequate because of the difficulty of legal translations.

Many reasons have been offered for this difficulty: excessive technical terms, complicated dependencies in the sentence structure, long sentences, etc. These reasons are not independent. A sentence that includes complicated expressions with many modifiers becomes too long. This difficulty degrades the quality of

statistical machine translation (SMT). SMT systems learn how likely it is that the words in the target language are the translations of words in the source language in a parallel corpus, which is a collection of bilingual aligned sentences, by automatic word alignment. The quality of the word alignment affects the translation. For long sentences, word alignment tends to fail due to excessive alignment candidates, which makes the translation quality degraded.

Hung et al. [6] split long statutory sentences into sub-sentences based on their logical structure and improved the translation of a tree-based SMT system. In contrast, we focus on multi-word expressions (MWEs), which are defined as "idiosyncratic interpretations that cross word boundaries (or spaces)" [17]. Statutory sentences include many MWEs, such as technical terms and boilerplate phrases. In this paper, we use MWEs in the broader sense, where they include not only compound nouns and idiomatic phrases but also functional expressions and other meaningless segments. Since some of them tend to be translated into a foreign language in a non-compositional way, their automatic word alignments suffer from noise. Tsvetkov et al. [19] focused on the fact that MWEs cause incorrect word alignment in a parallel corpus and proposed a general methodology to extract bilingual MWEs with such misalignments. They used GIZA++ [11], which is a popular bilingual word aligner, and acquired the translations of extracted MWEs from its phrase table. However, since their candidates are based on misalignment, the translation quality is unreliable. Thus, we introduce another method based on the Dice similarity coefficient to acquire the translations of MWEs and introduce a single-tokenization technique to integrate bilingual MWEs with SMT.

The purpose of this paper is to improve the translation quality of statutory SMT with MWEs. We propose a utilization method that extracts MWEs by Tsvetkov's method and acquire their translations by word alignment based on the Dice coefficient, and integrate the extract bilingual MWEs into an SMT system by Pal's single-tokenization strategy [12]. We evaluate the contribution of our method on SMT with translation experiments.

This paper is organized as follows: in the next section, we describe the related works of MWEs. We propose our MWE extraction method in Section 3. In Section 4, we describe and evaluate three experiments. The first experiment is an acquisition bilingual MWE dictionary using our proposed method. The second and third are translations using an SMT with the acquired bilingual MWEs. Finally, we summarize and conclude our paper in Section 5.

2 Related Works

2.1 Multi-Word Expressions for Natural Language Processing

MWEs are the expressions of compound words. Examples include conjunctions ("as well as"), idioms ("keep one's fingers crossed" that means to hope for a positive result), phrasal verbs ("find out"), compound nouns ("bus stop"), phrasal prepositions ("according to"), etc [17]. MWEs appear in all text genres and pose

significant problems for every kind of natural language processing (NLP) [17]. Since some have a meaning that cannot be derived from the component words, identifying and understanding MWEs is essential to language understanding and is important for any NLP applications that address robust language meaning and use. For example, MWEs play a role in tasks of word sense disambiguation since they are less polysemous than mono-words on average. Finlayson and Kulkarni [4] argued that the word *world* has nine different senses and *record* has fourteen, but the MWE *world record* has only one. In addition, many NLP tasks and applications like optical character recognition, morphological and syntactic analysis, information retrieval, computer-aided lexicography, and foreign language learning can be supported by MWEs [15].

2.2 Statistical Machine Translation with MWEs

Bilingual MWEs can contribute to SMT improvement. Improving the performance of an SMT system needs a good quality word and phrase alignment that acquires translation knowledge from a parallel corpus. MWEs can solve the major lexical ambiguity problems for any language and improve the quality of word alignment.

Ren et al. [16] proposed three methods to improve the translation model of SMT system using the bilingual MWEs extracted by an automatic process. Their first method is model retraining in which they take the automatically extracted bilingual MWEs as parallel sentence pairs, add them to the training corpus, and retrain the model using GIZA++. In the second method, the additional feature, they append one feature to a bilingual phrase table to indicate whether a bilingual phrase contains bilingual MWEs. The third method is the additional phrase table of bilingual MWEs in which they construct an additional phrase table that contains automatically extracted bilingual MWEs and combine the original phrase table and the newly constructed bilingual MWE table. These methods improved the SMT system and achieved the highest improvement with the additional features. SMT systems must train MWEs as special expressions that are different from common phrases.

Following the above study, Pal et al. [12] proposed a method of handling MWEs by tokenizing them into a single word: *single-tokenization*. The single-tokenization of MWEs is a process where the spaces in MWEs are replaced by a special symbol, "_", and an MWE is regarded as a single word. They successfully improved the SMT quality using this method to integrate bilingual MWEs that were automatically extracted.

2.3 MWE Extraction Methods

Since MWEs resemble collocations, early approaches to identifying them focused on their collocational behavior. However, in fact, collocation measures are inadequate for identifying MWEs. Hybrid methods, which combine word statistics with such linguistic information as morphological, static, and semantic idiosyncrasies, need to extract idiomatic MWEs [14].

Some works have focused on the semantic properties of MWEs. One work proposed a method that used Latent Semantic Analysis [8] and measured semantic properties to replace semantically related terms [2].

In recent years, other works have exploited the translational correspondences of MWEs from a parallel corpus. Caseli et al. [1] identified MWEs by an alignment-based approach. They extracted source sequences whose lengths exceeded two words and aligned them with one or more words in the target text. Then they filtered this set to determine whether it complies with the pre-defined POS patterns or whether they are sufficiently frequent in the parallel corpus. They marked precision below 40% and recall around 5%. Zarrieß and Kuhen [20] also used aligned parallel corpora and focused on one-to-many word alignments.

The previous works focused on specific syntactic patterns and assumed that MWEs are distinguished as phrases in the processing of alignment systems. While these methods rely on the quality of automatic word alignment, there are cases when their quality is insufficient due to the shortage of language resources. Tsvetkov et al. [19] proposed a method for identifying MWEs in bilingual corpora for their Hebrew-English domain that suffers from a dearth of parallel corpora, semantic dictionaries, and syntactic parsers. They extracted MWEs with automatic word alignment and focused on where the word alignment system failed. We adopt part of this scheme in our proposed method and explain it in detail in the next section.

3 Proposed Method of Bilingual MWE Utilization

In this section, we propose a method of bilingual MWE utilization to improve the SMT system of statutory sentences. Our methodology consists of three steps. First, we extract monolingual MWEs from a parallel corpus by Tsvetkov's method [19], which is language independent. Although Tsvetkov's method can extract bilingual MWEs, the translation quality is inadequate for our scheme. Thus, secondly, we acquire their translations by word alignment based on the Dice coefficient. Finally, we integrate the bilingual MWEs into an SMT system by Pal's single-tokenization strategy [12].

3.1 Monolingual MWE Extraction from Misalignments

Tsvetkov et al. [19] concluded that MWEs have a non-compositional meaning and may be translated into a single word in a foreign language. They assumed that there may be three sources to misalignments (anything without a one-to-one word alignment) in parallel texts: MWEs that trigger one-to-many or many-to-many alignments, language-specific differences (e.g., the source language lexically captures notions that are realized morphologically, syntactically or in some other way in the target), or noise (e.g., poor translations, low-quality sentence alignment, and the inherent limitations of word alignment algorithms). They focused on misalignments as a resource to extract MWEs. They trusted the quality of the one-to-one alignments, which they verified with a dictionary and searched for

MWEs exactly in the areas where proper word alignment failed without relying on the alignment in these cases.

We summarize Tsvetkov et al.'s method below.

Process I: Resources and preprocessing the corpora Their methodology needs the following resources: a small parallel, sentence-aligned bilingual corpus; two large monolingual corpora; morphological processors (analyzers and disambiguation modules) for the two languages; and a bilingual dictionary. The parallel corpus is the source of the MWE candidate extraction. Monolingual corpora are the sources to compute the word frequencies for measuring the statistics of the candidates. We only use one-to-one word alignment entries in their bilingual dictionary.

First, we preprocess the corpora and the dictionary to reduce the language specific differences and the noise of the word alignment because automatic word alignment algorithms are noisy, and given a small parallel corpus data sparsity is a serious problem. The processing includes tokenizing, lemmatizing, etc. We then compute the frequencies of all the word bigrams and unigrams that occur in each of the monolingual corpora.

Process II: Word alignment and identifying MWE candidates Next, we compute the word alignment on the parallel corpus. We use GIZA++ [11] to word-align the text and capture the alignments merged in both directions. We look up all the one-to-one alignments in the merged alignments in the dictionary. If a pair exists in their bilingual dictionary, we remove it from the sentence and replace it with a special symbol: "*". By the replacement, we can remove a word that can be translated into a word in the other language from the MWE candidates.

Figure. 1(a) shows a Japanese sentence and its English translation in the parallel corpus. Here, 不服申立て (appeal) and との (between) are an MWE that cannot be literally translated. Fig. 1(b) shows after the preprocessing and Fig. 1(c) shows after the word alignment. Since there are three pairs (と and *and*, 訴訟 and *lawsuit*, 関係 and *relation*) in the dictionary, we replace them with "*" (Fig. 1(d) shows).

Process III: Filtering MWE candidates Now the object sentences are word sequences separated by "*"s, and we can address each sequence that contains MWE candidates. But these candidates also include noise caused by the misalignment. In this stage, we do not rely on the alignments; we concentrate on distinguishing whether the bigram of each word is a compound expression.

We use association measure PMI^k to prune these candidates. PMI^k is a PMI-based score that measures the co-occurrence among two things. Its formula is defined as follows:

$$\mathrm{PMI}^k(x, y) = \frac{P(x, y)^k}{P(x)P(y)}, \tag{1}$$

(a)	不服申立てと訴訟との関係 Relations Between Appeal and Lawsuit					
(b)	不服 申 立て と 訴訟 と の 関係 relations between appeal and lawsuit					
(c)	不服 申 立て	と	訴訟	と の	関係	
	appeal	and	lawsuit	between	relation	
(d)	不服 申 立て	*	*	と の	*	
	appeal	*	*	between	*	

Fig. 1. Example of word alignment and replacement

# of unigram	4,399	11,742	7,467	519,883	3,092,086
	不服	申	立て	と	の
# of bigram	1,092	7,443		12,696	
PMI^k	3.091	324.2		0.074	

Fig. 2. Examples of PMI^k-score of the candidate

where $P(x)$ is the number of occurrences of unigram "x" in the monolingual corpus and $P(x, y)$ is that of bigram "$x\ y$". We set weight k based on heuristics. A word sequence of any length is considered an MWE if all of its adjacent bigrams contain a score above the threshold. We computed PMI^k-scores based on the statistics of a large monolingual corpus. Finally, we restored the original forms of the words in the candidates and extracted them as MWEs.

In Fig. 1(d), which shows the replacement by "*", we consider each substring (不服 申 立て and と の) an MWE candidate and verify it by the PMI^k-score in the monolingual corpus. Fig. 2 shows the result of the PMI^k-score of these candidates. If the PMI^k threshold is set to 1.0, term 不服 申 立て is an MWE, and term と の is not, so we extract only 不服申し立て as an MWE.

3.2 Translation of Extracted MWEs Based on the Dice Coefficient

Even though the above method of Tsvetkov et al. is reasonable, it has a problem extracting MWE translations. They extract the translations of candidate MWEs from these alignments. For each MWE in the source-language sentence, they consider the translations of all the words in the target-language sentence that are aligned to the word constituents of the MWE as long as they form a contiguous string. Since their candidates are based on misalignment, the translation quality is unreliable. They removed translations that are longer than four words because they are often wrong. Moreover, although MWEs can be translated into different expressions in different target sentences, they didn't show a strategy for ranking translation candidates. Since our statutory corpus has many long terms, some

of which may be translated into different expressions depending on the context, we want to adequately acquire their translations.

Thus, based on the Dice coefficient, we introduce another word alignment method to acquire the translations of MWEs. To measure the similarity between two terms, the Dice coefficient is described as follows:

$$Dice(x, y) = \frac{2 \cdot freq(x, y)}{freq(x) + freq(y)} \quad (0 \leq Dice(x, y) \leq 1), \tag{2}$$

where $freq(x)$ and $freq(y)$ denote the numbers of occurrences of term x in the source sentences and term y in the target ones, and $freq(x, y)$ denotes the number of co-occurrences of x and y in the aligned sentences.

In our proposed method, we give x as a candidate of source-language MWE and calculate its Dice coefficient to determine the highest y that consists of one or more words in the target-language sentences as a translation equivalent of x. Next we gather the remaining sentences that do not contain the equivalent and repeat the calculation of the Dice coefficient if its value exceeds a threshold to get multiple translation equivalents for one MWE.

3.3 Integration Strategy by Single-Tokenization Using Bilingual MWEs

We use the single-tokenization technique proposed by Pal et al. [12] to integrate our bilingual MWEs with the SMT system. The single-tokenization method embraces active processing to replace the spaces in MWEs as "_". It helps the SMT system treat MWEs as single words for training and decoding. We single-tokenize the bilingual MWEs if they appear in both the source and target sentences as preprocessing in the training corpus.

4 Experiments and Discussions

In these experiments, we extracted bilingual MWEs from a statutory corpus and evaluated them based on their contributions to improving the SMT system. First, we extracted bilingual MWEs using our proposed method. Second, we investigated an adequate threshold of the Dice coefficient. Finally, we evaluated the single-tokenization of MWEs to improve the translation model.

4.1 Extraction Experiment

In the first experiment, we extracted bilingual MWEs using our proposed method.

Resources We prepared three resources: a Japanese-English statutory parallel corpus, a Japanese statutory monolingual corpus, and a general bilingual dictionary. The parallel corpus, which contains 151,951 parallel sentences, is part

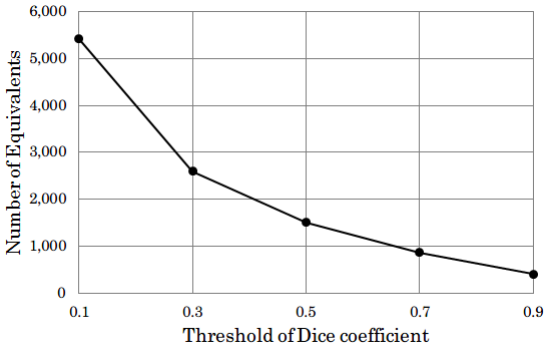


Fig. 3. Equivalents of each threshold of Dice coefficient

of the corpus of the Japanese Law Translation Database System[4], where we kept the rest of the corpus (randomly identified 15,026 sentences) for the next experiment's test. We used the Japanese subset of the parallel corpus as the monolingual corpus. The bilingual dictionary is Eijiro [3] to which we added 1,004 of our own translations from Chinese numerals to Roman numerals. The 5,618 entries of the bilingual dictionary occur in our parallel corpus with one-to-one word alignment.

Preprocess We lowercased and tokenized both corpora by morphological analyzers: MeCab [10] for Japanese and tokenizer.perl [9] for English. We also lemmatized them by scripts: MeCab for Japanese and Ruby Lemmatizer[5] for English. Since GIZA++ cannot handle long sentences, we cleaned up sentences longer than 80 words in Japanese or in English from the parallel corpus.

Bilingual MWE Extraction We extracted statutory bilingual MWEs using the method proposed in Section 3. We set weight k to 2.7 and the threshold to 1 for PMI^k. Since the extracted monolingual MWEs remained noisy, we cleaned up some of them, including their punctuation, and acquired English translations of the Japanese MWEs based on the Dice coefficient.

Results After removing the punctuation, the number of Japanese MWE candidates was 2,829. Fig. 3 shows the numbers of acquired MWE equivalents of each threshold in line charts. Since we allowed one Japanese MWE to have one or more English equivalents, the number of equivalents exceeds the number of Japanese MWEs. 544 candidates could not acquire any English equivalent even when the threshold value was 0.1.

[4] `http://www.japaneselawtranslation.go.jp/`
[5] `https://github.com/yohasebe/lemmatizer/`

4.2 Translation Experiment of Dice Coefficient

To evaluate our bilingual MWEs that we extracted above, we compared the translation models trained on the corpus with MWEs. In this experiment, bilingual MWEs include some improper translation equivalents since automatic translation equivalent acquisition using Dice coefficient is not perfect. We evaluated the influence of such mistranslations by comparing the translations using the bilingual MWEs by the Dice coefficient with those that used the bilingual MWEs whose equivalents were translated by hand.

Training Data As training data, we used the same parallel corpus that consist of 151,951 parallel sentences from previous section and prepared three translation model types: baseline, Dice, and manual. We lowercased and tokenized the corpus, where the baseline model was trained on the normally tokenized corpus, and the others were trained on the corpus whose MWEs were single-tokenized. For the Dice models, we acquired translations of the MWEs by the Dice coefficient and prepared five models by changing the threshold from 0.1 to 0.9. For the manual models, we modified the MWE translations by hand after acquiring them by the Dice coefficient, and prepared five models.

Since GIZA++ cannot handle long sentences, we cleaned up sentences longer than 80 words from the corpus, as in the previous section. Since the single-tokenization technique reduces the sentence length, the Dice and manual models used a slightly larger corpus than the baseline model.

Test Data We prepared 15,026 Japanese statutory sentences as test data, none of which overlapped with the training data. We lowercased and tokenized them, like the training data; we single-tokenized the MWEs in the case of translating with the Dice and manual models.

Translation We used the following freely available translation tools: GIZA++, SRILM [18], and Moses [9]. GIZA++ trains a translation model from the parallel corpus described above. In this experiment, we used '-grow-diag-final-and -msd-bidirectional-fe' as the command options. SRILM trains a language model in English from the parallel corpus. We used '-ukndiscount -interpolate' as the smoothing command option. The other parameters were set to default.

Evaluation To evaluate the outputs of the translation systems, we used automatic evaluation metric BLEU [13], which scores the system's outputs by referring to human translations. We removed the '_'s of the single-tokenized MWEs in the translated sentences in the evaluations.

Results and Discussion Table 1 shows the evaluation results. The scores that are significantly higher ($p < 0.05$) than the baseline model are written in bold. Fig. 4 shows the line charts of the scores.

Table 1. Scores of models using bilingual MWEs by the single-tokenization

Model		BLEU Score
Baseline model		30.32
Dice model	0.1	**31.19**
	0.3	30.75
	0.5	**30.98**
	0.7	**31.26**
	0.9	30.51
Manual model	0.1	30.89
	0.3	**30.97**
	0.5	**31.30**
	0.7	**31.35**
	0.9	30.70

Both the Dice and manual models significantly improved the BLEU scores, where the peaks of the Dice threshold were 0.7 in both models. The Dice coefficient's threshold is related to the number of bilingual MWEs (Fig. 3). Table 1 and Fig. 3 imply that the scores are reduced by too many MWEs, which are extracted by the low threshold. Because of the incorrect translations acquired by the Dice coefficient, these noisy equivalents may disturb the training and decoding in the Dice model. In the manual model, on the other hand, the scores are also not improved by adding many more bilingual MWEs, even though their translation equivalents are correct. One reason is because of the variation of the translations. The Dice coefficient scores tend to be low when the MWEs have plural translations, which include variants representing tense, voice, singular, plural, etc. These variations negatively affect the single-tokenization translations. Thus, setting the thresholds to 0.7 is suitable for utilizing the bilingual MWEs in our method.

Although some of the translation equivalents acquired by the Dice coefficient are incorrect, as mentioned above, the scores of the Dice and manual models are close at the 0.7 threshold. We examined the MWEs acquired at the 0.7 threshold, and found some of their equivalents are partially correct. Table 2 shows their examples. They are not correct translations, but they co-occur in the parallel text with a high probability. Single-tokenization integrates MWEs into a single word and reduces the search space and the misalignments. As a result, even if we acquire the partial fragments of the correct equivalents, they can help the SMT system.

While most of the MWE pairs extracted at the 0.7 threshold are noun phrases, some are verb phrases, and the others are stereotyped expressions. We acquired translations over four words that cannot be extracted by the method of Tsvetkov et al. [19] For example, the long noun MWE, '地域_密着_型_介護_予防_サービス_費', got a proper translation that consisted of six words: 'allowance for community-based long-term preventative care'.

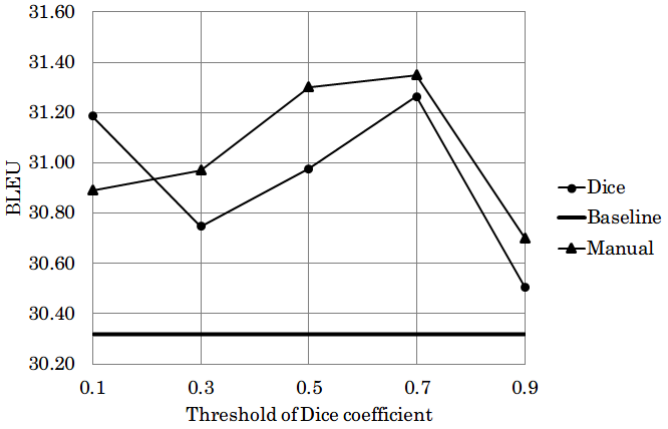


Fig. 4. Scores of models of using bilingual MWEs by the single-tokenization

4.3 Translation Experiment of Single-Tokenization

In our third experiment, we evaluated MWE single-tokenization by comparing other methods that utilize MWEs. We used the bilingual MWEs extracted by the Dice threshold of 0.7.

Integration Methods We tested two other naive methods to utilize MWEs, which are *train* and *table* methods, because Ren et al. [16] used them for evaluation. For the train method, we added bilingual MWEs to the training corpus and obtained, as results, new alignments and a phrase table. For the table method, bilingual units are incorporated as a new phrase table in addition to the baseline model's phrase table, where the Moses system uses multiple phrase tables. We gave the probability weighting by the Dice coefficient scores to the new phrase table of the MWEs.

Results and Discussion Table 3 shows the evaluation results by the BLEU metric. The scores, which are significantly higher ($p < 0.05$) than the baseline, are written in bold.

Compared to the baseline, the train and table methods improved the BLEU scores, but not significantly. Our methodology using single-tokenization is more effective than these naive methods.

5 Conclusion and Future Work

We described a methodology for utilizing multi-word expressions to improve the statistical machine translation of statutory sentences. We extracted the monolingual MWEs from a parallel corpus, automatically acquired their translations

Table 2. Acquired partial translation of MWEs at 0.7 threshold

Japanese MWE	Acquired translation	Correct translation
銀行 持株 会社	bank holding	**bank holding** company **bank holding** companies
産前 産後	after childbirth	before or **after childbirth** before and **after childbirth**
行為 に対する 罰則	penal provisions to	**penal provisions to** actions **penal provisions to** acts **penal provisions to** an act **penal provisions to** any acts
政令 で 定める	cabinet order	prescribed by **cabinet order** designated by **cabinet order** set forth in **cabinet order** stipulated by **cabinet order** ...

Table 3. Scores of models using bilingual MWEs by each integration methods

Model	BLEU
Baseline	30.32
Train	30.58
Table	30.48
Single-tokenization	**31.26**

based on the Dice coefficient, and integrated the extracted bilingual MWEs into the SMT system by a single-tokenization strategy. A SMT system with our proposed method significantly improved the translation quality more than both the baseline system and one using the other naive methods by MWEs. The system worked best when we set its Dice coefficient threshold to 0.7. Although automatic translation equivalent acquisition using the Dice coefficient is not perfect, the system's best score was close to one that used bilingual MWEs whose equivalents were translated by hand.

Our future works have four directions. The first is the further analyze of our method's performance. We wonder why the score of the Dice model at the 0.1 threshold is close to that of the model at the 0.7 and whether the performance depends on kinds of statute. We should evaluate the result in more detail to answer these questions. The second is to apply our method to other languages. Since our method is the language independent approach and MWEs appear in all text genres, we expect it to perform as well as it does in this paper. The third is further improvement of the SMT system's translation quality. Since our method is useful to reduce the search space where the word alignment system works, we believe that it is compatible with word reordering approach[5]. The last is the construction of multilingual MWE dictionaries for human translators working on Japanese statues. The acquired MWE dictionary includes both noun and verb phrases, which benefit human translators. We also plan to introduce a screening technique and select proper entries from it.

References

1. Caseli, H.d.M., Villavicencio, A., Machado, A., Finatto, M.J.: Statistically-Driven Alignment-Based Multiword Expression Identification for Technical Domains. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 1–8. (2009)
2. Van de Cruys, T., Moirón, B.V.: Semantics-based Multiword Expression Extraction. In: Proceedings of the Workshop on A Broader Perspective on Multiword Expressions, pp. 25–32. (2007)
3. EDP, ALC Press Inc.: Eijiro, 8 edn. (2014)
4. Finlayson, M.A., Kulkarni, N.: Detecting Multi-Word Expressions improves Word Sense Disambiguation. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World, pp. 20–24. (2011)
5. Hideki I., Katsuhito S., Hajime T., and Kevin D.: Head Finalization: A Simple Reordering Rule for SOV Languages. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR, pp.244–251. (2010)
6. Hung, B.T., Le Minh, N., Shimazu, A.: Translating Legal Sentence by Segmentation and Rule Selection. International Journal on Natural Language Computing, Vol. 2, No. 4, pp. 35–54. (2013)
7. Katsuhiko, T., Daichi, S., Yasuhiro, S., Yasuhiro, O., Tokuyasu, K., Tariho, K., Yoshiharu, M.: Design and Development of Japanese Law Translation Memory Database System. Law via the Internet 2011, 12 pages. (2011)
8. Katz, G., Giesbrecht, E.: Automatic Identification of Non-Compositional Multi-Word Expressions using Latent Semantic Analysis. In: Proceedings of the Work-

shop on Multiword Expressions: Identifying and Exploiting Underlying Properties, pp. 12–19. (2006)
9. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. (2007)
10. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis. In: Proceedings of Empirical Methods in Natural Language Processing, pp. 230–237. (2004)
11. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, Vol. 29, No. 1, pp. 19-51. (2003)
12. Pal, S., Naskar, S.K., Bandyopadhyay, S.: MWE Alignment in Phrase Based Statistical Machine Translation. In: Proceedings of the XIV Machine Translation Summit, pp. 61–68. (2013)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. (2002)
14. Piao, S.S., Rayson, P., Archer, D., McEnery, T.: Comparing and combining a semantic tagger and a statistical tool for MWE extraction. Computer Speech & Language, Vol. 19, No. 4, pp. 378–397. (2005)
15. Ramisch, C.: Multiword Expressions Acquisition: A Generic and Open Framework. Springer (2014)
16. Ren, Z., Lü, Y., Cao, J., Liu, Q., Huang, Y.: Improving Statistical Machine Translation Using Domain Bilingual Multiword Expressions. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 47–54. (2009)
17. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword Expressions: A Pain in the Neck for NLP. In: Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing, pp. 1–15. (2002)
18. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: Proceedings the 7th International Conference on Spoken Language Processing, Vol. 2, pp. 901–904. (2002)
19. Tsvetkov, Y., Wintner, S.: Extraction of Multi-word Expressions from Small Parallel Corpora. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1256–1264. (2010)
20. Zarrieß, S., Kuhn, J.: Exploiting Translational Correspondences for Pattern-Independent MWE Identification. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 23–30. (2009)

Graph-based Legal Document Similarity Search

Katsumasa Yoshikawa and Daisuke Takuma

IBM Research, Tokyo,
19-21 Nihonbashi Hakozaki-cho Chuo-ku, Tokyo 103-8510, Japan
{KATSUY,TA9MA}@jp.ibm.com
http://www.research.ibm.com/labs/tokyo/index_j.shtml

Abstract. This paper presents a novel approach to legal document similarity searches. Models for document similarity searches (DSS) usually exploit linguistic similarities based on linguistic features. However, such linguistic similarities are not enough for handling legal documents. We define a *legal similarity* tailored for legal document processing. Our system captures legal similarity by using a graph structure composed of law articles and legal documents.
We experimented on our system by using hand-crafted data. The results demonstrated that our approach outperforms a conventional approach without legal similarities. Moreover, we performed a qualitative analysis to see whether our system effectively outputs legally similar documents.

Keywords: legal text processing, document similarity search, legal similarity, graph random walk

1 Introduction

Legal document (text) processing is one of the most important areas in artificial intelligence (AI) and law. Some significant successes in natural language processing (NLP) have given it a boost. In particular, a DeepQA system, *IBM Watson* [9–11], won a quiz show called *Jeopardy!*, and this victory hints at the potential of applying AI systems to technical domains such as medical, financial, and legal.

In this paper, we propose a novel approach to legal document similarity searches. Keyword-based legal document search systems are commonly used by lawyers, jurists, and students in law schools. But selecting the right keywords is often a difficult task. Here, the document similarity search (DSS) is an important component of modern information retrieval that enables us to avoid keyword selection. Rather than several query keywords, a DSS uses a document as a query and finds similar documents to the query document. However, there are differences between *general* and *legal* document similarity searches. In the general domain, similarities are determined from linguistic features. On the other hand, in the legal domain, legal features are more valuable information for capturing legal similarities.

Table 1 shows several cases involving Japanese Supreme Court decisions that illustrate the differences between general and legal similarities. Supposing Q is a query document, which cases (C1-C4) are the most legally similar? Q is a "高知落石事件" in which the plaintiff demanded compensation for damages suffered in a car accident caused by a fallen rock on a national road. The legal standpoint of the suit is exactly the issue of *administrative defect*. General DSS systems do not take into account such legal matters. As a result, they may identify C1 as the most similar case. C1 is linguistically

Table 1: Differences between General and Legal Similarities

ID	Date	Description	Name
Q	[S45.8.20]	Claim for damages of the car accident happened by an obstacle (fallen rock) on a national road	高知落石事件 (Kōchi Fallen Rock Accident)
C1	[H12.1.27]	Claim for clearing obstacles against cars on a road designated by the Building Standards Act 42-II	車止め撤去請求事件 (Demand to clear car obstacles on a road)
C2	[S50.7.25]	Claim for damages of the car accident happened by a leaving truck on a national road	87 時間大型故障車放置事件 (87 hours Track Leaving Accident)
C3	[S61.3.25]	Claim for damages of the a fall accident at a station because of missing studded paving blocks	点字ブロック事件 (Studded Paving Block Accident)
C4	[S50.6.26]	Claim for damages of flood disaster due to the insufficent maintenance of national river	多摩川水害訴訟 (Tama River Flood Disaster Suit)

very similar to Q because it shares many keywords such as *car, road*, and *obstacle*. However, C1 is far from Q in its legal points because C1 is not about compensation from the government but is rather a petition for the statement of interference. Though C2 is the most correct answer, C3, and C4 can also be considered correct. C3 and C4 are linguistically much different from Q but they are legally very similar to each other. In C3 and C4, the issues are administrative defects pertaining to a train station and river, respectively.

Thus, simple linguistic similarity is not enough for legal DSS. We have to consider legal points at issue represented in laws. Here, we define *legal similarity* as the number of the shared law articles. In the cases of Table 1, we can simply evaluate their legal similarity by whether or not they refer to *State Redress Act 2-I*. Actually, Q, C2, C3, and C4 refer to the law article, but C1 does not. This is a clear reason why Q is similar to C2-C4 and different from C1.

We devised a way to reveal legal similarities by using the edges of a graph structure. The search for similar documents is driven by a graph random walk algorithm. We experimented on Japanese decision papers and found that our method could effectively perform legal DSSs. We also analyzed the outputs and evaluated the improvements had by introducing legal similarity. Our contributions are as follows.

- We propose a new method for legal document similarity searches.
- We investigate the differences between the models with and without legal similarities.

The remainder of this paper is organized as follows: Section 2 describes the related work on legal document processing and document similarity searches. Section 3 explains our method in detail. Section 4 presents the setup of our experiments, and Section 5 discusses the results. Section 6 concludes the paper and presents ideas for future work.

2 Related Work

In this section, we describe previous studies on legal document processing (Section 2.1) and document similarity search (Section 2.2).

2.1 Legal Document Processing

Dealing with laws automatically is one of the goals of artificial intelligence (AI). There have been various studies of legal document processing (LDP). Studies on legal reasoning build theoretical models based on laws and try to handle legal documents [2, 3, 7, 30, 35]. Finding and analyzing arguments plays an important role in legal reasoning [27, 21, 25, 4].

Legal engineering is another branch of LDP, and its studies focus on how to develop and maintain laws by analogy to software engineering. It applies natural language processing (NLP) techniques to legal documents. In recent years, not only rule-based analyzers [16, 14, 23] but also machine learning based systems [6, 5, 34] have been developed.

There have been a number of workshops on using LDP in conjunction with NLP and information retrieval (IR). Semantic Processing of Legal Texts (SPLeT) [12, 22] has been held since 2008 as part of the LREC, a NLP conference.[1] Moreover, the Text Retrieval Conference (TREC) in the IR community has a Legal Track [2] that requires participants to retrieve information for e-Discovery. Our work also uses NLP technology for analyzing legal documents, and our search method is an unsupervised approach executed on graph structures. Hence, our study is at the intersection of LDP, NLP, and IR.

2.2 Document Similarity Search

The document similarity search (DSS) is a modern IR task to find documents that are similar to a query document. According to user experience, DSS is more advantageous than keyword-based search [8, 36, 15]. Traditional keyword-based search engines require us to select several appropriate query words. However, in technical domains such as law and medicine, selecting the right keywords can be very difficult even for experts.

Measuring document similarity is a popular task in the NLP field. Moreover, it can be applied to other tasks such as document classification, question answering, and document summarization. There are a large number of document representations: bag-of-words (BoW) vector [29], generalized latent semantic analysis [20], and concept graph [1]. Though our graph-based approach can work in conjunction with the other complex vector representations, we simply encode a document into a vector by using the well-known *TF-IDF term weighting scheme* [28, 19].

3 Method

We framed the legal DSS task as a graph mining problem. Section 3.1 describes how to construct a graph of legal documents and law articles. Section 3.2 presents a search process on the graph.

3.1 Construction of Graph with Documents and Laws

We first define the basic graph structure for legal DSS. Figure 1 illustrates a schematic view of the graph that we will define.

[1] http://www.lrec-conf.org/

[2] http://trec-legal.umiacs.umd.edu/

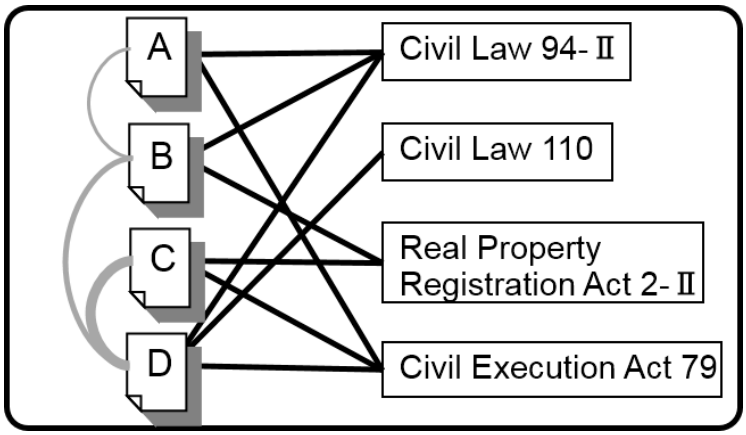


Fig. 1: Example of a Legal Graph

Graph Definition

Figure 1 has four document nodes and four article nodes. The document nodes (e.g. A-D) represent the targets documents that we have to analyze. The article nodes represent law articles (e.g. *Civil Law 94-II, Civil Law 110, Real Property Registration Act 2-II, Civil Execution Act 79*), and they provide us with important clues for measuring similarity. In Figure 1, we define two types of edges those between two document nodes (shown as grey lines) and those between a document node and an article node (shown as black lines). As we mentioned in Section 1, we define legal similarity as *the number of shared law articles*. In the proposed graph, the documents sharing law articles represent the document nodes with incoming edges from the same article node. For brevity, we call the two types of edges *D-D* (document-document) edges and *D-L* (document-law article) edges. A model with only *D-D* edges does not consider features of law articles and captures only linguistic similarity of documents. In order to address legal similarity, we have to involve *D-L* edges. The differences between models with and without *D-L* edges are shown in Sections 4 and 5. The proposed graph enables us to capture both legal and linguistic similarities in a unified framework.

Edge Weight Estimation

Before searching, we have to estimate the weights of the graph edges. The edges in Fig. 1 have several weights. Their weights should be based on the relevance scores between the documents or between the document and article they connect.

First, we represent all documents by vector space model using TF-IDF term weighting [28]. Both term frequency (TF) and inverse document frequency (IDF) are calculated for the legal documents and law articles. For a document d, we define the weight vector v_d as a combination of term weights $w_{t,d}$ such that,

$$\begin{aligned} v_d &= [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T \\ w_{t,d} &= tf_{t,d} \cdot idf_{t,D} \end{aligned} \tag{1}$$

where t is a context word, N is the total number of words in the vocabulary, and D is the number of the whole documents. The details of the TF-IDF model are described in [28].

Table 2: Definition of Symbols

Symbol	Definition
$\mathbf{W} = \left[w_{i,j}\right]$	the weighted graph, $1 \leq i, j \leq n$
$\tilde{\mathbf{W}}$	the normalized weighted adjacent matrix associated with $\mathbf{W}$
$\mathbf{Q}$	the system matrix associated with $\mathbf{W}$: $\mathbf{Q} = \mathbf{I} - c\tilde{\mathbf{W}}$
$\boldsymbol{r_i} = \left[r_{i,j}\right]$	$n \times 1$ ranking vector, $r_{i,j}$ is the relevance score of node j wrt node i
$\boldsymbol{e_i}$	$n \times 1$ starting vector, the i-th element 1 and 0 for others
c	the restart probability

After representing a document as a vector, we can estimate similarity scores between documents. We define *D-D* similarity as the cosine similarity between document vectors,

$$sim(d_1, d_2) = \frac{v_{d_1} \cdot v_{d_2}}{\|v_{d_1}\| \, \|v_{d_2}\|}, \tag{2}$$

in which the distance between the document vectors is based on linguistic features. Though *D-D* edges can be constructed by taking the Cartesian product of all document nodes, we omit the *D-D* edges which have the similarity scores less than 0.1. Note that there are some node pairs which have no edges.

D-L edges are much more difficult to give weights automatically. An article is often much smaller than a document and does not have enough features for estimating similarities. In addition, law articles are too numerous to be able to estimate similarities with pinpoint accuracy. Here, we collected 8142 Japanese laws containing 197509 articles for our experiment. We chose the referenced law annotations that were annotated by hand for each legal document (decision paper). We fixed the weights of *D-L* edges as 1.0. Some of the documents had no referenced law annotations, thereby causing a problem with the recall of these manual annotations. However, our method overcomes this recall problem of missing referenced laws by using a graph random walk. We explain how it does so in the next section.

3.2 Graph Search for Legal Document Similarity Search

Our legal DSS system estimates legal similarity as a graph proximity search. There have been various studies to address node-to-node proximities (similarities) [17, 32, 18, 13]. Random walk with restart (RWR) [26] is one of the most popular methods. RWR calculates a proximity based on not only the graph distance of the nodes, but also the features of the targeted graph structure [33, 31].

Table 2 lists the symbols used for our explanation. RWR is defined as follows,

$$\boldsymbol{r_i} = c\tilde{\mathbf{W}}\boldsymbol{r_i} + (1 - c)\boldsymbol{e_i} \tag{3}$$

which represents the iterative process starting from the query node i. $\boldsymbol{r_i}$ is a column vector where $r_{i,j}$ denotes the probability that the random walk is at node j. $\boldsymbol{e_i}$ is a column vector where the only i-th element is 1 and the others are 0.

The RWR process can be described in terms of a random walk particle. The first term of equation (3) represents the *random step*. A particle moves to its neighborhood with the transition probability that is proportional to their edge weights of $\tilde{\mathbf{W}}$. The second term of equation (3) drives the *restart step* which returns a particle to the starting node i with some probability $1 - c$. The relevance score of node j to node i is defined as a steady-state probability $r_{i,j}$. $r_{i,j}$ is the final existence probability after a large number

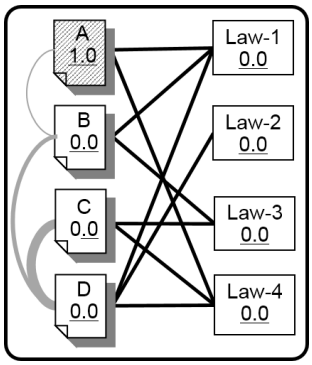


Fig. 2: Initial State

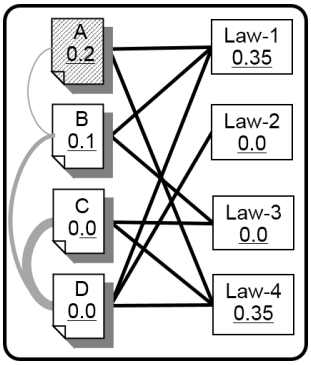


Fig. 3: First Step

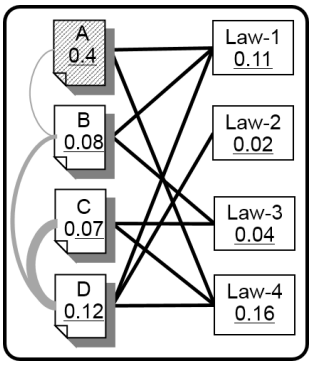


Fig. 4: Steady State

of iterations of the particle being at a certain node. c is an interpolation coefficient that we set to 0.8. Though we tested RWR process changing c from 0.5 - 0.9, the changes do not effect the DSS performance after a sufficient number of iterations.

The steady-state probability of RWR can also be obtained analytically because equation (3) defines a linear system problem. $\boldsymbol{r_i}$ is determined by equation (4):

$$\boldsymbol{r_i} = (1-c)(\mathbf{I} - c\tilde{\mathbf{W}})^{-1}\boldsymbol{e_i} = (1-c)\mathbf{Q^{-1}}\boldsymbol{e_i}. \tag{4}$$

The system matrix $\mathbf{Q}$ defines all the steady-state probabilities. When pre-computing $\mathbf{Q}^{-1}$, we can get $\boldsymbol{r_i}$ in a real-time manner. However, pre-computing and storing $\mathbf{Q}^{-1}$ is often infeasible when the dataset is large. RWR with an iterative process is called *OnTheFly* and RWR with a pre-computed $\mathbf{Q}^{-1}$ is called *PreCompute* [33]. We chose to take the *OnTheFly* approach because of its feasibility.

Let us illustrate how well suited *OnTheFly* RWR is to a legal DSS system. Figures 2-4 show an example of how RWR works in a legal DSS system. The figure is a simplified version of Figure 1. Each node has a name and an existence probability such as *A* and *1.0*. Let us consider a graph proximity search where we look for documents that are similar to a given query document. In Figure 2, node *A* represents a query document. In the initial state, the existence probability of the particle being only node *A* (Figure 2). In the next step, the particle could move to one of three nodes, *B*, *Law-1*, and *Law-4*, that are directly connected to node *A* (Figure 3). In the steady state, the particle could be at any of the nodes (Figure 4), and document *D* is estimated to be the most similar to document *A*. Note that graph random walk does not distinguish document nodes from law article nodes. For each query document, RWR process finds the similar documents and law articles, simultaneously.

Exploiting RWR for legal DSS has at least two strong advantages. First, we can consider similarities based on the global structure of the graph rather than the local structure between two nodes. The steady-state of RWR is globally optimized wherein the estimated similarities are calculated by taking into account the relationships of all documents and law articles. Global optimization often yields more reliable results than local optimization, which only considers the relationships between two nodes.

Second, RWR allows us to capture multi-faceted relationships between two nodes. RWR considers multiple paths between nodes which connote transitivity relations such

as ($\alpha \rightarrow \beta$) and ($\beta \rightarrow \gamma$) lead to ($\alpha \rightarrow \gamma$). For example, in Figure 4, node *B* is directly connected to node *A*, but *D* is not. However, node *D* is finally identified as the most similar to node *A* because it has many indirect paths to *A* such as (*D*→*Law-1*→*A*), (*D*→*Law-4*→*A*) and (*D*→*B*→*A*). The multi-faceted feature relaxes the recall problem of *D-L* edges we mentioned in Section 3.1. Moreover, the transitivity of RWR potentially provides a solution to the problem of legal reform. There are many similar documents which do not share common law articles due to legal reforms.[3] For example, Japanese *River Act* drastically changed in 1964, and the legal cases before and after the change do not share the same law articles. RWR may be able to close gaps between before and after changes to laws.

In addition to the various advantages of precision and recall, RWR also has the advantages of speed. Similarly to other graph-based approaches, RWR consists of matrix calculations. We could use sophisticated matrix calculation to get results faster [33, 13]. In particular, legal DSS requires a large amount of documents and articles, and the diverse acceleration techniques available for matrix calculations would be a big help for legal DSS.

4 Experimental Setup

The goal of our experiments was to answer two questions: (1) How well can our graph-based model address legal document similarity search tasks? (2) What differences are there between the models with and without *D-L* edges? In this section, we present the experimental setup used to answer these questions.

4.1 Used Data

We collected legal documents and law articles. As the legal documents, we collected decision papers from the Supreme Court of Japan.[4] We targeted only the cases of the Supreme Court (not cases in high courts or local courts) and collected 28778 documents. The law articles were collected from those of the Government of Japan (e-Gov).[5] We divided each law into articles systematically by using a finite state transducer (regular expression). Finally, we obtained 197509 articles of 8142 laws.

4.2 Pre-processing

For our graph-based approach, we performed basic natural language pre-processing on the collected decision papers and law articles. Because decision papers were provided in PDF format, we converted them into raw text format by using Poppler.[6] First, we split the documents and articles into sentences with a rule based sentence splitter, which simply breaks the sentences at terminal characters ".", "。", "?", etc. After that, we performed tokenization and POS tagging with a Japanese morphological analyzer MeCab.[7] We only utilized nouns and verbs in the documents and articles to compute the TF-IDF scores, and obtained a document vector for each document. For all pairs of documents, we calculated cosine similarities and filtered out less similar document pairs by using a threshold lower than 0.1. Within the similar document pairs, we extracted

[3] Such pairs of documents could share law articles if we did not have legal reform.
[4] http://www.courts.go.jp/
[5] http://www.e-gov.go.jp/
[6] http://freedesktop.org/wiki/Software/poppler/
[7] http://taku910.github.io/mecab/

Table 3: Specifications of the Constructed Graph

	Document	Article	D-D	D-L
Graph	28778	8599	48853	34795

Table 4: Evaluation Dataset

Class	Category	Case
Constitution	14	95
Civil Law	14	49
Administrative Law	6	57
Total	34	201

the ten most similar documents for each document and made *D-D* edges for them. Note that due to the filtering, there were some documents that had no similar documents.

For the *D-L* edges, we exploited the meta data of the decision papers. The meta data consisted various information on the lawsuits, such as the number of the case, type, the concluded date, and *referenced law articles*. Here, we constructed *D-L* edges by using the *Referenced law articles*. We limited the article nodes which had at least one incoming *D-L* edge. The resulting graph is shown in Table 3. We ran our graph-based legal DSS on this graph.

Graph construction and graph mining require high performance matrix calculations. Here, we used la4j[8], an open source Java library that supports both dense and sparse matrix calculations.

4.3 Evaluation Dataset and Measures

We prepared evaluation data to examine the performance of legal DSS systematically. We collected legal cases for the exam of Japanese administrative scrivener[9] from the Case Database[10] and the Complete Collection of Important Cases [24].

The collected legal cases are divided into three classes of jurisprudence; *Constitution, Civil Law*, and *Administrative Law*. From the legal cases, we compiled the evaluation data by hand. The legal cases of the exam have categories which represent the points at issue such as *standing to sue, national compensation, false manifestation of intention, etc.* In total, we obtained 201 legal cases with 34 categories. Table 4 lists the numbers of categories and legal cases for the three classes. We defined the cases with the same categories to be the correct answers of similar documents.

For each case, we performed legal DSSs and our system output a rank list of ten most similar documents. We used two evaluation measure of information retrieval i.e., *mean reciprocal rank (MRR)* and *mean average precision (MAP)*. Let us consider Table 1 again. There are five cases, and *Q* is a query. Supposing *C1-C4* to be the estimated answers, *C1* is incorrect and *C2, C3, C4* are correct. In this four most similar example, how are MRR and MAP scored? Reciprocal rank (RR) simply evaluates the highest rank of the correct answers and the other lower ranks do not matter. Here, it only considers the second rank (C2) and is calculated as $0.5(=\frac{1}{2})$. MRR is the RR averaged over the all samples. On the other hand, average precision (AP) takes into consideration all four ranks. Precisions are calculated by rank such as $0.00=\frac{0}{1}, 0.5=\frac{1}{2}, 0.67=\frac{2}{3}, 0.75=\frac{3}{4}$. We averaged the four precisions and obtained an AP of $0.48=(0.00+0.5+0.67+$

[8] http://la4j.org/

[9] 行政書士試験 (in Japanese)

[10] http://www.gyosei-i.jp/page007.html (in Japanese)

Table 5: Results of Legal DSS

Class	Measure	*D-D*	*D-L*	*D-D+D-L*
Constitution	MAP	0.038	0.056	0.052
	MRR	0.118	0.173	0.153
Civil Law	MAP	0.038	0.027	0.045
	MRR	0.103	0.068	0.128
Administrative Law	MAP	0.057	0.027	0.080
	MRR	0.152	0.072	0.187

$0.75)/4$. AP was calculated as:

$$\text{AP} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{5}$$

where n is the size of the ranked list, and P_i is the precision at the ith rank. MAP is the AP scores averaged over all the samples. All our experiments were evaluated using the MRR and MAP of the ten most similar ranks.

5 Results

5.1 Results of Experiments

Table 5 shows the results of the legal document similarity search. It lists results for three systems based on the edges we used. Models *D-D* and *D-L* used only D-D and D-L edges, respectively. Model *D-D+D-L* used both D-D and D-L edges. The *MAP* and *MRR* were calculated for each system. Results are shown for; the *Constitution*, *Civil Law*, and *Administrative Law* classes.

First, let us look at the results of Models *D-D* and *D-L* for each class. In Constitution, Model *D-L* outperformed Model *D-D* in terms of both *MAP* and *MRR* (at 0.018 and 0.055 margins). On the other hand, in the other two classes, Model *D-D* achieved the better results. Especially in Administrative Law, Model *D-L* was much weaker than Model *D-D*. Administrative law is not composed of a law code. It is a set of laws related to administration such as "Administrative Procedures Act", "Administrative Case Litigation Act", and "Administrative Execution by Proxy Act". Hence, some cases do not share the same law articles even though they are legally similar. In such cases, the model with only *D-L* edges suffers from too wide a variety of law articles. However, the *D-L* edges we used are far from the perfect. There were many cases that had no referenced law articles. In spite of the missing edges, Model *D-L* was able to handle many cases.

The combination of *D-D* and *D-L* (Model *D-D+D-L*) led to other results. *D-D* edges and *D-L* edges encode the different features of linguistic and legal points. In Civil Law and Administrative Law, Model *D-D+D-L* outperformed Models *D-D* and *D-L* by significant margins. However, in Constitution, Model *D-L* was slightly superior to Model *D-D+D-L*. We found that the cases in Constitution strongly prefer the legal features (*D-L*) to linguistic features (*D-D*). Though the cases in Constitution have large linguistic diversity, the Constitution of Japan consists of only 103 articles. Therefore, capturing legal similarity via law articles is often easier than capturing linguistic similarity via documents. Thus, we confirmed the advantages of using *D-L* edges in legal DSS.

Table 6: Output of Model *D-D* for *National Compensation*

Rank	ID	Case Number	Case Name
Query	(6Q)	昭和 42(オ)921	Claim for damages (高知落石事件)
1	(6a)	平成 8(オ)1248	Claim for clearing obstacles against cars
2	(6b)	昭和 34(オ)117	Claim for damages
3	(6c)	平成 8(オ)1361	Claim for removal of forestall
4	(6d)	昭和 62(オ)741	Claim for removal of workpiece
5	(6e)	平成 4(オ)1504	Regulation for noise and exhaust of Route-43, Hanshin expressway 1
6	(6f)	平成 4(オ)1503	Regulation for noise and exhaust of Route-43, Hanshin expressway 2
7	(6g)	平成 15(受)1886	Claim for removal of workpiece
8	(6h)	昭和 37(オ)108	Claim for damages
9	(6i)	昭和 39(オ)1382	Claim for damages and the counter-suit
10	(6j)	昭和 31(オ)407	Claim for vocating a building

Table 7: Output of Model *D-D+D-L* for *National Compensation*

Rank	ID	Case Number	Case Name
Query	(7Q)	昭和 42(オ)921	Claim for damages (高知落石事件)
1	(7a)	平成 4(オ)1504	Regulation for noise and exhaust of Route-43 Hanshin expressway 1
2	(7b)	平成 3(オ)1534	National compensation
3	**(7c)**	**昭和 63(オ)791**	**Claim for damages (多摩川水害訴訟)**
4	**(7d)**	**昭和 53(オ)492**	**Claim for damages (大東水害訴訟)**
5	(7e)	平成 1(オ)1628	Appeal of claim for damages
6	(7f)	昭和 61(オ)256	Claim for damages and restoration of provisional execution
7	(7g)	昭和 61(オ)255	Claim for damages and restoration of provisional execution
8	**(7h)**	**昭和 47(オ)704**	**Claim for damages (87 時間大型故障車放置事件)**
9	(7i)	昭和 58(オ)1440	Claim for damages
10	(7j)	平成 4(オ)1503	Regulation for noise and exhaust of Route-43 Hanshin expressway 2
13	**(7k)**	**昭和 46(オ)887**	**Claim for damages (奈良赤色灯事件)**
17	**(7l)**	**昭和 58(オ)1132**	**Claim for damages (点字ブロック事件)**

5.2 Discussion

In this section, we discuss our results from a qualitative viewpoint. How does our model capture legal similarities for the purposes of legal DSS? For an answer, let's look at some of outputs and compare the results of Model *D-D* and Model *D-D+D-L*.

The first example is on the cases of *Government Compensation* which are classified in *Administrative Law*. We performed legal DSS by inputting "高知落石事件 (Kōchi Fallen Rock Accident)" as the query document, as described in Section 1. Tables 6 and 7 show the results of Model *D-D* and Model *D-D+D-L*, respectively. We mainly explain each case with *ID* such as (6a) or (7c) shown in the second column in each table. In order to maintain the reproducibility, we left the Japanese *Case Number*, such as "昭和 42(オ)921" or "平成 8(オ)1248", in the original form.

As shown in Table 6, Model *D-D* did not output any correct answers at least in our evaluation dataset.[11] On the other hand, Model *D-D+D-L* output several correct

[11] There could be some correct answers which are not in the evaluation dataset.

Table 8: Output of Model *D-D* for *The Right to Pursue Happiness*

Rank	ID	Case Number	Case Name
Query	(8Q)	平成 1(オ)1649	Claim for solatium (ノンフィクション「逆転」事件)
1	**(8a)**	**平成 15(受)281**	**Claim for damages (肖像権訴訟)**
2	(8b)	昭和 58(オ)516	Claim for damages
3	(8c)	平成 16(受)930	Claim for damages
4	(8d)	平成 12(受)222	Claim for injunction based on infringement of copyright
5	(8e)	昭和 34(オ)780	Claim for use prohibition of the record
6	(8f)	昭和 51(オ)923	Claim for damages
7	(8g)	平成 4(オ)1443	Claim for injunction based on infringement of copyright
8	(8h)	平成 20(受)889	Claim for injunction based on infringement of copyright
9	(8i)	平成 19(受)1105	Claim for injunction based on infringement of copyright
10	(8j)	昭和 59(オ)1204	Claim for injunction based on copyright infringement of the music

Table 9: Output of Model *D-D+D-L* for *The Right of Pursue Happiness*

Rank	ID	Case Number	Case Name
Q	(9Q)	平成 1(オ)1649	Claim for solatium (ノンフィクション「逆転」事件)
1	**(9a)**	**平成 15(受)281**	**Claim for damages (肖像権訴訟)**
2	(9b)	昭和 58(オ)516	Claim for damages
3	(9c)	昭和 41(オ)970	Claim for solatium
4	(9d)	昭和 34(オ)900	Claim for damages
5	(9e)	昭和 30(オ)264	Claim for delivery of the building
6	**(9f)**	**昭和 37(オ)815**	**Claim for damages and solatium of defamation (衆議院議員選挙 立候補者経歴詐称記事事件)**
7	**(9g)**	**昭和 58(オ)1311**	**Claim for published apology (氏名を正確に呼称される権利)**
8	(9h)	平成 6(オ)1084	Claim for damages
9	(9i)	平成 6(オ)715	Claim for damages
10	(9j)	昭和 53(オ)940	Claim for damages

answers such as **(7c) 多摩川水害訴訟**, **(7d) 大東水害訴訟**, and **(7h) 87 時間大型故障車放置事件**. These answers are not linguistically similar but rather legally similar cases. When we made both models output the 20-best answers, Model *D-D+D-L* output the two more correct results but Model *D-D* did not output any more.

The second example is the cases of *The Right to Pursue Happiness*, defined in the Constitution of Japan, Article 13. Model *D-D* output only one correct answers, while Model *D-D+D-L* output three correct answers (Table 8 vs. Table 9). *The Right to Pursue Happiness* has various senses such as *Privacy Rights, Nonsmokers' Rights, Right to light, etc.* Hence, it is very hard to capture similarities from only linguistic features. The query document was a lawsuit about *Privacy Rights* infringed by a book. The outputs of Model *D-D* were dominated by "infringements of copyright" and strayed from the main subject, i.e., *The Right to Pursue Happiness*. The legal features provided by the *D-L* edges improved the bias and captured the diverse senses of the right to pursue happiness, such as *moral right* and *Namensrecht* (in *BGB*).

Thus, the most advantageous aspect of *D-L* edges is their defining clear legal points. Though linguistic similarities have wide varieties, *D-L* edges helped to narrow down the referenced law articles to the main discussion points in terms of jurisprudence.

6 Conclusion

In this paper, we proposed a new graph-based approach to legal document similarity searches (DSS). Our model captures legal similarities through referenced law articles for each legal document. Document similarity and legal similarity are implemented using two types of edges: document-document (*D-D*) and document-law articles (*D-L*). Through our experiments, we evaluated models with and without *D-L* edges. *D-D* edges and *D-L* edges capture different features from each other. Combining *D-D* and *D-L* edges improved performance of legal DSS. A qualitative analysis showed that, the model with *D-L* edges extracted the legally similar documents based on the law articles as evidence.

In the future, we will combine other document representations with our graph-based model. We hope to be able to control the level of legal similarity by changing the representation of the documents and the structure of the graph. We will also try legal DSS in several languages and see if our model works well in various environments beyond the Japanese statute law system.

References

1. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Similarity search using concept graphs. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 719–728. CIKM '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2661829.2661995
2. Ashley, K.D.: Modelling Legal Argument: Reasoning with Cases and Hypotheticals. Ph.D. thesis, University of Massachusetts, Amherst, Amherst, MA, USA (1988), order No: GAX88-13198
3. Ashley, K.D.: Reasoning with cases and hypotheticals in hypo. In: Int. J. ManMachine Studies 34, 753–796 (1991)
4. Ashley, K.D., Walker, V.R.: Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 176–180. ICAIL '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2514601.2514622
5. Bach, N.X., Minh, N.L., Oanh, T.T., Shimazu, A.: A two-phase framework for learning logical structures of paragraphs in legal articles. ACM Transactions on Asian Language Information Processing (TALIP) 12(1), 3:1–3:32 (Mar 2013), http://doi.acm.org/10.1145/2425327.2425330
6. Bach, N.X., Nguyen, M.L., Tran, O.T., Shimazu, A.: Learning logical structures of paragraphs in legal articles. In: Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011. pp. 20–28 (2011), http://aclweb.org/anthology/I/I11/I11-1003.pdf
7. Bench-Capon, T.J.M.: Argument in artificial intelligence and law. Artif. Intell. Law 5(4), 249–261 (1997), http://dx.doi.org/10.1023/A:1008242417011
8. Dean, J., Henzinger, M.R.: Finding related pages in the world wide web. In: Proceedings of the Eighth International Conference on World Wide Web. pp. 1467–1479. WWW '99, Elsevier North-Holland, Inc., New York, NY, USA (1999), http://dl.acm.org/citation.cfm?id=313234.313077
9. Ferrucci, D.A.: Ibm's watson/deepqa. SIGARCH Computer Architecture News 39(3) (2011), http://doi.acm.org/10.1145/2024723.2019525
10. Ferrucci, D.A.: Introduction to "this is watson". IBM Journal of Research and Development 56(3), 1 (2012), http://dx.doi.org/10.1147/JRD.2012.2184356

11. Ferrucci, D.A., Levas, A., Bagchi, S., Gondek, D., Mueller, E.T.: Watson: Beyond jeopardy! Artificial Intelligence 199, 93–105 (2013), http://dx.doi.org/10.1016/j.artint.2012.06.009
12. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.): Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language, Lecture Notes in Computer Science, vol. 6036. Springer (2010), http://dx.doi.org/10.1007/978-3-642-12837-0
13. Fujiwara, Y., Nakatsuji, M., Onizuka, M., Kitsuregawa, M.: Fast and exact top-k search for random walk with restart. Proc. VLDB Endow. 5(5), 442–453 (Jan 2012), http://dx.doi.org/10.14778/2140436.2140441
14. Höfler, S., Sugisaki, K.: From drafting guideline to error detection: Automating style checking for legislative texts. In: EACL 2012 Workshop on Computational Linguistics and Writing. pp. 9–18. Association for Computational Linguistics (April 2012), http://dx.doi.org/10.5167/uzh-62049
15. Jiang, Q., Sun, M.: Semi-supervised simhash for efficient document similarity search. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 93–101. Association for Computational Linguistics, Portland, Oregon, USA (June 2011), http://www.aclweb.org/anthology/P11-1010
16. Kimura, Y., Nakamura, M., Shimazu, A.: Treatment of legal sentences including itemized and referential expressions : Towards translation into logical forms. Lecture Notes in Computer Science 5447, 242–253 (2009), http://ci.nii.ac.jp/naid/120001856645/
17. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity graphs in networks. ACM Trans. Knowl. Discov. Data 1(3) (Dec 2007), http://doi.acm.org/10.1145/1297332.1297336
18. Lizorkin, D., Velikhov, P., Grinev, M., Turdakov, D.: Accuracy estimate and optimization techniques for simrank computation. The VLDB Journal 19(1), 45–66 (Feb 2010), http://dx.doi.org/10.1007/s00778-009-0168-8
19. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
20. Matveeva, I.: Document representation and multilevel measures of document similarity. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Doctoral Consortium. pp. 235–238. Association for Computational Linguistics, New York City, USA (June 2006), http://www.aclweb.org/anthology/N/N06/N06-3007
21. Moens, M.F., Boiy, E., Palau, R.M., Reed, C.: Automatic detection of arguments in legal texts. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law. pp. 225–230. ICAIL '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1276318.1276362
22. Montemagni, S., Peters, W., Tiscornia, D.: Semantic Processing of Legal Texts. Springer (2010)
23. Nakamura, M., Ohno, T., Ogawa, Y., Toyama, K.: Acquisition of hyponymy relations for agricultural terms from a japanese statutory corpus. Information Processing in Agriculture 1(2), 95–104 (2014)
24. Nishimura, K.: Perfect Collection of Important Cases. Jyutaku Shimposha (2015)
25. Palau, R.M., Moens, M.F.: Argumentation mining: The detection, classification and structure of arguments in text. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law. pp. 98–107. ICAIL '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1568234.1568246

26. Pan, J.Y., Yang, H.J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 653–658. KDD '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1014052.1014135
27. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proceedings of the 10th International Conference on Artificial Intelligence and Law. pp. 85–94. ICAIL '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1165485.1165500
28. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (Nov 1975), http://doi.acm.org/10.1145/361219.361220
29. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
30. Sartor, G.: Legal Reasoning: A Cognitive Approach to Law. Springer (2005)
31. Tong, H., Faloutsos, C.: Center-piece subgraphs: Problem definition and fast solutions. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 404–413. KDD '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1150402.1150448
32. Tong, H., Faloutsos, C., Koren, Y.: Fast direction-aware proximity for graph mining. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 747–756. KDD '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1281192.1281272
33. Tong, H., Faloutsos, C., Pan, J.Y.: Fast random walk with restart and its applications. In: Proceedings of the Sixth International Conference on Data Mining. pp. 613–622. ICDM '06, IEEE Computer Society, Washington, DC, USA (2006), http://dx.doi.org/10.1109/ICDM.2006.70
34. Tran, O.T., Ngo, B.X., Nguyen, M.L., Shimazu, A.: Automated reference resolution in legal texts. Artif. Intell. Law 22(1), 29–60 (Mar 2014), http://dx.doi.org/10.1007/s10506-013-9149-8
35. Walton, D.: Argumentation Methods for Artificial Intelligence in Law. Springer-Verlag, Berlin, Heidelberg (2010)
36. Wan, X., Yang, J., Xiao, J.: Towards a unified approach to document similarity search using manifold-ranking of blocks. Inf. Process. Manage. 44(3), 1032–1048 (May 2008), http://dx.doi.org/10.1016/j.ipm.2007.07.012

COLIEE-2015 : Evaluation of Legal Question Answering

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

University of Alberta, Canada[1]
National Institute of Informatics, Japan[2]
{miyoung2, rgoebel}@ualberta.ca, ksatoh@nii.ac.jp

Abstract. We present the details on evaluation of legal question answering of the Competition on Legal Information Extraction/Entailment (COLIEE)-2015. The COLIEE-2015 task consists of three sub-tasks: legal information retrieval (sub-task 1), recognizing entailment between articles and queries (sub-task 2), and combination of the previous two sub-tasks (sub-task 3). Participation was open to any group based on any approach, and the task attracted 6 teams. We received 6 submissions to the sub-task 1 (for a total of 12 runs), 4 submissions to the subtask 2 (for a total of 11 runs), and 4 submissions to the subtask 3 (for a total of 11 runs).

Keywords: legal question answering, recognizing textual entailment, information retrieval

1. Introduction

The Juris-Informatics workshop series has been created to promote community discussion on both fundamental and practical issues from various backgrounds such as law, social science, information and intelligent technology, logic and philosophy, including the conventional "AI and law" area.

The information extraction/reasoning from legal data is one of the important targets of JURISIN, including legal information representation, relation extraction, textual entailment, summarization, and their applications. Participants in the JURISIN workshops have examined a wide variety of information extraction techniques and environments with a variety of purposes, including retrieval of relevant articles, entity/relation extraction from legal cases, reference extraction from legal cases, finding relevant precedents, summarization of legal cases, and legal question answering.

In 2014, we held the first competition on legal information extraction/entailment (COLIEE-2014) on a legal data collection, and it helped establish a major experimental effort in the legal information extraction/retrieval field. We held the second competition (COLIEE-2015) this year, with the motivation of continuing to help create a research community of practice for the capture and use of legal information. The COLIEE competition is about the legal question answering task, and our decomposition into three subtasks of retrieval, information extraction, and entailment.

2. The Legal Question Answering Task

This competition focuses on two aspects of legal information processing related to answering yes/no questions from Japanese legal bar exams (the relevant data sets have been translated from Japanese to English).

1) Phase 1 of the legal question answering task involves reading a legal bar exam question Q, and extracting a subset of Japanese Civil Code Articles S_1, S_2,..., S_n from the entire Civil Code, which are appropriate for answering the question such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and the entire Civil Code Articles, we have to retrieve the set of "S_1, S_2, ..., S_n" as the outcome of phase 1.

2) Phase 2 of the legal question answering task involves the identification of an entailment relationship such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and relevant articles S_1, S_2, ..., S_n, we have to determine if the relevant articles entail "Q" or "not Q". The answer of this track is binary: "YES"("Q") or "NO"("not Q"). This phase typically require some information extraction (e.g., named entity identification, relation extraction), followed by methods for textual inference to confirm entailments. Details are given in the next section.

2.1 Phase 1 Details

Our goal is to explore and evaluate legal document retrieval technologies that are both effective and reliable. The task investigates the performance of systems that search a static set of civil law articles using previously unseen queries, and return relevant articles. We say an article is "Relevant" to a query if and only if the query sentence can be entailed from the meaning of the article. If combining the meanings of more than one article (e.g., "A", "B", and "C") can answer a query sentence, then all the articles ("A", "B", and "C") are considered "Relevant." If a query can be answered by an article "D", and it can be also answered by another article "E" independently, we also consider both "D" and "E" are "Relevant". This task requires the retrieval of all the articles that are relevant to answering a query.

Japanese civil law articles (and English translation) have been provided, and training data consists of pairs of a query and relevant articles. The process of executing the queries over the articles and generating the experimental runs should be entirely automatic. Test data includes only queries but no relevant articles.

2.2 Phase 2 Details

The goal of Phase 2 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. The task investigates the performance of systems that answer "Y" or "N" to previously unseen queries by comparing the intersection of meanings between queries and relevant articles. Training data consists of triples: a query, relevant articles and a correct answer "Y" or "N." The process of executing the queries over the relevant articles and generating the experimental runs should be entirely automatic. Test data includes only queries and relevant articles, but no "Y/N" label.

2.3 Phase 3 Details

Phase 3 combines Phase 1 and Phase 2. If a "Yes/No" legal bar exam question is given, a legal information retrieval system retrieves relevant Civil Law articles. Then, the task investigates the performance of systems that answer "Y" or "No" to previously unseen queries by comparing the meanings of the queries and retrieved Civil Law articles. Test data includes only queries, but no "Y/N" label, and no relevant articles.

3. The Legal Question Answering Data Corpus

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

1) Phase 1 problem is to use an identified set of legal yes/no questions to retrieve relevant Civil Law articles. In this case, the correct answers have been determined by a collection of law students, and those answers are used to calibrate the performance of a program to solve Phase 1.

2) Phase 2 task requires some method of information extraction from both the question and the relevant articles, and then to confirm a simple entail relationship as described in the Section 2: either the articles confirms "yes" or "no" as an answer to the yes/no questions.

3) Phase 3 task is combination of Phase 1 and Phase 2. It requires both of the legal information retrieval system and textual entailment system. If legal yes/no questions are given, each participant's legal information retrieval system retrieves relevant Civil Law articles. After that, each participant confirms a "Yes/No" entailment relationship between input yes/no question and the retrieved articles.

Participants can choose which Phase they will attempt, amongst the three sub-tasks as follows:

1. Sub-task 1: legal information retrieval Task. Input is a bar exam "Yes/No" question and output should be relevant civil law articles. (Phase 1)

Table 1. Example of a query and a relevant article

Question	*A person who made a manifestation of intention which was induced by duress emanated from a third party may rescind such manifestation of intention on the basis of duress, only if the other party knew or was negligent of such fact.*
Related Article	*(Fraud or Duress) Article 96 (1)Manifestation of intention which is induced by any fraud or duress may be rescinded.(2)In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.(3)The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.*

Table 2. Examples of sentence pairs holding different entailment relations

Question	A special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty.
Related Article	(Special Agreement Disclaiming Warranty)Article 572 Even if the seller makes a special agreement to the effect that the seller will not provide the warranties set forth from Article 560 through to the preceding Article, the seller may not be released from that responsibility with respect to any fact that the seller knew but did not disclose, and with respect to any right that the seller himself/herself created for or assigned to a third party.
Label	Yes
Question	A manager must engage in management exercising care identical to that he/she exercises for his/her own property.
Related Article	(Urgent Management of Business)Article 698 If a Manager engages in the Management of Business in order to allow a principal to escape imminent danger to the principal's person, reputation or property, the Manager shall not be liable to compensate for damages resulting from the same unless he/she has acted in bad faith or with gross negligence.
Label	No

2. Sub-task 2: Recognizing Entailment between law articles and queries. Input is a pair of a question and relevant article(s), and output should be “Yes” or “No. (Phase 2)

3. Sub-task 3: Combination of sub-task 1 and sub-task 2. Input is a bar exam “Yes/No”question and output should be “Yes” or “No.” (Phase 3)

Table 1 shows an example of query and relevant articles.

In the entailment subtask, given two sentences A and B, a system must determine whether the meaning of B was entailed by A. In particular, a system is required to assign to each pair either the “Yes” label (when A entails B, viz., B cannot be false

when A is true), or the “No” label (when A does not entail B). Table 2 shows examples of sentence pairs holding different entailment relations.

The subtask 3 is combination of subtask 1 and subtask 2.

The structure of the test corpora is derived from a general XML representation developed for use in RITEVAL, one of the tasks of the NII Testbeds and Community for Information access Research (NTCIR) project, as described at the following URL:

http://sites.google.com/site/ntcir11riteval/

The RITEVAL format was developed for the general sharing of information retrieval on a variety of domains. The format of the JURISIN competition corpora derived from an NTCIR representation of confirmed relationships between questions and the articles and cases relevant to answering the questions, as in the following example:

<pair label="Y” id="H18-1-2">
<t1>
(Seller's Warranty in cases of Superficies or Other Rights)Article 566 (1)In cases where the subject matter of the sale is encumbered with for the purpose of a superficies, an emphyteusis, an easement, a right of retention or a pledge, if the buyer does not know the same and cannot achieve the purpose of the contract on account thereof, the buyer may cancel the contract. In such cases, if the contract cannot be cancelled, the buyer may only demand compensation for damages. (2)The provisions of the preceding paragraph shall apply mutatis mutandis in cases where an easement that was referred to as being in existence for the benefit of immovable property that is the subject matter of a sale, does not exist, and in cases where a leasehold is registered with respect to the immovable property.(3)In the cases set forth in the preceding two paragraphs, the cancellation of the contract or claim for damages must be made within one year from the time when the buyer comes to know the facts.
(Seller's Warranty in cases of Mortgage or Other Rights)Article 567(1)If the buyer loses his/her ownership of immovable property that is the object of a sale because of the exercise of an existing statutory lien or mortgage, the buyer may cancel the contract.(2)If the buyer preserves his/her ownership by incurring expenditure for costs, he/she may claim reimbursement of those costs from the seller.(3)In the cases set forth in the preceding two paragraphs, the buyer may claim compensation if he/she suffered loss.
</t1>
<t2>
There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.
</t2>
</pair>

The above is an example where query id “H18-1-2” is confirmed to be answerable from article numbers 566 and 567 (relevant to Phase 1). The pair label “Y” in this

example means the answer of query is “Yes", which is entailed from the relevant articles (relevant to Phase 2 and 3).

COLIEE training data was built from the bar exam (short answer test) civil code part published in 2006-2011, and consist of 6 XML files. Each file corresponds to one year’s publication. The total number of queries in the training data is 267, and each query has average 1.25 relevant articles. The test data size is 79 queries for Phase 1, extracted from the bar exam of 2012, and 66 queries for Phase 2 from the bar exam of 2013. For Phase 3, we use the same test data of Phase 1.

4. Evaluation Metrics and Baselines

The measures for ranking competition participants are intended only to calibrate the set of competition submissions, rather than provide any deep performance measure. The data sets for Phases 1 and 2 are annotated, so simple information retrieval measures (precision, recall, F-measure [7], accuracy) can be used to rank each submission. As noted above, the intention is to start to build a community of practice regarding legal textual entailment, so that the adoption and adaptation of general methods from a variety of fields is considered, and that participants share their approaches, problems, and results.

For Phase 1, evaluation measure will be precision, recall and F-measure:

Precision =(the number of correctly retrieved articles for all queries)/(the number of retrieved articles for all queries),
Recall = (the number of correctly retrieved articles for all queries)/(the number of relevant articles for all queries),
F-measure = (2 x Precision x Recall)/(Precision + Recall).

For Phase 2, the evaluation measure will be accuracy, with respect to whether the yes/no question was correctly confirmed:

Accuracy = (the number of queries which were correctly confirmed as true or false)/(the number of all queries).

For Phase 3, the evaluation measure is the same with that of Phase 2.

For Phase 1, we consider the traditional TF-IDF [6] as the baseline model. For Phase 2, there is a balanced positive-negative sample distribution in the dataset (51.52% yes, and 48.48% no) for the formal run, so we consider the baseline for true/false evaluation is the accuracy when returning always “yes,” which is 51.52%.
Tables 3 and 4 presents the performances of the baseline models.

Table 3. Baseline of Phase one

Phase One Baseline	F-measure
TFxIDF with lemmarization (choosing the most relevant article)	0.4862

Table 4. Baseline of Phase two

Phase Two Baseline	Accuracy
When 'No' labels are all chosen	0.5152

5. Submitted Runs and Results

Overall, 6 teams participated in the task. Some participants submitted multiple runs for a task. We received 6 submissions to the subtask 1 (for a total of 12 runs), 4 submissions to the subtask 2 (for a total of 11 runs), and 4 submissions to the subtask 3 (for a total of 11 runs).

Tables 5 and 6 show the submitted systems' techniques. Because we have not received one participant's paper, the tables include 5 teams' runs and their techniques. We present the results achieved by runs against the Information Retrieval and Entailment subtasks in Tables 7, 8, and 9, respectively. Some systems performed well above the best baselines from Tables 3 and 4.

D. S. Carvalho et al. [2] extracted n-gram features from the query and articles using lexical and morphological characteristics, and then used their own relevance score for Phase 1. For Phase 2, the query and relevant article are represented in vector space model, and then they used AdaBoost (basic classifier: DecisionStump) to classify the queries between 'Yes' label and 'No' label.

Sushimita et al. [3] participated in only Phase 1, and used voting of three information retrieval techniques: Hiemstra, BM25 and PL2F.

Y. Kano [4] introduced their own keyword weighting method and snippet scoring after diving data into snippets . V.D.Tran et al. [5] participated in only Phase 1 and used Hidden Markov model (HMM) as a generative query model for legal information retrieval.

Table 5. Approaches of submitted systems for IR (Phase 1)

ID	Run	Approaches
UA [1]	UA1	Ranking SVM
	UA2	TFxIDF with lemmarization
Kano-lab[4]	Kanolab1	their own keyword weighting method snippet scoring (ptag=0.3)
	Kanolab2	their own keyword weighting method snippet scoring (ptag=0.4)
	Kanolab3	their own keyword weighting method snippet scoring (ptag=0.5)
	Kanolab4	their own keyword weighting method snippet scoring (topic=0.1)
	Kanolab5	their own keyword weighting method snippet scoring (topic=0.3)
JAIST [2]	JAIST01	Ranking by n-gram model
ALV [5]	ALV2015	Hidden Markov Model
	ALV2015-2	Hidden Markov Model with different parameter setting
Ismir [3]	ismir	Voting of three models: Hiemstra, BM25 and PL2F

Table 6. Approaches of submitted systems for the Entailment (Phase 2)

ID	Run	Approaches
UA [1]	UA	Convolutional Neural Network
Kano-Lab [4]	Kanolab1	Assigning Threshold to the snippet score (ptag=0.3)
	Kanolab2	Assigning Threshold to the snippet score (ptag=0.4)
	Kanolab3	Assigning Threshold to the snippet score (ptag=0.5)
	Kanolab4	Assigning Threshold to the snippet score (Section=0.1)
	Kanolab5	Assigning Threshold to the snippet score (Section=0.2)
	Kanolab6	Assigning Threshold to the snippet score (Section=0.4)
	Kanolab7	Assigning Threshold to the snippet score (Topic=0.1)
	Kanolab8	Assigning Threshold to the snippet score (Topic=0.3)
JAIST [2]	JAIST01	Adaboost (classifier: DecisionStump) using distance features and statistical features

The UA2 method in Table 7 uses TFxIDF with lemmatization which is the baseline

Table 7. IR results(Phase 1) on the formal run data

Run	Prec.	Recall	F-m.	Run	Prec.	Recall	F-m.
UA1 [1]	0.6329	0.4902	0.5525	Kanolab5[4]	0.3038	0.2353	0.2652
UA2 [1]	0.5570	0.4314	0.4862	JAIST01[2]	0.5663	0.4608	0.5081
Kanolab1[4]	0.3291	0.2549	0.2873	ALV2015[5]	0.3418	0.5294	0.4154
Kanolab2[4]	0.3291	0.2549	0.2873	ALV2015-2[5]	0.2250	0.3529	0.2748
Kanolab3[4]	0.3291	0.2549	0.2873	Ismir[3]	0.2778	0.2941	0.2857
Kanolab4[4]	0.3038	0.2353	0.2652				

Table 8. Entailment results (Phase 2) on the formal run data

Run	Accu.	Run	Accu.
UA [1]	0.6667	Kanolab5 [4]	0.5152
Kanolab1 [4]	0.6061	Kanolab6 [4]	0.5152
Kanolab2 [4]	0.6212	Kanolab7 [4]	0.5152
Kanolab3 [4]	0.6212	Kanolab8 [4]	0.5152
Kanolab4 [4]	0.5152	JAIST01 [2]	0.3788

Table 9. Combined results (Phase 3) on the formal run data

Run	Accu.	Run	Accu.
UA [1]	0.6582	Kanolab5 [4]	0.4937
Kanolab1 [4]	0.5696	Kanolab6 [4]	0.5317
Kanolab2 [4]	0.5949	Kanolab7 [4]	0.4557
Kanolab3 [4]	0.6203	Kanolab8 [4]	0.5696
Kanolab4 [4]	0.4684	JAIST01 [2]	0.5823

model. We can see that there are two systems which show better performance than the baseline model: UA1 and JAIST01. UA1 shows the best performance using ranking SVM, and it achieved significant increase of the F-measure compared to the baseline model. JAIST01 used their own ranking score from n-gram model. UA1, UA2 and Kanolab produced only one most relevant article for each query (in total 79 articles). JAIST01 produced 82 articles, and ALV2015 and ALV2015-2 produced 157 articles and 160 articles respectively. The reason that ALV2015 shows best recall is that they produced 157 results (average 1.99 articles per query), but their precision was not good. Ismir also produced 108 results, which is average 1.37 articles for each query.

Table 8 reports the results of Phase 2, where UA showed best accuracy, which is significantly better than the baseline performance. They used a convolutional neural network. Kanolab1, Kanolab2 and Kanolab3 also showed significantly increased

performance than the baseline, and they used their own threshold value from the snippet scoring to determine if the label is “yes” or “no.”

In the Table 9, results of Phase 3, Kanolab6, Kanolab8, and JAIST01 showed better performance than their performances in Phase 2. That means their answering of “yes” “no” questions is better when they used their own relevant articles than those marked as correct in the training set. This indicates the need for further detailed investigation.

6. Conclusion

We presented the results of the COLIEE-2015 competition. Three subtasks were offered: (1) Phase 1: finding relevant articles (information retrieval) (2) Phase 2: answering yes/no questions by comparing the meanings between a query and relevant articles (textual entailment), (3) combination of Phase 1 and Phase 2. There were 6 submissions to Phase 1 (for a total of 12 runs), 4 submissions to Phase 2 (for a total of 11 runs), and 4 submissions to Phase 3 (for a total of 11 runs), for a total of 6 participating teams. Various methods for Phase 1 are provided: ranking SVM, TFxIDF, n-gram features from the query and articles using lexical and morphological characteristics and their own relevance score. Some information retrieval techniques: Hiemstra, BM25 and PL2F, keyword weighting method and snippet scoring, and Hidden Markov Model have also been proposed. For Phase 2, the participants used AdaBoost using their own features, a convolutional Neural Network, and their own threshold from the snippet score. Most of the systems do not depend on the deeply linguistic features which require domain knowledge, but instead propose feature weighting and scoring methods. Some systems significantly outperformed baseline systems, and we need to deeply investigate how the performances of each Phase 1 and Phase 2 affect the overall performance of Phase 3.

Acknowledgements

This research was supported by the Alberta Innovates Centre for Machine Learning (AICML), the Natural Sciences and Engineering Research Council (NSERC), and also partially supported by JSPS KAKENHI Grant Numbers 26280091.

References

1. M-Y. Kim, Y. Xu, and R. Goebel, “A Convolutional Neural Network in Legal Question Answering”, Ninth International Workshop on Juris-informatics (JURISIN), 2015
2. D. S. Carvalho, M-T. Nguyen, C-X. Tran, and M-L. Nguyen, “Lexical-Morphological Modeling for Legal Text Analysis”, Ninth International Workshop on Juris-informatics (JURISIN), 2015
3. Sushmita, A. Kanapala, and S. Pal, “Legal Information Retrieval Task: Participation from ISM, Dhanbad”, Ninth International Workshop on Juris-informatics (JURISIN), 2015
4. Y. Kano, “Keyword and Snippet Based Yes/No Question Answering System for COLIEE 2015”, Ninth International Workshop on Juris-informatics (JURISIN), 2015

5. V. D. Tran, A. V. Phan, L. H. Trieu, and M. L. Nguyen, “An Approach for Retrieving Legal Texts”, Ninth International Workshop on Juris-informatics (JURISIN), 2015
6. K. J. Sparck. “A statistical interpretation of term specificity and its application in retrieval.” Journal of documentation 28.1, 11-21, 1972
7. G. Hripcsak, and A. S. Rothschild. “Agreement, the f-measure, and reliability in information retrieval”. Journal of the American Medical Informatics Association, 12(3), 296-298, 2005

Lexical-Morphological Modeling for Legal Text Analysis

Danilo S. Carvalho[1⋆], Minh-Tien Nguyen[1,2], Chien-Xuan Tran[1], and Minh-Le Nguyen[1]

[1] School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan
[2] Hung Yen University of Education and Technology (UTEHY), Hung Yen, Vietnam
{danilo, tiennm, chien-tran, nguyenml}@jaist.ac.jp

Abstract. In the context of the Competition on Legal Information Extraction/Entailment (COLIEE), we propose a method comprising the necessary steps for finding relevant documents to a legal question and deciding on textual entailment evidence to provide a correct answer. The proposed method is based on the combination of several lexical and morphological characteristics, to build a language model and a set of features for Machine Learning algorithms. We provide a detailed study on the proposed method performance and failure cases, indicating that it is competitive with state-of-the-art approaches on Legal Information Retrieval and Question Answering, while not needing extensive training data nor expert produced knowledge.

1 Introduction

Answering legal questions has been a long-standing challenge in the Information Systems research landscape. A topic that is drawing progressively more attention, as we experience an explosive growth in legal document availability on the World Wide Web and specialized systems. This growth is not accompanied by a matching increase in information analysis capabilities, which points to a severe under-utilization of the available resources and a potential for information quality issues [1]. As a consequence, professionals of law, in particular, have been put into increasingly pressure, since having the relevant and correct information is a vital step in legal case solving and thus is closely tied to the matter of professional ethics and liability. This problem is often referred as "information crisis" of law.

The ability to retrieve relevant and correct information given a legal query has improved over time, with the combination of expert Knowledge Engineering and NLP methods. On the other hand, the ability to answer questions in the legal domain is of special difficulty, due to the need of reasoning over different types of information, such as past decisions, laws and facts. Furthermore, the way legal concepts are applied in the language often differs from common usage,

⋆ Supported by CNPq – Brazil scholarship grant

and differences in laws and procedures from each country prevent the creation of comprehensive and coherent international law corpora. Common legal ontologies are among the efforts to facilitate automatic legal reasoning, but have not seen strong development in the past years [2]. In this context, *Textual Entailment Recognition* plays a very important role, as a set hypothesis presented in a question will most certainly have answers in the previously cited types of information (decisions, laws, facts). The Recognition of Textual Entailment (RTE) challenge series [3], although not specific to the legal domain, is a recognized benchmark for methods that can be adapted to legal texts.

To effectively answer legal questions, one fundamental set of information that must be available is the law, presented as the collection of codes, sections, articles and paragraphs that should be unequivocally referenced when a hypothesis is raised as part of a legal inquiry. Therefore, adequate representation of law corpora is the basis of a functional system for legal question answering. The representation problem is often associated with ontologies and other annotated knowledge bases, but these methods are costly and more difficult to automate when compared to fully text-based approaches, such as bag-of-words, n-gram and topic models.

In this work, we propose a fully text-based method for legal text analysis, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE), covering both the tasks of Information Extraction and Question Answering. The goal is the retrieval of relevant law articles to a given yes/no legal question and the use of the retrieved articles to correctly answer the question in a completely automated way. The proposed method is based on a broad lexical and morphological analysis of the English translated Japanese Civil Code, comprising tokenization, POS-tagging, lemmatization, word clustering and a set of lexical statistics. Such analysis allows the construction of a mixed size n-gram model and a set of features appropriate for Machine Learning algorithms. The n-gram model is used for Relevance Analysis on the legal corpus, with respect to the queries provided, and the Machine Learning features are used for Textual Entailment classification. A study on success and fail cases is provided, with common baseline practices and related works used as means of performance comparison.

The remaining of this work is structured as follows: Section 2 presents the related works and relevant results; Section 3 details the Legal Question Answering problem and the COLIEE competition shared task; Section 4 explains our approach to the competition problem; Section 5 presents the experimental setting, results and discussion; Finally, Section 6 offers some concluding remarks.

2 Related Works

Liu, Chen and Ho [3] presented the three-phase prediction (TPP) method for retrieval of relevant statutes in Taiwan's criminal law, given general language queries. The method is a hierarchical ranking approach to law corpora, featuring

[3] www.aclweb.org/aclwiki/index.php?title=Recognizing_Textual_Entailment

a combination of several Information Retrieval techniques, as well as Machine Learning and Feature selection ones. Results were evaluated in terms of recall, achieving from 0.52 to 0.91, from the top 3 to 10 retrieved results, respectively.

Inkpen et al. showed one of the first successful models for RTE by SVMs [4]. Later, Castillo proposed a system for solving RTE by using SVMs, in which training data includes RTE-3, annotated data set of RTE-4, and development set of RTE-5 [5]. 32 features were used and the training model achieved the best F-measure of 0.69 in two-way and 0.67 in three-way task. Most of features from [5] were used in this study along with *Word2Vec*[16] similarity instead of *WordNet*.

Nguyen et al. [6] conducted a study of RTE on a Vietnamese version of RTE-3 [7] translated from [8]. The author used SVMs based on 15 features including two groups: distance and statistical features. A voting combined three classifiers built on three feature groups (distance, statistical, and combined features) was used to judge entailment relation. The method obtained 0.684 of F-measure in two-way task. In this study, we use all features in [6] and add additional features: Word2Vec, term frequency - inverse document frequency (TF-IDF), or rules.

Research dedicated to Question Answering (QA) in legal text has seen less attention when compared to general QA. Tran et al. solved legal text QA by using inference [9]. The author used requisite-effectuation structures of legal sentences and similarity measures to find out correct answers without training data and achieved 60.8% of accuracy with 51 articles on Japanese National Pension Law.

Kim et al. proposed a hybrid method containing simple rules and unsupervised learning using deep linguistic features to solve RTE in civil laws [10]. The author also constructed a knowledge base for negation and antonym words which would be used for classifying simple questions. To deal with difficult questions, the author used morphological, syntactic and lexical analysis to identify premises and conclusions. The accuracy was 68.36% with easy questions and 60.02 with difficult ones.

3 Legal Question Answering

Legal Question Answering (LQA) is to find out and provide *"correct answers"* given by a legal question for users. An overview of LQA is shown in Fig. 1.

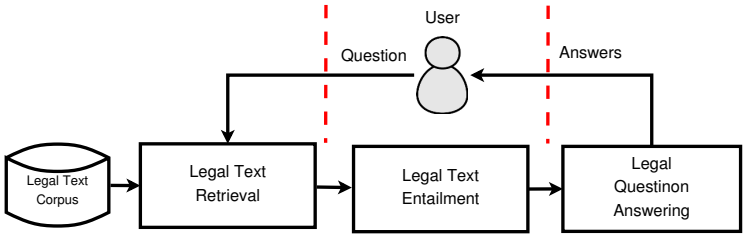


Fig. 1. The model of legal text question answering system

LQA can be described in three tasks: 1) retrieving relevant articles, i.e., the ones containing the answer; 2) finding correct evidence in the relevant articles that allows answering the question; and 3) answering the question. While the first task is essentially IR, the second can be considered as a form of RTE, in which given a question, a LQA system has to decide whether and how a relevant article can answer the question. The final one is the combination of the two previous tasks.

Legal texts are very different in comparison to general ones e.g., new articles due to their characteristics. Firstly, they have specific logical sentence structures e.g., requisite and effectuation [11]. Secondly, words and writing style are used in a strict form because law documents require high correctness and should avoid ambiguity. An alternative aspect is that law documents are written in a highly abstract level [12]; therefore, they usually require reference of readers to understand law articles and to answer a law question. The use of references leads to a situation in which there is not much, or in some cases, even no word overlapping between a law question and its relevant articles.

In this work, LQA tasks are considered into the context of COLIEE, a competition on legal information extraction/entailment which was first held in 2014, in association with Workshop on Juris-informatics (JURISIN). COLIEE 2015 [4] is the second competition consisting of three phases:

- Phase One: retrieving relevant articles from all Japanese Civil Code Articles given a set of YES/NO questions.
- Phase Two: confirming the entailment relationship between the question and retrieved articles.
- Phase Three: combination of Phase One and Phase Two, the system will retrieve list of relevant articles given a query, and then decide the entailment relationship between retrieved articles and provided question.

The Japanese Civil Code is composed by a collection of numbered articles, each one containing a set of declarations pertaining to a specific topic of the law, e.g., labor contracts, mortgages.

Information Retrieval Task: Relevance Analysis

The first phase consists on an explicit IR task, for which the goal is to retrieve the relevant articles that can be used to correctly answer a given yes/no question. The challenge in this task is to determine the relative relevance, i.e., Relevance Analysis (RA), of an article to the query presented in the question. Different articles dealing with the same topic often have similar wording and is common for questions not to refer to topic keywords or refer to alternative versions of them. Furthermore, the restricted size of Japanese Civil Code means that obtaining reliable linguistic information from the articles is difficult and most questions will present new language structures that can range from useful to necessary for answering.

[4] webdocs.cs.ualberta.ca/m̃iyoung2/COLIEE2015/

Simple Question Answering Task: Textual Entailment

The goal of Textual Entailment (TE) is to decide whether a legal query/question can be answered by a set of relevant articles retrieved by RA. This task can be accomplished by recognizing textual entailment (RTE), in which the query/question is treated as an hypothesis and relevant articles are textual information. Given a question Q and a set of relevant articles A, ($A = \{a_1, ..., a_n\}$), if Q is answered by a_i ($1 \leqslant i \leqslant n$), then a_i entails Q [8,13] and a pair *(Q, a_i)* is assigned label YES; otherwise is NO. Assuming only relevant articles are provided, the challenge in this task is to identify the textual features that characterize conforming and conflicting information. To this end, the linguistic challenges of phase one also apply.

4 Proposed Approach

In order to be able to perform both Relevance Analysis and textual entailment independently in phases one and two, and jointly in phase three, IR and classifier methods were developed separately. First, both the legal corpus and training data are analyzed and combined into representation models. The models are then used to rank articles or classify answers according to the task. The representation model used for Relevance Analysis is a mixed size n-gram model and the one used for textual entailment is a multi-feature vector for Machine Learning. Figure 2 shows the overall view of the proposed method.

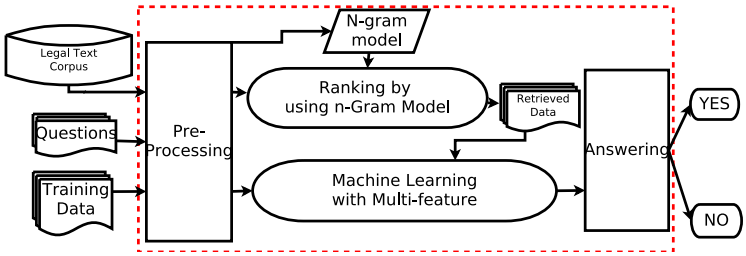


Fig. 2. Model overview

4.1 Relevance Analysis

A detailed analysis of the Civil Code and training data revealed that lexical and syntactic overlapping may vary to a high degree between questions and articles, and also between articles concerning the same topic. However, certain morphological features, such as lemmas, retain a higher level of consistency among topics. For this reason, the adopted representation model was a mixed size n-gram model, with $n : [1, k]$, i.e., terms made by sequences up to k words, in which the terms are lemmatized. For simplicity, the Relevance Analysis method hereon described was named R_2NC (Ranking Related N-gram Collections). A summarized view of the process is shown on Figure 3.

The steps to build the model are detailed as follows:

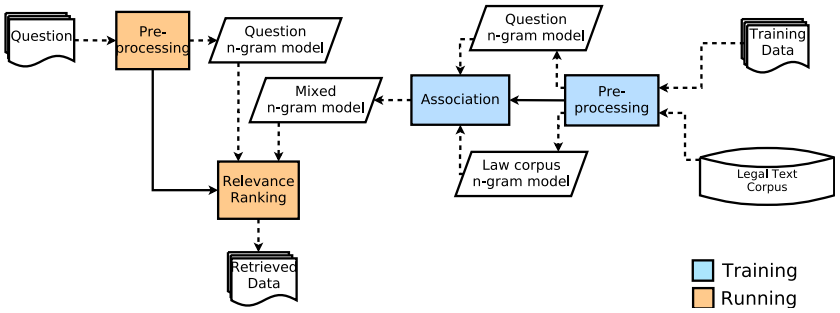


Fig. 3. The process of legal text retrieval

1. Collect the entire content for each article, including section title;
2. Check references between articles and annotate accordingly;
3. Tokenize and POS-tag;
4. Remove stopwords;
5. Lemmatize words;
6. Generate n-grams;
7. Expand the n-gram set, by including references n-grams;
8. Associate article number and references;
9. Store the model.

Except for step 4, each step is responsible for adding new information to the model. The information is obtained either from the text, e.g., section title, references, or from morphological analysis, e.g., POS-tags, lemmas. If an article have references, its n-gram set is expanded with the references' n-grams. This is done so that all the necessary information for interpretation of any single article is self-contained. Besides the n-grams, links between the articles are also stored. To include the training data information, the same process is repeated for the questions, and n-gram sets of the trained questions are used to expand the associated articles n-gram model.

To determine the relative relevance of an article with regard to the content of a question, a ranking approach was adopted. First, the n-gram set of the question is obtained by applying steps 1-6, using the question content instead of article. Then, for each article in the Civil Code, a relevance score is calculated using the following formula:

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

where q_ng_set is the set of n-grams for the question, art_ng_set is the set of n-grams for the article in the stored model, I_q is the relative significance of the question n-gram set size and I_{art} is the relative significance of the article n-gram set size. $idf(t)$ is the Inverse Document Frequency for the term t over the articles collection

$$idf(t) = log\frac{N}{df_t} \tag{2}$$

where N is the total number of articles and df_t is the number of articles in which t appears.

The formula (1) is a variation of the traditional *TF-IDF* scoring method, disregarding term frequency and giving different weights for the two types of document being evaluated: articles and questions, according to their size. I_q and I_{art} are parameters to be adjusted according to the corpus characteristics. This formula was developed after initial experiments with a TF-IDF based classifier showed poor results for this task and further observation showed that TF did not contribute for article relevance in many cases. With TF removed, document size becomes a more relevant feature and must be considered in the scoring.

From this point, the articles are sorted by descending score and the 10 best are selected for filtering. The filtering step consists in fetching the best scoring article and verifying if it exceeds a parameter threshold $confidence\_thresh$. If it does, all the articles in the list that are referred by the first and exceed a parameter threshold $reference\_thresh$ are also fetched. The fetched articles compose the final list of relevant articles to the input question. All R_2NC parameters I_q, I_{art}, $confidence\_thresh$, $reference\_thresh$ and also k, the maximum n-gram size, were adjusted empirically on the training data.

4.2 Textual Entailment

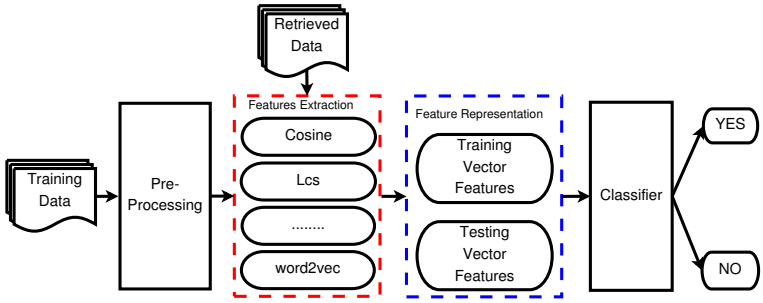


Fig. 4. The process of legal textual entailment recognition

Textual Entailment (TE) in law domain can be represented in form of binary classification [14,5,4]. Given a civil law question and a relevant article, TE relation is assigned by YES label if the article contains information for answering the question; otherwise is NO. More precisely, we aim to solve TE by using machine learning in form of ensemble methods suggested in [6].

TE process is represented in Fig. 4, in which training data is pre-processed by sentence and word segmentation, and removing stop words; next, features are extracted from both training and retrieved data (from the RA step); subsequently, the training and retrieved data are represented in vector space model,

in which training vectors will be used to train a classification model; finally, the model judges entailment relation on retrieved data.

To train the model, a straightforward method is to extract features which represent relevant characteristics of the data. A data observation was conducted to capture the characteristics. The observation is illustrated in Tab. 1.

Table 1. Data statistic in phase two

	# pairs	**# sentences**	**# tokens**	**% uni-gram word overlapping**
Training Set	267	273	36.562	58.80

The observation suggested that word overlapping and word similarity between a question and an article can be useful to capture TE relation. Based on the observation, we proposed the use of 15 lexical features in the form of two groups: distance and statistical features due to its common characteristics. The features are shown in Tab. 2. Note that features in Section 4.1 can be also used for this task; however, due to time constraint, these features will be investigated in the future.

Table 2. The feature groups; Avg is average; Q is a question, S is a sentence

	Feature	**Description**
Distance	Manhattan	Manhattan distance from two text fragments
	Euclidean	Euclidean distance from two text fragments
	Cosine similarity	Cosine similarity distance
	Matching coefficient	Matching coefficient of two text fragments
	Dice coefficient	Dice coefficient of two text fragments
	Jaccard	Jaccard distance of two text fragments
	Jaro	Jaro distance of two text fragments
	Damerau-Levenshtein	Damerau Levenshtein distance of two text fragments
	Levenshtein	Levenshtein distance of two text fragments
Statistical	Lcs	The longest common sub string of two text fragments
	Average of TF-IDF	Term frequency-inverse document frequency
	Avg-TF of Q and S	Avg-TF of words in a Q appearing in a S
	Avg-TF of S and Q	Avg-TF of words in a S appearing in a S
	Word overlapping	# word overlapping in a Q appearing in a article
	Average of Word2Vec	Average of word2vec similarity

After extracting features, pairs in the training dataset were represented by feature vectors with two pre-defined classes: entailment (YES) or no-entailment (NO). These vectors were used to find hypothesis for the classification model. The model would be used to judge the entailment relation on testing dataset.

5 Experiments and Results

5.1 Experimental Setup

The dataset was obtained from the published data for the COLIEE shared task [5], consisting in a text file with the Japanese Civil Code and a set of XML files with

[5] webdocs.cs.ualberta.ca/m̃iyoung2/COLIEE2015/

training and testing data for phases one to three. The training set for the three tasks contains 267 pairs (question, relevant articles). Experiments where divided in phases one and two only, dealing with Information Retrieval and Textual Entailment methods respectively. Each experiment comprised: i) data analysis, ii) model and parameter adjustments and iii) test runs.

For the IR task, data analysis suggested that the problem of restricted linguistic information could be overcome by including external corpora into the n-gram generation process. Thus, attempts were made to expand the R_2NC n-gram model with three different corpora, as follows:

- *News*: One billion word collection of news articles (in English). Text tokenized and cleaned from markups [15].
- *CA—LA_Law*: Collection of Civil Codes from U.S. states of California [6] and Louisiana [7]. Contains 3420 cleaned and tokenized articles, with about 1.7 million words in total.
- *JPN_Law*: Collection of all Civil law articles of Japan's constitution [8]. Contains 642 cleaned and tokenized articles, with about 13.5 million words.

Terms in corpora were clustered by means of n-gram cosine similarity using *Word2Vec* [16]. For a given article content, the n-gram cluster with the highest cosine similarity to the text was included into its n-gram set. Tokenization and lemmatization were done using *NLTK*[9] (v. 3.0.2) and POS-tagging was done using *Stanford Tagger*[10] (v. 3.5.2). Experiment results are shown in Table 3.

Parameters were adjusted for each test run and the best results are reported. The final parameter values used in the competition are $k = 3$, $I_q = 0.965$, $I_{art} = 0.035$, $confidence\_thresh = 0.32$ and $reference\_thresh = 0.2$.

For the TE task, AdaBoost [17] was used to train the model, where: *classifier* was the standard *DecisionStump*, with 10 iterations, $seed = 1$, no *re-sampling* and $weightthreshold = 100$.

5.2 Baselines

As for the second edition of COLIEE, there is still no definite baseline for the competition dataset. However, common baseline practices and related works could be used for evaluating performance on each task. For phase one, a relationship can be drawn between R_2NC and TPP [3], the latter achieving a recall of 0.52 for the top 3 retrieved statutes and 0.91 for the top 10.

Support Vector Machine (SVM) [18] in the LibSVM [11] implementation, integrated in Weka[12] was used as the baseline for phase 2. The parameters of SVM are $C = 1$, $\gamma = 0$, *kernel Type = radial basis function*. Another baseline was using SVMs as weak learners for AdaBoost, instead of the standard DecisionStump.

[6] leginfo.legislature.ca.gov
[7] legis.la.gov
[8] www.japaneselawtranslation.go.jp
[9] www.nltk.org
[10] nlp.stanford.edu/software/tagger.shtml
[11] www.csie.ntu.edu.tw/c̃jlin/libsvm/
[12] weka.wikispaces.com

5.3 Evaluation Method

Given the limited training data available, leave-one-out validation was used to evaluate the performance of the model in both tasks on the training dataset with three measures: precision (P), recall (R) and F-measure (F) as in Eq. (3), (4) and (5). In phase two, accuracy (A) measurement is also used as in Eq. (6).

$$P = \frac{Cr}{Rt} \quad (3) \qquad R = \frac{Cr}{Rl} \quad (4) \qquad F = \frac{2(P * R)}{P + R} \quad (5) \qquad A = \frac{Cq}{Q} \quad (6)$$

where Cr counts the correctly retrieved articles for all queries, Rt counts the retrieved articles for all queries, Rl counts the relevant articles for all queries, Cq counts the queries correctly confirmed as true or false and Q counts all the queries.

5.4 Results

Relevance analysis results were as follows:

Table 3. Experiment results for phase one (Information Retrieval) with R_2NC. Model and parameter adjustments were made for each test run. The best results are presented.

Ext. corpus	Precision	Recall	F-measure
None	0.568	0.516	**0.54**
News	0.461	0.485	0.472
CA—LA_Law	0.476	0.498	0.486
JPN_Law	0.541	0.52	0.53

Additionally, recall was 0.643 and 0.77 for the top 3 and top 10 retrieved articles respectively. This indicates that R_2NC is expected to be competitive with state-of-the-art approaches to relevance analysis in legal documents. However, the proposed method is much simpler when compared to *TPP* [3] and operates with considerably less training data: 266 documents for R_2NC against 1518 documents for *TPP*. R_2NC design also makes it difficult for the model to be overtrained beyond the parameter adjustment, since no training data is counted more than one time and the method is single-shot, as opposed to convergence-based. The test with no external corpus was repeated with traditional TF-IDF scoring, yielding 0.51 F-score. An important observation is that the results got worse when expanding the n-gram model with external data. This could indicate that the external corpora contains a fair amount of noise, or that the questions are highly corpus oriented, and the relevant information is at a abstraction level not reachable by morphological analysis. The drop in F-measure as the external data becomes semantically farther from the Civil Code corpus and the fact that using Japan Law corpus improved the recall support the latter point of view.

Results of TE in Tab. 4 indicate that our method significantly outperforms the baselines 0.084 (8.4%) and 0.113 (11.3%) of F-measure, respectively. More importantly, the precision and accuracy of our method also achieves high improvements in comparison the baselines. This concludes that our method is expected to be efficient for solving TE in legal domain.

Table 4. The performance our method vs. SVMs, where our method uses *DecisionStump* and SVMs uses *Radial basis function* kernel; with the strongest features

	Precision	**Recall**	**F-measure**	**Accuracy (%)**
Our method	0.621	0.614	0.597	**61.42**
SVMs	0.537	0.543	0.513	*54.30*
Our method	0.621	0.614	0.597	**61.42**
AdaBoost with SVMs	0.485	0.491	0.484	*49.06*

An alternative interesting point is that Word2Vec similarity contributes to improve the performance of our model. In reality, as stated in Section 3, legal documents usually require an inference to understand and answer a question; therefore, semantic similarity from Word2Vec can help to improve the performance. The results also show the efficiency of lexical features.

The performance of TE model, however, is not comparable with the same task in common data i.e., news articles [5,6] due to the characteristics of law dataset in LQA stated in Section 3. The performance is also not improved so much even when many features in both phase one and two were combined. Moreover, the observations from Relevance Analysis issues also apply here. This suggests that sophisticated features e.g., semantic inference or semantic rules should be considered.

5.5 Feature Evaluation

Further evaluation of feature impact on TE model was also conducted. In the evaluation, each feature was removed and the remain ones were kept to find out the most effective features. The most effective features are shown in Tab. 5.

Table 5. Top 4 influential features, italic is statistical features

Features	**Influential value**	**Features**	**Influential value**
Euclidean	0.005	*Lcs*	0.0001
Damerau-Levenshtein	0.154	Average of Word2Vec	0.024

Results in Tab. 5 show that all effective features contribute to the method. Note that both *Damerau-Levenshtein* and *Euclidean* are distance features whereas *the longest common substring* is statistical feature. The results support that in legal texts, there is not much word overlapping between a question and relevant articles. An interesting aspect is that Word2Vec similarity has a big positive impact to the model. This supports the conclusion on similarity stated in Section 5.4.

5.6 Error Analysis and Discussion

Investigation of the ranked list obtained with R_2NC in phase one (see Section 4.1) revealed that in most cases, relevant articles ranked lower than second had keywords not present in the corpus nor in the trained models. This reinforces the view that the questions are highly directed, albeit in a conceptual

level. Relevant articles that ranked lower than 15th (approx. 20%) were found to require a relatively high level of abstraction to obtain an interpretation that could link to the corresponding question. Table 6 shows an example of complex relevance relationship.

Table 6. Example of low ranking, high relevance article and corresponding question

ID	Article	Question	Ranked in
H18-2-1	Article 697(1)A person who commences the management of a business for another person without being obligated to do so (hereinafter in this Chapter referred to as "Manager") must manage that business (hereinafter referred to as "Management of Business") in accordance with the nature of the business, using the method that best conforms to the interests of that another person (the principal).(2)The Manager must engage in Management of Business in accordance with the intentions of the principal if the Manager knows, or is able to conjecture that intention.	In cases where a person plans to prevent crime in their own house by fixing the fence of a neighboring house, that person is found as having intent towards the other person.	424th

Table 7. Examples of entailment judgment; P is predicted and A is annotated

ID	Article	Question	P	A
H18-2-4	(Managers' Claims for Reimbursement of Costs)Article 702(1)If a Manager has incurred useful expenses for a principal, the Manager may claim reimbursement of those costs from the principal.(2)The provisions of Paragraph 2 of Article 650 shall apply mutatis mutandis to cases where a Manager has incurred useful obligations on behalf of the principal.(3)If a Manager has engaged in the Management of Business against the intention of the principal, the provisions of the preceding two paragraphs shall apply mutatis mutandis, solely to the extent the principal is actually enriched.	In cases where a person repairs the fence of a neighboring house after it collapsed due to a typhoon, but the neighbor had intended to replace the fence with a concrete-block wall in the near future, if a separate typhoon causes the repaired sections to collapse the following week, reimbursement of repair fees can no longer be demanded.	YES	YES
H18-26-1	(Renunciation of Shares and Death of Co-owners)Article 255 If one of co-owners renounces his/her share or dies without an heir, his/her share shall vest in other co-owners.	In cases where person A and person B co-own building X at a ratio of 1:1, if person A dies and had no heirs or persons with special connection, ownership of building X belongs to person B.	NO	YES

Table 7 shows that our method gives correct decisions e.g., ID H18-2-4. In this example, there several common words leading our model can correctly judge the TE relation e.g., *reimbursement*. In addition, several words can be inferred from the questions by using Word2Vec similarity e.g., *person* $\sim$ *manager*, *fees* $\sim$ *costs or expenses*. This supports our observation that TE can be solved by using lexical features and word similarity. An alternative interesting point is that even H18-2-4 contains a few common words and needs a reference to understand and answer the question; however, our method can still predict the TE relation correctly. This indicates the efficiency of our model as well as the features, especially word similarity feature.

On the other hand, the pair H18-26-1 exemplifies a case in which our model predicted NO while TE relation was annotated YES even when the question and

answer share some common words. This shows the limitation of our feature set in cases where the question and answer are short. In this case, after removing stop words, a few remaining words may not be enough to capture the TE relation. Moreover, lack of important words e.g., *building*, *connection* or *belong* reveals a big challenge for our method to decide the TE relation. This suggests that a keyword enriching mechanism such as term expansion used in phase one could improve the results.

6 Conclusion

This paper explores the challenging issue of building a QA system in the legal domain. We propose a model including three stages: legal text retrieval, legal textual entailment and legal text answering. In the first stage, a mixed size n-gram model built from morphological analysis is used to find out relevant articles corresponding to a legal question; next, pairs of questions and retrieved articles are judged by a machine learning algorithm trained on lexical features, to decide whether the questions can be answered positively or negatively by the retrieved articles; and finally, correct answers would be provided for users in the final stage. The contributions of this work in IR and TE task are: 1) a simple, yet effective language model for law corpora coupled with a Relevance Analysis method (R_2NC) capable of exploiting such model; 2) a set of RTE features, including Word2Vec similarity for Machine Learning algorithms. With a F-measure of 0.54 for the first retrieved articles and recall of 0.64 for the top 3, R_2NC appears as competitive when compared to state-of-the-art similar work, in spite of being much more simple and applicable with less training data. By combining RTE features and Word2Vec similarity, our method for LQA also significantly outperforms the baselines 0.084 (8.4%) and 0.113 (11.3%) of F-measure.

For future directions, information on a higher abstraction level, e.g., syntactic mappings, could be used to improve the language model for the IR task. In the TE task, since a sentence in a legal article is usually long, a sophisticated method of sentence partition e.g., requisite and effectuation should be considered. In feature extraction, features in IR should be combined with lexical features in TE and investigated to improve the quality of the judgment. Moreover, capturing contradictions in the TE relation by current statistical features is a big challenge. To solve this issue, semantic rules should be defined and incorporated into the feature extraction. Finally, we would like to investigate and apply sentence similarity calculation by Sent2Vec to improve the performance of the TE.

Acknowledgements

This work is supported partly by the grant of NII Research Cooperation and AIST's Research grant.

References

1. Robert C. Berring: "The heart of legal information: The crumbling infrastructure of legal research". *Legal information and the development of American law*. St. Paul, MN: Thomson/West, 2008.

2. Rinke Hoekstra, Joost Breuker, Marcello Di Bello and Alexander Boer: "The LKIF Core ontology of basic legal concepts". Proc. of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007), 2007.
3. Yi-Hung Liu, Yen-Liang Chen and Wu-Liang Ho: "Predicting associated statutes for legal problems". *Information Processing & Management* 51.1: 194-211, 2015.
4. Diana Inkpen, Darren Kipp and Vivi Nastase: "Machine Learning Experiments for Textual Entailment. *Proceedings of the Second Challenge Workshop Recognising Textual Entailment*: 17-20, 2006.
5. Julio Javier Castillo: "An approach to Recognizing Textual Entailment and TE Search Task using SVM. *Procesamiento del Lenguaje Natural*, 44: 139-145, 2010.
6. Minh-Tien Nguyen, Quang-Thuy Ha, Thi-Dung Nguyen, Tri-Thanh Nguyen and Le-Minh Nguyen: "Recognizing Textual Entailment in Vietnamese Text: An Experiment Study." KSE, 2015 (accepted).
7. Quang Nhat Minh Pham, Le Minh Nguyen and Akira Shimazu: "Using Machine Translation for Recognizing Textual Entailment in Vietnamese Language." RIVF: 1-6, 2012.
8. Danilo Giampiccolo, Bernardo Magnini, Ido Dagan and Bill Dolan: "The third PASCAL recognising textual entailment challenge". *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*. Association for Computational Linguistics: 1-9, 2007.
9. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen and Akira Shimazu: "Answering Legal Questions by Mining Reference Information". New Frontiers in Artificial Intelligence. Springer International Publishing: 214-229, 2014.
10. Mi-Young Kim, Ying Xu, Randy Goebel and Ken Satoh: "Answering Yes/No Questions in Legal Bar Exams". New Frontiers in Artificial Intelligence. Springer International Publishing: 199-213, 2014.
11. Bach Xuan Ngo, Minh Le Nguyen and Akira Shimazu: "RRE Task: The Task of Recognition of Requisite Part and Effectuation Part in Law Sentences." J. IJCPOL 23(2): 109-130, 2010.
12. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen and Akira Shimazu: "Reference Resolution in Legal Texts." In Proc. of ICAIL: 101-110, 2013.
13. Ido Dagan, Bill Dolan, Bernardo Magnini and Dan Roth: "Recognizing textual entailment: Rational, evaluation and approaches - Erratum". *Natural Language Engineering* 16(1): 105-105, 2010.
14. Rui Wang: *"Intrinsic and Extrinsic Approaches to Recognizing Textual Entailment.* Saarland University, ISBN 978-3-933218-32-2: 1-219, 2011.
15. Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn and Tony Robinson: "One billion word benchmark for measuring progress in statistical language modeling", arXiv preprint arXiv:1312.3005, 2013.
16. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean: "Distributed Representations of Words and Phrases and their Compositionality". In Proceedings of NIPS, 2013.
17. Yoav Freund and Robert E. Schapire: "A decision-theoretic generalization of online learning and an application to boosting". Journal of computer and system sciences 55.1: 119-139, 1997.
18. Corinna Cortes and Vladimir Vapnik: "Support-Vector Networks". *Machine Learning* 20(3): 273-297, 1995.

Keyword and Snippet Based Yes/No Question Answering System for COLIEE 2015

Yoshinobu Kano[1,*]

[1] Faculty of Informatics, Shizuoka University, Japan
kano@inf.shizuoka.ac.jp

Abstract. A central issue of Yes/No question answering is usage of knowledge source given a question. The most basic, but fundamental information is keyword extraction, keyword distribution, and document structure in the knowledge source. We employed our question answering system that was originally developed for multiple-choice style question answering of another domain. Our system uses keyword distribution and document snippet structure, in order to score relevant snippets and penalize irrelevant snippets with regard to the given question. As our first challenge to the COLIEE task, our system performance is comparable to other previous systems for the legal bar exam without tuning to the legal domain. The result shows the importance of the points described above, while we found many difficult issues specific to this legal domain.

Keywords: COLIEE, Question Answering, Legal Bar Exam, Legal Information Extraction.

1 Introduction

Question-answering techniques could include logic, reasoning, syntactic and semantic analysis. However, almost all of techniques rely on keyword extraction. Given a yes/no answering proposition, we determine yes or no using knowledge sources. The first step of such a determination process is always keyword extraction. Other deeper analyses are based on the keyword extraction results. This implies that quality of keyword extraction from knowledge sources play a critical role.

We have been trying to answer Yes/No questions of the Center Test (Japanese National Center Test questions for University Admissions). Center Test is a common examination for Japanese university applicants who are asked to answer multiple-choice problems. Our system [1][2] achieved the best result [3] in the History subjects of Mock Exam Challenge held by the Todai Robot project [4]. As our first challenge for this COLIEE Bar Exam legal entailment challenge, we employed the same method of our previous Center Exam solver system rather than to try another method, in order to observe difference between these two exams.

In this paper, we assume that keyword distribution is sufficient to perform the legal bar exam question answering tasks to a certain extent of performance in addition to the precise keyword extraction.

2 Previous Works

The Todai-Robot project aims to solve university entrance examinations automatically as a challenging task of artificial intelligence. The target examinations include the Center Test. The Center Test problems are also used in the NTCIR RITE2, RITEVal [5], and QALab [6] subtasks. Textbooks of corresponding subjects are provided as knowledge sources. These textbooks have document structures of paragraph, subsection, and section annotated.

Most frequent types of questions in the History subjects are “select the correct choice” type, “select the wrong choice” type, and “combination” type. In the History subjects, domain, location and age could be different in the choices of the same question. Thus, it is not clear for which domain the system should search for the knowledge source. In addition, the expressions used in the questions and those used in the knowledge source are usually different and may be described in several sentences. Furthermore, in the case of wrong choices, there should be no corresponding part in knowledge source. These observations demonstrate that this task is difficult for machines to solve.

The multiple-choice style problems can be developed to a set of Yes/No questions. Therefore, our Center Test solver could be directly applied to this COLIEE task.

3 System

3.1 Scoring

We assume that the answer of a question is described in only one place (snippet) in the knowledge source with affirmative expressions. We designed our system in a domain independent way, which uses an unsupervised method for QA.

Our QA system originally outputs a confidence score w.r.t. the input in the case of “select correct/wrong choice” type of questions. In other words, let x be the given choice, our system output S(x) as the confidence of x. Roughly speaking, our system performs (1) keyword extraction from the input, (2) keyword weighting of the input, (3) textbook search and scoring.

(1) Keyword Extraction

Because Japanese texts are concatenation of characters not having spaces between words, we apply a morphological analyzer Kuromoji [7], which is based on Mecab [8], to the input. We augmented the dictionary of Kuromoji with the headings of the entries

of Japanese Wikipedia. We extract all strings that match with the Wikipedia entries by longest match in the input as the keywords.

(2) Keyword Weighting

Let c_i be the frequency of i-th distinct keyword in the input, then the weight of i-th keyword is

$$w_i = 1/(c_i z)$$

In this equation, $z = \sum_i 1/c_i$ is a normalizing constant, where i is defined over the distinct keywords in the input. The frequency c_i was counted over the textbook data.

(3) Knowledge Source Searching and Scoring

We divided the knowledge source data into snippets. We tried three types of snippets as described later. We search for the snippet that has the highest score w.r.t the input keyword set ***K***, which consists of the keywords in the input.

Let ***R*** be the word set extracted from a snippet, then the score of ***R*** is

$$s_{\boldsymbol{R}} = \sum_{l \in \boldsymbol{R} \cap \boldsymbol{K}} w_l - \sum_{m \in \boldsymbol{K} - \boldsymbol{R}} w_m$$

This expression means that the score of the snippet is the sum of the weights of the input keywords included in the snippet minus that not included in the snippet. If a given choice is correct, keywords in the choice should be densely included in a specific snippet of the textbook; if a given choice is wrong, its keywords should be scattered across snippets. The above equation penalizes such a scattered keyword distribution.

Finally, we regard the maximum $s_{\boldsymbol{R}}$ among all of snippets as the confidence score of the corresponding input.

In Phase one, we extract "*X*条" in a snippet of the largest score. Then we returned the number *X* as our answer.

In Phase two, we answered Yes or No determined by a threshold value given the confidence score above.

We do not consider negations because our system originally assumes textbooks as knowledge source which normally describe events in an affirmative way.

Our proposed method above does not depend on any domain specific information, even on any specific language.

4 Experiments

Experiments were conducted on the COLIEE 2015 Japanese subtask dataset. All of our evaluation results used the RITEVal official evaluation tool. Since our system is unsupervised, we did not use the development set.

Table 1 shows results of our proposed method in the training set (evaluation result of the formal run is not provided before the submission deadline). As described in the

previous section, we used three types of snippets, section, subsection and paragraph, larger to smaller in this order. Section corresponds to “条”, while paragraph corresponds to “項”. Subsection corresponds to a concatenation of “項”. “項” sometimes refer to another “項” using expressions like “前項” or “第*X*項”. We concatenated such a referred “項” and regarded as a single snippet of the subsection level.

Table 1. Results of the training dataset.

Dataset	**H18**	**H19**	**H20**	**H21**	**H22**	**H23**
Accuracy	61.11	59.52	62.22	62.50	61.70	56.10
F-Score	61.11	42.19	61.54	51.14	61.26	55.43

5 Discussion

Our system was originally designed to solve the multiple-choice problems in the Center Test. Although the COLIEE data set is similar in the style of questions, there are a couple of differences from the Center Test problems.

Firstly, technical terms in the legal bar exam are quite different from the History examination. We dare not to tune the dictionary in order to compare these two datasets as our first challenge. However, as far as we observed, keyword extraction was fine enough for the COLIEE dataset.

Secondly, the technical terms in the COLIEE dataset tend to include logical relations implicitly, while the technical terms in the History exam are mostly names or concepts. This sort of logical relation extraction would be still very difficult to perform in sufficiently high accuracy considering the current NLP technologies.

Thirdly, the document structure of the given knowledge source, the Japanese Civil Law articles, is special. The Civil Law articles have references; logic and conditions are described in a single sentence, or scattered across snippets. We tried to solve the reference issue as described in the previous section, which increased our system performance significantly

Fourthly, our system is originally developed for multiple-choice style questions but not for binary style questions.

Finally, the Civil Law articles include specific negation and acronym expressions which are critical in solving the problems. Our system does not perform any solution to these issues.

Future work would include solutions to the above issues.

6 Conclusion

We employed our Yes/No question answering system which was originally developed for the Japanese Center Test problems. As our first challenge to the legal bar exam problems, we used our system as it is to observe the system behavior in the legal bar exam, while tuned the knowledge source structure to fit with our system. Our system performance was comparable to the previous COLIEE systems. There are still a couple of issues specific to the legal domain such as logical keywords. Future work would include solutions to these issues.

Acknowledgements

This work was partially supported by MEXT Kakenhi and National Institute of Informatics.

References

1. Kano, Y. (2014). Answering Yes-No Questions by Keyword Distribution: KJP System at NTCIR-11 RITEVal Task. *Proceedings of the 12th NTCIR Conference*.
2. Kano, Y. (2014). Solving History Exam by Keyword Distribution: KJP System at NTCIR-11 QALab Task. *Proceedings of the 12th NTCIR Conference*.
3. Kano, Y. (2014). Solving History Problems of the National Center Test for University Admissions (in Japanese). *Proceedings of the Annual Conference of JSAI*
4. http://21robots.org/
5. 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. *Proceedings of the 11th NTCIR*
6. Shibuki, H, Sakamoto, K., Kano, Y., Mitamura, T., Ishioroshi, M., Itakura, K., Wang,D., Mori, T. and Kando,N. (2014). Overview of the NTCIR-11 QA-Lab Task. *Proceedings of the 11th NTCIR*
7. http://www.atilika.org/
8. https://code.google.com/p/mecab/

A Convolutional Neural Network in Legal Question Answering

Mi-Young Kim, Ying Xu, and Randy Goebel

Dept. of Computing Science, University of Alberta, Edmonton, Canada

{ miyoung2,yx2,rgoebel}@ualberta.ca

Abstract. Our legal question answering system combines legal information retrieval and textual entailment, and we propose a legal question answering system that exploits a deep convolutional neural network. We have evaluated our system using the training data from the competition on legal information extraction/entailment (COLIEE). The competition focuses on the legal information processing related to answering yes/no questions from Japanese legal bar exams, and it consists of two phases: legal ad-hoc information retrieval, and textual entailment. Phase 1 requires the identification of Japan civil law articles relevant to a legal bar exam query. For that phase, we have implemented TF-IDF and Ranking SVM. Phase 2 is to answer "Yes" or "No" to previously unseen queries, by comparing the meanings of queries with relevant articles. Our choice of features used for Phase 2 focuses on word embeddings, syntactic similarities and identification of negation/antonym relations. Phase 2 is a textual entailment problem, and we use a convolutional neural network with dropout regularization and Rectified Linear Units. To our knowledge, our study is the first approach adopting deep learning in the textual entailment field. Experimental evaluation demonstrates the effectiveness of the convolutional neural network and dropout regularization. The results show that our deep learning-based method outperforms the SVM-based supervised model.

Keywords: legal question answering, recognizing textual entailment, information retrieval, convolutional neural network

1. Task Description

The deep neural network (DNN) is an emerging technology that has recently demonstrated dramatic success in several areas, including speech feature extraction and recognition. Incorporation of convolution and subsequent pooling into a neural network gives rise to a technique called a Convolutional Neural Network (CNN) [22]. CNN showed good performance in image and speech recognition [18], and many studies have been proposed applying CNN in natural language processing [11,12]. Here we adapt a CNN for legal question answering. Legal question answering requires a number of intermediate steps. For instance, to answer a question such as "Is it true that a special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty." a system must first identify and retrieve relevant

documents, typically legal statutes, and subsequently, identify a most relevant sentence. Finally it must compare the semantic connections between question and the relevant sentences, and determine whether an entailment relation holds.

The Competition on Legal Information Extraction/Entailment (COLIEE) 2015 focuses on two aspects of legal information processing related to answering yes/no questions from legal bar exams: Legal document retrieval (Phase 1), and Yes/No Question answering for legal queries (Phase 2).

Phase 1 is an ad-hoc information retrieval (IR) task. The goal is to retrieve relevant Japan civil law articles that are related to a question in legal bar exams.

We approach this problem with two models based on statistical information. One is the TF-IDF model [1], i.e. term frequency-inverse document frequency. The relevance between a query and a document depends on their intersection word set. The importance of words is measured with a function of term frequency and document frequency as parameters. Our terms are lemmatized words, which mean verbs "attending," "attends," and "attended" are lemmatized as the same form "attend."

Another popular model for text retrieval is a Ranking SVM model [2]. That model is used to re-rank documents that are retrieved by the TF-IDF model. The model's features are lexical words, dependency path bigrams and TF-IDF scores. The intuition is that the supervised model can learn weights or priority of words based on training data in addition to or as an alternative to TF-IDF.

The goal of Phase 2 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. The answer to a question is typically determined by measuring some kind of heuristically computed semantic similarity between question and answer. While there are many possible approaches, we consider that Neural network-based distributional sentence models have achieved successes in many natural language processing tasks such as sentiment analysis [12], paraphrase detection [13], document classification [14], and question answering [11]. As a consequence of this success, it appears natural to approach textual entailment using similar techniques. In this paper, we show that a neural network-based sentence model can be applied to the task of textual entailment. After constructing a set of pre-trained semantic word embeddings using the *word2vec* [20], we train a supervised model to learn a heuristic semantic matching between question and corresponding articles.

In addition to the semantic word embeddings, our system uses features that depend on the syntactic structure, and the presence of negation. We employ a convolutional neural network algorithm with dropout regularization and Rectified Linear Units, and compare its performance with Support vector machines.

2. Phase 1: Legal Information Retrieval

2.1 IR Models

2.1.1 TF-IDF model

Here, we introduce our TF-IDF and Ranking SVM models. Queries and articles are all tokenized and parsed by the Stanford NLP tool. For the IR task, the similarity of a

query and an article is based on the terms within them. Our terms for TF-IDF are lemmatized words.

For TF-IDF, we use the simplified version of Lucene's similarity score of an article to a query as suggested in [15]:

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

The score *tf-idf(Q,A)* is a measure which computes the relevance between a query Q and an article A. First, for every term t in the query A, we compute *tf(t,A),* and *idf(t).* The score *tf(t,A)* is the term frequency of t in the article A, and *idf(t)* is the inverse document frequency of the term t, which is the number of articles that contain t. After some normalization process in the Lucene package, we multiply *tf(t,A)* and *idf(t),* and then we compute the sum of these multiplication scores for all terms t in the query A. This summation result is *tf-idf(Q,A).* The bigger *tf-idf(Q,A)* is, the more relevant between the query Q and the article A. The real version has some normalized parameters in terms of an article's length to alleviate the functions biased towards long documents.

2.1.2 Ranking SVM

The Ranking SVM model was proposed by Joachims [2]. That model ranks the documents based on user's data. Given the feature vector of a training instance, i.e., a retrieved article set given a query, denoted by $\Phi(Q, A_i)$, the model tries to find a ranking that satisfies constraints:

$$\emptyset(Q,A_i) > \emptyset(Q,A_j)$$

where A_i is a relevant article for the query Q, while A_j is less relevant.

We adopt the same model and features suggested in [15]. The three types of features are as follows:

- Lexical words: the lemmatized normal form of surface structure of words in both the retrieved article and the query. In the conversion to the SVM's instance representation, this feature is converted into binary features whose values are one or zero, i.e., depending on if a word exists in the intersection word set or not.
- Dependency pairs: word pairs that are linked by a dependency link. The intuition is that, compared with the bag of words information, syntactic information should improve the capture of salient semantic content. Dependency parse features have been used in many NLP tasks, and improved IR performance [3]. This feature type is also converted into binary values.
- TF-IDF score (Section 2.1.1).

2.2 Experiments

The legal IR task has several sets of queries with the Japan civil law articles as documents (724 articles in total). Here follows one example of the query and a corresponding relevant article.

Question: A person who made a manifestation of intention which was induced by duress emanated from a third party may rescind such manifestation of intention on the basis of duress, only if the other party knew or was negligent of such fact.

Related Article: (Fraud or Duress) Article 96(1)Manifestation of intention which is induced by any fraud or duress may be rescinded.(2)In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.(3)The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.

Before the final test set was released, we received 6 sets of queries for a dry run in COLIEE 2015. The 6 sets of data include 267 queries, and 326 relevant articles (average 1.22 articles per query). We used the corresponding 6-fold leave-one-out cross validation evaluation. The metrics for measuring our IR model performance is Mean Average Precision (MAP):

$$MAP(Q) = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{m}\sum_{k\in(1,m)} precision(R_k)$$

where Q is the set of queries, and m is the number of retrieved articles. R_k is the set of ranked retrieval results from the top until the k-th article. In the following experiments, we set m as 5 for all queries, corresponding to the column *MAP@5* in Table 1.

Table 1 shows the results of experiments with our two IR models on the legal IR task. The ensemble SVM-Ranking model is slightly better than the TF-IDF model. Table 2 shows the results of our SVM-ranking model on the final test set.

Table 1. IR results on dry run data with different models.

Id	Models	MAP@5
1	TF-IDF with lemma	0.294
2	SVM-ranking	0.302

Table 2. IR results on test data using the SVM-ranking model

Participant ID	Performance on Phase 1	
UA (University of Alberta)	* The number of submitted articles: * The number of correctly submitted articles:	
	Precision	0.6329
	Recall	0.4902
	F-measure	**0.5525**

3. Phase 2: Answering 'Yes'/'No' Questions Using a Convolutional Neural Network

Our system uses syntactic information in addition to word embedding to predict textual entailment. We exploit syntactic similarity features, negation and antonym in Kim et al.[15]. We describe our system in the next subsections in detail.

3.1 Our System

3.1.1 Model description

The problem of answering a legal yes/no question can be viewed as a binary classification problem. Assume a set of questions Q, where each question $q_i \in Q$ is associated with a list of corresponding article sentences $\{a_{i1}, a_{i2}, ..., a_{im}\}$, where $y_i = 1$ if the answer is 'yes' and $y_i = 0$ otherwise. We choose the most relevant sentence a_{ij} using the algorithm of Kim et al. [15], and we simply treat each data point as a triple (q_i, a_{ij}, y_i). Therefore, our task is to learn a classifier over these triples so that it can predict the answers of any additional question-article pairs.

Our solution to this problem assumes that correct answers have high semantic similarity to questions. We model questions and answers as vectors using word embedding and linguistic information, and evaluate the relatedness of each question-article pair in a shared vector space. Following Yu et al. [11], given the vector representations of a question $\boldsymbol{q}$ and a most relevant article sentence $\boldsymbol{a}$, the probability of the answer being correct is

$$p(y=1 \mid \boldsymbol{q}, \boldsymbol{a}) = rectifier(\boldsymbol{q}^T\boldsymbol{M}\boldsymbol{a} + \boldsymbol{b}),$$

Where the bias term b and the transformation matrix $\boldsymbol{M}$ in $R^{d \times d}$ are model parameters. This formation can be understood that we generate a question through the transformation $\boldsymbol{q'}=\boldsymbol{M}\boldsymbol{a}$, and then measure the similarity of the generated question $\boldsymbol{q'}$

and the given question $\boldsymbol{q}$ by their dot product. The rectifier function is used as an activation function.

Convolutional Neural Network(CNN) is a biologically-inspired variant of Multi-Layer Perceptron. CNN has two techniques: (1) restricting the network architecture through the use of local connections known as receptive fields. (2) Constraining the choice of synaptic weights through the use of weight-sharing. Most of CNNs include max-pooling layer which reduces and integrates the neighboring neurons' outputs. CNNs exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers.

CNN-based models have been proved to be effective in applications such as twitter sentiment prediction [12] and semantic parsing [16]. Figure 1 illustrates the architecture of the CNN-based sentence model in one dimension. We use word embedding and linguistic features with one convolutional layer and one pooling layer. A word embedding is a parameterized function mapping words in some language to high-dimensional vectors. We use the Bag-of-words model of [11] for word embedding.

The bag-of-words model is like that: Given word embeddings, the bag-words model

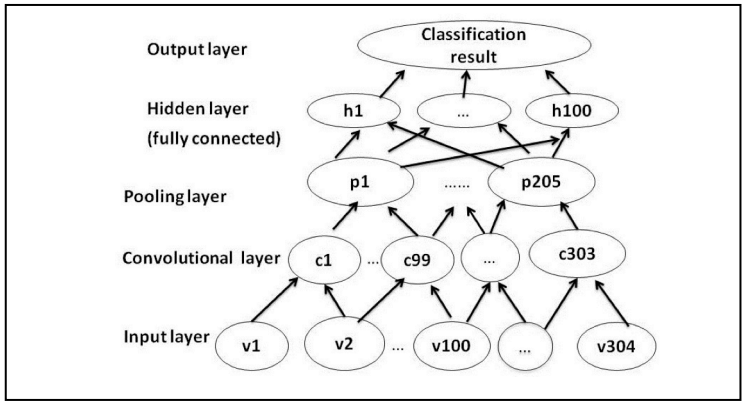


Figure 1. Our CNN architecture

generates the vector representation of a sentence by summing over the embeddings of all words in the sentence after removing stopwords. The vector is then normalized by the length of the sentence.

$$s = \frac{1}{n}\sum_{i=1}^{n} s_i$$

In the formula, s and s_i are d-dimensional vectors. s_i corresponds to the vector of the *i*-th word in the sentence, and *n* is the number of words in the sentence. We used *word2vec* [20] word embedding technique (d=50, which is typically used), and used training data of COLIEE 2014 for training *word2vec*. *Word2vec* [20] used a neural network consisting of input layer, projection layer, and output layer and they removed a hidden layer to improve learning speed.

In addition to word embeddings, the types of features we use are as follows:

a) Word Lemma
b) Tree structure features Considering only roots.

Feature 1 : $w_{root}(condition_{query\_n})$
Feature 2: $w_{root}(condition_{article\_n})$
Feature 3: $w_{root}(conclusion_{query\_n})$
Feature 4: $w_{root}(conclusion_{query\_n})$
Feature 5: neg_level($condition_{query\_n}$)
Feature 6: neg_level($condition_{article\_n}$)
Feature 7: neg_level($conclusion_{query\_n}$)
Feature 8: neg_level($conclusion_{query\_n}$)

In Figure 1, the input layer consists of the following word embedding vectors and binary values:

1) v1, v3,..., v99 (odd index of nodes between v1 and v99): word embedding vector of the query sentence
2) v2, v4, ..., v100 (even index of nodes between v2 and v100): word embedding vector of the relevant article sentence
3) v101, v103, ..., v199 (odd index of nodes between v101 and v199): word embedding vector of the Feature 1
4) v102, v104, ..., v200 (even index of nodes between v102 and v200): word embedding vector of the Feature 2
5) v201, v203, ..., v299 (odd index of nodes between v201 and v299): word embedding vector of the Feature 3
6) v202, v204, ..., v300 (even index of nodes between v202 and v300): word embedding vector of the Feature 4
7) v301-v304 : binary values of the Features 5-8

In the features above, $article_n$ is the most relevant article of the query $query_n$. First we detect condition part and conclusion part in the question and corresponding article, and also compute negation value (neg_level()) of each part according to Kim et al. [15]. $w_{root(s)}$ means the root word in the syntactic tree of the sentence s. Features 1-4 consider both lexical and syntactic information, and Features 5-8 incorporate negation and antonym information. We use some morphological and syntactic analysis to extract lemma and dependency information. Details of the morphological and syntactic analyzer are given in Section 3.2.

As shown in Figure 1, in the input layer, the nodes of even indices between v1 and v100 (such as v2, v4, v6.... and v100) indicate the word embedding vector for a relevant article sentence, and the nodes of odd indices between v1 and v100 (such as v1, v3, ... and v99) show the word embedding vector for a query sentence. The nodes between v101 and v304 are linguistic features. Because the adjacent two nodes indicate the same feature type (e.g., v1 and v2 indicate the first index value of each word embedding vector), we make a convolutional layer constructed from the adjusting two input nodes.

The convolutional vector $\mathbf{t}$ in R^2 (the 2-dimensional real number space) projects adjacent two nodes into a feature value c_i, computed as follows:

$$C_i = rectifier(\boldsymbol{t} * v_{i:i+1} + b),$$

where *rectifier(x) = max(0,x)*. We explain the rectifier function in the subsection 3.1.2. In the pooling layer, we just do summation of the 99 c_i values. Because 99 c_i values are the result of comparison between two embedding vectors between a query and a relevant article sentence. The p_i values in the pooling layer as follows.

$$p_i = \sum_{k=i}^{98+i} c_i$$

The training is done with multithreaded mini-batch gradient descent.

3.1.2 Dropout regularization and Rectified Linear Units

When a neural network is trained on a small training set, it typically performs poorly on test data. This "overfitting" is greatly reduced by randomly omitting some of the feature detectors on each training case. It is called 'dropout'. The dropout prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Random dropout gives big improvements on many benchmark tasks and sets new records for speech and object recognition [21]. We found that the dropout rate needs to be between 0.6 and 0.7 for a hidden layer and 0.1 for an input layer in order to make it effective in achieving low errors.
We also employ the Rectified Linear Unit for CNN. Neural networks with rectified linear unit (ReLU) non-linearities have been highly successful for computer vision tasks and have been shown to be faster to train than standard sigmoid units [17].
ReLU is a neuron which uses a rectifier instead of a hyperboolic tangent or logistic function as an activation function. *Rectifier f(x)=max(0,x)* allows a network to easily obtain sparse representations.

3.1.3 Supervised learning with SVM

We have compared our method with SVM, as a kind of supervised learning model. Using the SVM tool included in the Weka [4] software, we performed cross-validation for the 179 questions using word embedding vector and linguistic features in Section 3.1.1. 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 reduced memory requirements than non-linear kernels, because of the compact representation of the decision function.

3.2 Experimental setup for Phase 2

Table 3. Experimental results on dry run data for Phase 2

Our method	Accu(%)
Baseline	55.87
Cross-validation with Supervised learning (SVM) [15]	59.43
Rule-based model + K-means clustering [15]	61.96
Cross-validation with Supervised learning (SVM) using our features	60.12
Convolutional Neural Network with Word Embedding + Linguistic features + Dropout	63.87
Convolutional Neural Network with Word Embedding +Linguistic features	62.65
Convolutional Neural Network with only linguistic features	56.30

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 (55.87% yes, and 44.13% no) for a dry run of COLIEE 2014 dataset, so we consider the baseline for true/false evaluation is the accuracy when returning always "yes," which is 55.87%. Our data for dry run has 179 questions.

The original examinations are provided in Japanese and English, and our initial implementation used a Korean translation, provided by the Excite translation tool (http://excite.translation.jp/world/). The reason that we chose Korean is that we have a team member whose native language is Korean, and the characteristics of Korean and Japanese language are similar. In addition, the translation quality between two languages ensures relatively stable performance. 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 [5].

3.3 Experimental results

To compare our performance with Kim et al. [15], we measured our system's performance on the dry run data of COLIEE 2014. Table 3 shows the experimental results. An SVM-based model showed accuracy of 60.12%, and a convolutional neural network with pre-trained semantic word embeddings and dropout showed best performance of 63.87% with the setting of input layer dropout rate of 0.1, hidden layer dropout rate of 0.6, and 100 hidden layer nodes. When we did not use the dropout regularization, the accuracy was lower by 1.22%. Without dropout and word embedding, the accuracy was 56.30% which showed no big difference with the baseline accuracy. We also compare our performance with the Kim [15]'s performance using the same dataset. In [15], they used their linguistic features for

SVM learning, and also proposed a model combining rule-based method and k-means clustering. Our CNN performance outperformed both of their SVM model and combined model.

4 Related work

Only very recently have researchers started to apply deep learning to question answering [11,16,19]. Relevant work includes Yih et al. [16] who constructed models for single-relation question answering with a knowledge base of triples. In the same direction, Bordes et al. [19] used a type of siamese network for learning to project question and answer pairs into a joint space. Finally, Yu et al. [11] selected answer sentence, which includes the answer of a question. They modelled semantic composition with a recursive neural network. However these tasks differ from the work presented here in that our purpose is not to make a choice of answer selection in a document, but to answer 'yes' or 'no'.

There was a textual entailment method from W. Bdour et al. [6] which provided the basis for 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 also 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. Nielsen et al. [7] extracted features from dependency paths, and combined them with word-alignment features in a mixture of an expert-based classifier. Zanzotto et al. [8] proposed a syntactic cross-pair similarity measure for RTE. Harmeling [9] took a similar classification-based approach with transformation sequence features. Marsi et al. [10] described a system using dependency-based paraphrasing techniques. All these previous systems uniformly conclude that syntactic information is helpful in RTE: we also use syntactic information. As further research, we will try unsupervised pre-training in a CNN to solve the problem of small training dataset. We are also considering adopting more convolutional layers and pooling layers in the CNN architecture and investigating the effect of more layers in the textual entailment problem.

5 Conclusion

We have described our implementation for the Competition on Legal Information Extraction/Entailment (COLIEE) Task.

For Phase 1, legal information retrieval, we implemented a Ranking-SVM model for the legal information retrieval task. By incorporating features such as lexical words, dependency links, and TF-IDF score, our model shows better mean average precision than TF-IDF.

For Phase 2, we have proposed a method to answer yes/no questions from legal bar exams related to civil law. We used a convolutional neural network model using dropout regularization Rectified Linear Units with pre-trained semantic word embeddings. We also extract deep linguistic features with lexical, syntactic information based on morphological analysis and dependency trees. We show the improved performance over previous systems, using a convolutional neural network.

Acknowledgements

This research was supported by the Alberta Innovates Centre for Machine Learning (AICML) and the Natural Sciences and Engineering Research Council (NSERC).

References

1. K. Sparck Jones, A statistical interpretation of term specicity and its application in retrieval. In: Willett, P. (ed.) Document Retrieval Systems, pp. 132-142. Taylor Graham Publishing, London, UK, UK, 1988
2. T. Joachims, Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 133-142. KDD '02, ACM, New York, NY, USA, 2002
3. K.T. Maxwell, J. Oberlander, W.B. Croft. Feature-based selection of dependency paths in ad hoc information retrieval. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 507-516. Association for Computational Linguistics, Sofia, Bulgaria, August 2013
4. 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
5. 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
6. 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
7. R. D. Nielsen, W. Ward, and J. H. Martin. Toward dependency path based entailment. In Proceedings of the second PASCAL Challenges Workshop on RTE, 2006
8. F. M. Zanzotto, A. Moschitti, M. Pennacchiotti, and M.T. Pazienza. Learning textual entailment from examples. In Proceedings of the second PASCAL Challenges Workshop on RTE, 2006
9. S. Harmeling, 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
10. E. Marsi, E. Krahmer, and W. Bosma. Dependency-based paraphrasing for recognizing textual entailment. In Proceedings of ACL PASCAL Workshop on Textual Entailment and Paraphrasing, 2007
11. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. arXiv preprint arXiv:1412.1632. 2014
12. N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In Proceedings of ACL, 2014.

13. R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of EMNLP-CoNLL, 2012.
14. K. M. Hermann and P. Blunsom. Multilingual Models for Compositional Distributional Semantics. In Proceedings of ACL, 2014.
15. M-Y. Kim, Y. Xu, and R. Goebel. Alberta-KXG: Legal Question Answering Using Ranking SVM and Syntactic/Semantic Similarity. JURISIN Workshop, 2014
16. Wen-tau Yih, Xiaodong He, and ChristopherMeek. Semantic parsing for single-relation question answering. In Proceedings of ACL, 2014.
17. G. E. Dahl, T. N. Sainath, and G. E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout. In Proceedings of Acoustics, Speech and Signal Processing (ICASSP), pp. 8609-8613, 2013
18. L. Deng, O. Abdel-Hamid, and D. Yu. "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion." In Proceedings of Acoustics, Speech and Signal Processing (ICASSP), pp. 6669-6673. IEEE, 2013.
19. Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. In Proceedings of EMNLP, 2014.
20. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems pp. 3111-3119, 2013
21. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012
22. M. Kouylekov, 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

Legal Information Retrieval Task: Participation from ISM, Dhanbad

Sushmita, Ambedkar Kanapala, Sukomal Pal *

Indian School of Mines, Dhanbad, India
sush.annie2009@gmail.com, ambedkar.kanapala@live.com,
sukomalpal@gmail.com

Abstract. This paper presents a study on exploiting the existing IR tool Terrier to retrieve relevant Civil Law articles from an identified set of legal Yes/No questions. The paper describes a system submitted to Jurisin 2015 Task 1: Legal Information retrieval Task. To answer these Yes/No questions, three models *Hiemstra, BM25* and *PL2F* are used under Terrier. The system is trained in such a way that it compares the results of these three models and writes the outcome which has occurred highest number of times. The civil law articles are then retrieved by mapping the file containing the results of the models with the file having articles and pair ids of the query. This system has achieved an accuracy of over 71.16% of correct civil law articles on training data.

keywords: Legal information retrieval, Language model, Civil Law articles

1 Introduction

A legal information retrieval demands to retrieve every information related to a specific query. The paper describes participation to *NTCIR COLIEE-2015* task which aims to automatically retrieve the Civil Law articles to legal Yes/No questions by retrieving information from a collection of documents (Here, 'documents' refers to articles, paragraphs, items, or sub-items in the legal domain). The task deals with the search of static set of civil law articles with the use of previously unseen queries. An article is considered "Relevant" to a query, if it can answer Yes/No to the query, entailed from meaning of the article. The task requires reading a number of legal bar exam questions Q_1,Q_2,...,Q_m , and the extraction of Japanese Civil Code articles S_1,S_2,...,S_n.
For a given query Q_i, a subset of the articles "S_1,S_2,...,S_n" are retrieved. This task requires the retrieval of all the articles that are relevant to answering a query. Despite its great importance and large applications, only a little work has been done in this field. This Information Retrieval Task can help layman and law-professionals for easier access of Civil Law articles.

* JURIX 2015 (The 28th International Conference on Legal Knowledge and Information Systems)

The remainder of this paper is structured as follows. Section 2 focuses some of the works already done in this area. Section 3 describes in detail the approach presented. Section 4 describes about the system. In Section 5 we explain the experiments we carried out. Section 6 shows the results which we got with the training data. Section 7 is a brief discussion of the results and the errors found during experimenting with training data. Finally conclusions are described in Section 8.

2 Related Work

A number of experiments are in progress and many of them have already been done in this specific legal domain. The development of QAs ('QAs' stands for Question Answering System) is important as it deals with the issues that requires knowledge from a variety of fields such as Artificial Intelligence, Infomation Extraction, NLP and Psychology. Particulary for legal domain in the framework of the JURIX forum, the workshop Question Answering for interrogating legal documents took place in 2003. Early work in this area has been done by Tran et al. [5] who has used QAs to retrieve articles with other information related to the query. Mi-Young et al. [1] describes a system which can answer Yes/No questions relevant to a civil laws in legal bar exam.

The Graphical approach has also been used for shallow QA in legal document. This system returns a set of articles extracted from several regulations documents. Here the graph is represented by set of all articles [2].

Also the same experiment for shallow QA has been done using NLP in legal domain. Here the experiment was carried out using natural language techniques such as lemmatizing and using manual and automatic thesauri for improving question based document retrieval [3].

3 Approach

For the Legal information retrieval task, we did a number of experiments with the help of existing IR tools. We started with basic Lemur(Indri) [1] tool as the training data was in XML format and then tried with Terrier [2]. But when initial results of these tools were compared, Terrier was giving comparatively better result than Lemur. So the idea of working on Lemur was dropped and carried on with Terrier. However in Terrier, we ended up with only three models *Hiemstra, BM25* and *PL2F* among the various models as their performance was better than others. Then a voting based algorithm is applied to compare the outputs of the three models and the best is taken into account. Here, best is considered as that output which has occurred highest number of times in the three models and thus accordingly article is retrieved.

[1] http://www.lemurproject.org/
[2] http://terrier.org/docs/v4.0/

4 System Description

In this section we present our system in detail which is able to retrieve the relevant articles related to the query.

4.1 Data

Both training Data and test data is provided by *COLIEE-2015*. The experiment was carried out and tested with the training data which has 267 documents with queries in it. The task data contains 78 queries.

4.2 Tools Used

4.2.1 Lemur

Lemur is a toolkit designed to support research on using statistical language models for information retrieval task where IR includes technologies such as ad hoc and distributed retrieval, with structured queries, cross-language IR, summarization, filtering and categorization. The system's underlying architecture support all the above technologies. It supports basic text retrieval systems using language modeling methods, as well as traditional methods such as space model and probabilistic model. Lemur is free and open source software. It also includes some baseline retrieval algorithms such as TFIDF and KL Divergence for use and comparisons. It can be extended to build your own search system.
Lemur currently supports the following features:

Indexing
- English, Chinese and Arabic text
- word stemming (Porter and Krovetzstemmers)
- stopwords
- supports for TREC Text, TREC Web, plain text, HTML, XML, PDF, MBox, Microsoft Word, and Microsoft PowerPoint
- incremental indexing

Retrieval
- Supports language modeling approaches such as Indri, KL-divergence,as well as vector space, TFIDF, Okapi and Inquery
- passage and XML element retrieval
- cross-lingual retrieval
- smoothing
- relevance feedback
- structured query language

4.2.2 Terrier

Terrier is open source and a very efficient and effective platform for research and experimentation in text retrieval. Terrier is highly flexible and has multilingual properties and thus can support corpora written in languages other than English. It implements indexing and retrieval functionalities.

Indexing

Its main data structures are the direct index, the document index, the inverted index and the lexicon [4]. Direct index stores the identifiers of terms that appear

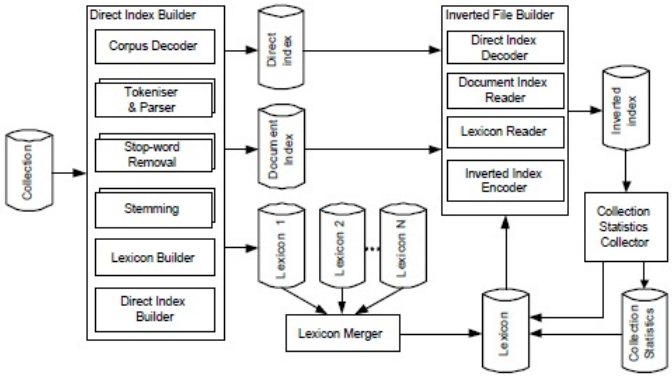


Fig. 1. Steps involved in Indexing of Terrier

in each document and the corresponding frequencies.
Document index stores information about the document length and identifier, and a pointer to the corresponding entry in the direct index.
Inverted index stores the posting lists.
Lexicon stores the collection vocabulary and the corresponding document and term frequencies.
Finally, statistics is collected about the document collection and lexicon is updated.

Retrieval

For a given query, Terrier is able to automatically select the optimal document weighting model or the appropriate retrieval approaches such as query expansion, anchor text, link analysis. The most informative terms from the top ranked documents are added to the query.

But ultimately we switched to Terrier as it was giving higher accuracy with the training data than Lemur. All the experiment was carried at term frequency normalization parameter-value 0.4 (as at this parameter value it was giving good result).

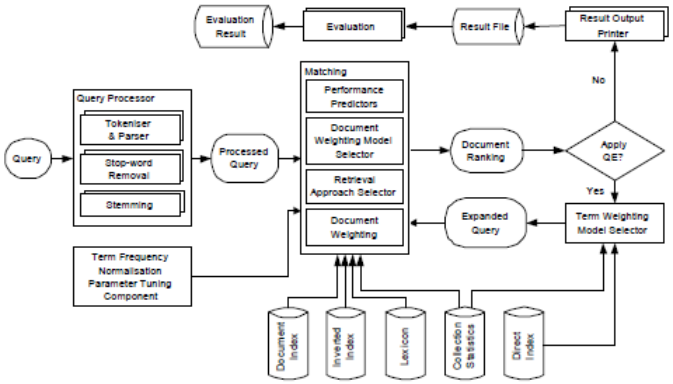


Fig. 2. Steps involved in Retrieval of Terrier

Since the title portion of the document file contains more relevant information than its text portion, so it made us to use the existing field-based model such as *BM25F, PL2F, ML2* and *MDL2*. These generally include weights for each field, namely $w.0$, $w.1$, etc. Here, $w.0$ stands for the weight of the title portion and is given five times more weight in comparison to $w.1$ which stands for the weight of the text portion of the document file i.e

$$w.0 = 10 \text{ and } w.1 = 2$$

Some per-field normalization models, such as *BM25F and PL2F* were also tested during the experiment. So they require normalization parameters for each field, namely $c.0$, $c.1$, and so on. Here the values used for normalization parameter is taken as

$$c.0 = 1.0 \text{ and } c.1 = 2.3$$

for carrying out the experiment. Here $c.0$ is normalization parameter of title portion and $c.1$ is normalization parameter of the text portion of the document file.

5 Experiments

5.1 Preprocessing

The preprocessing is done with the training data and following steps were carried out :

– The query and the document parts are separated out from the training data and are divided into chunks of files.

– The name of each document chunk was named with their corresponding pair ids of the query.
– A text file containing pair ids and the articles is also separated.
– Terrier is then used for further experiment.

An abstract approach of the preprocessing is shown through a flow chart:

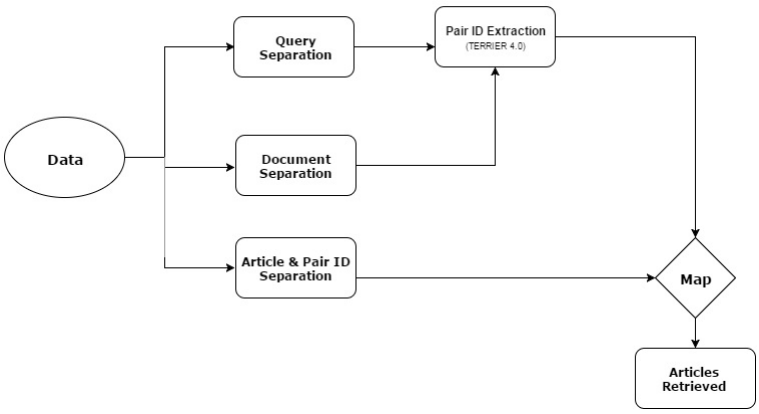


Fig. 3. Steps involved in articles Retrieval

5.2 Retrieval

Models such as *Hiemstra, BM25* and *PL2F* under Terrier are applied on extracted documents using text query set and results are produced. The results of the models are then combined using voting algorithm.

Voting Algorithm:
step 1: Open three input files f_i(i=1,2,3) and output file.
step 2: Read the line from each input file f_i.
step 3: If $line_1$=$line_2$ && $line_1$=$line_3$
step 4: Write $line_1$ in the output file.
step 5: else if $line_2$=$line_3$
step 6: then write $line_2$
step 7: else write $line_1$
step 8: If more lines then goto step 2.

– Here in the algorithm f_i(i=1,2,3) stands for the file containing the output of *Hiemstra, BM25* and *PL2F*.
– Each line of each files are read and their results are compared.

– The extracts having highest number of voting are taken into consideration but if all the three models have different results then the result of *Hiemstra* was considered.
– After using this voting algorithm, a number of pair ids are extracted in a text file.
– The content of this file is then mapped with the file containing Pair ids and articles.
– Finally articles related to the queries are retrieved.

6 Experiment Results

The result we got with the training data are summarized as follows:

Table 1. result of training data

IR Models	**Accuracy (%)**
Hiemstra	72.65
BM25	68.53
PL2F	64.41
Final Result	71.16

7 Discussion

Under Legal Information Retrieval Task, Terrier models *Hiemstra* gave the best result followed by *BM25* and *PL2F*. Finally all these three models are combined with the hope of getting a better result than *Hiemstra* but a slight decrease in percentage has occurred after this Voting Algorithm approach with the training data. But on an average, this approach will definitely be helpful in retrieving correct legal articles. We have trained our system with and without stopwords removal. But we couldn't see any changes i.e the results were same. So we used stopwords in Terrier.

7.1 Error Analysis

During the experiment, a discrepancy was noticed when a number of subsequent queries were retrieving the same document as the top ranked document. However the result of all the these queries were not same. So for some query it was giving correct result but subsequently leads to erroneous result for other queries.

8 Conclusion

Our system was based on an approach that follows the existing IR tools guidelines to extract the articles. Regardless of the fact that our system showed a slight decrease in percentage of accuracy of the result in comparison with the existing model *Hiemstra* in Terrier, we consider that the approach is suitable to deal with legal information retrieval task on an average.

References

1. Kim, M.Y., Xu, Y., Goebel, R., Satoh, K.: Answering yes/no questions in legal bar exams. In: New Frontiers in Artificial Intelligence, pp. 199–213. Springer (2014)
2. Monroy, A., Calvo, H., Gelbukh, A.: Using graphs for shallow question answering on legal documents. In: MICAI 2008: Advances in Artificial Intelligence, pp. 165–173. Springer (2008)
3. Monroy, A., Calvo, H., Gelbukh, A.: Nlp for shallow question answering of legal documents using graphs. In: Computational Linguistics and Intelligent Text Processing, pp. 498–508. Springer (2009)
4. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Advances in Information Retrieval, pp. 517–519. Springer (2005)
5. Tran, O.T., Ngo, B.X., Le Nguyen, M., Shimazu, A.: Answering legal questions by mining reference information. In: New Frontiers in Artificial Intelligence, pp. 214–229. Springer (2014)

An Approach for Retrieving Legal Texts

Vu Duc Tran, Anh Viet Phan, Long Hai Trieu, and Minh Le Nguyen

Japan Advanced Institute of Science and Technology
{vu.tran,anhphanviet,trieulh,nguyenml}@jaist.ac.jp

Abstract. Retrieving legal texts is an important step for building a question answering system on law domain, which finds relevant articles to answer a query. In this paper, we use hidden Markov model (HMM) as a generative query model for legal information retrieval. Experimental results show that using HMM is a promising approach for building an effective legal information retrieval system.

Keywords: Question Answering, Legal Information Retrieval, Retrieving Legal Text, Hidden Markov Model

1 Introduction

Information retrieval (IR) is the first step and plays an important role in a legal Question Answering (QA) system. For a legal bar exam question Q, the task is to extract a subset of articles $S_1, S_2, ..., S_n$ that are appropriate for answering the question. A core part of the task is selection of an adequate retrieval model that fits the specific characteristic of the supplied data. Applying an inadequate retrieval function would return a subset of paragraphs where the answer could not appear, and thus the subsequent techniques applied in order to detect the answer within the subset of candidates paragraphs would fail. Therefore, performance of an information retrieval subsystem serves as an upper bound for the performance of a QA system [18].

A popular method in IR is Vector Space Model in which a set of documents is represented as vectors [8]. Another method has been also applied in this task is Topic Modelling as proposed in Blei et al. [2]. Miller et al. [9] present a method for IR using Hidden Markov Models (HMMs) and develop a general framework for incorporating multiple word generation mechanisms within the same model. A more semantically contentful feature set is generated from the Stanford typed dependencies representation [5] (Semantic Dependency methods). Word Alignment, Title Matching are also methods that contribute to solve this task. Applying Language Modelling in IR is described in[1], [13].

In this work, we use HMM as a generative query model. The main idea of using HMM in IR is selecting relevant articles for each user question based on conditional probability P(document D is Relevant|question Q). This value are calculated by HMM. We estimated probabilities for not only individual articles but also combinations of pairs of articles to enhance the accuracy. This framework includes three steps. Firstly, each question is separated into words and

removed stopwords. Secondly, probability estimation, $P(D_k \text{ is Relevant}|Q)$, is calculated for all articles and the question Q. We also combine some candidate articles to calculate probabilities and perform re-ranking the list of sorted articles. Finally, we determine relevant articles for each question by a given threshold (obtained from training stage).

The rest of our paper is structured as follows. First, we survey some methods used in IR and discuss applying these methods for retrieving legal texts (Section 2). Our system which uses HMM for legal IR is described in detailed in Section 3. In Section 4, we explain experiments conducted in our system. A discussion about experimental results of combining some methods for legal IR is also in Section 4. Section 5 describes related works. Finally, Section 6 is conclusions and our future works.

2 IR Methods in Our Experiments

First of all, we survey some methods used for IR and try to apply these methods for legal IR. Then, we choose using HMM in our system, which is described in Section 3.

2.1 Vector Space Model

We implement Vector Space Model[16], popular for information retrieval, as the baseline method. Articles and queries are represented as vectors size of n, the number of distinct words in the civil code. The word weights in the article vectors are calculated by adopting term frequency-inverse document frequency model. Query vectors, instead, are in binary representation which the word weight is either 1 if the word appears in the query or 0 otherwise. The relevance of article d to query q is measured by the cosine similarity of their vectors, where the higher the similarity the more relevant (Equation 8).

$$\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n\} \tag{1}$$

$$v^{(d)} = (w_1^{(d)}, w_2^{(d)}, ..., w_n^{(d)}) \tag{2}$$

$$w_i^{(d)} = tf_i^{(d)} * idf_i \tag{3}$$

$$tf_i^{(d)} = \text{number of times word } \mathbf{w}_i \text{ appearing in article } d \tag{4}$$

$$idf_i = \log \frac{|\{d\}|}{1 + |\{d \ | \mathbf{w}_i \in d\}|} \tag{5}$$

$$v^{(q)} = (w_1^{(q)}, w_2^{(q)}, ..., w_n^{(q)}) \tag{6}$$

$$w_i^{(q)} = \begin{cases} 1 : \text{if } \mathbf{w}_i \in q \\ 0 : \text{otherwise} \end{cases} \tag{7}$$

$$\text{relevance}(d, q) = \text{cosine}(v^{(d)}, v^{(q)}) \tag{8}$$

- d: an article
- q: a query
- $\mathbf{W}$: the set of distinct words in the civil code
- $w_i^{(d)}, w_i^{(q)}$: word weights of word $\mathbf{w}_i \in \mathbf{W}$ for article d and query q respectively
- $v^{(d)}, v^{(q)}$: vector representations of article d and query q respectively

2.2 Topic Modeling

Legal articles may refer to various topics, which yields differences in topic distribution. We, then, model this property by training a Latent Dirichlet Allocation (LDA) model [2], expecting that a query has similar topic distribution to its related articles and different from the unrelated. In the training phase, the set of articles is the input corpus with each stopwords removed article as one document. In the inference phase, the articles and queries are inferred from the trained LDA model to get topic distribution vectors. Cosine similarity of those vectors measures the topic relevance between the queries and the articles (Equatiion 9).

$$\text{topic-relevance}(d, q) = \text{cosine}(\text{topic-distribution}(d), \text{topic-distribution}(q)) \tag{9}$$

2.3 Word Alignment

We observe from training data that some queries share no word or few words with their related articles. We try to model the link between the queries and the related articles in such case by applying Word Alignment technique in Statistical Machine Translation[3]. We treat each related article as a foreign document to be translated to the query. After training, we obtain a word alignment table which gives the score to link from each word in the articles to each word in the query.

The alignment score from an article to a query is calculated in Equation 10. For each word $\mathbf{w}_j$ in query q, we find the word $\mathbf{w}_i$ in article d that is best aligned to by computing $\max_i w_i^{(d)} * w_j^{(q)} * \text{align}(\mathbf{w}_i, \mathbf{w}_j)$, where $w_i^{(d)}$, $w_j^{(q)}$ are word weights computed by Equation 3, 7 respectively and align($\mathbf{w}_i$,$\mathbf{w}_j$) is the alignment score from the alignment table. The sum of those weighted word alignment forms the article-query alignment score.

$$\text{align}(d, q) = \sum_j \max_i w_i^{(d)} * w_j^{(q)} * \text{align}(\mathbf{w}_i, \mathbf{w}_j) \tag{10}$$

2.4 Semantic Dependency

A more semantically contentful feature set is generated from the Stanford typed dependencies representation [5]. The feature set is built based on Vector Space Model, Topic Modeling and Word Alignment with bi-gram dependencies $\mathbf{w}_x \xrightarrow{\texttt{dep}} \mathbf{w}_y$, extracted from the representation as "words". In other words, a query or an article is transformed into sentences of bi-gram dependencies. The "word" weighting is, then, bi-gram dependency weighting computed by multiplying of each component word weight.

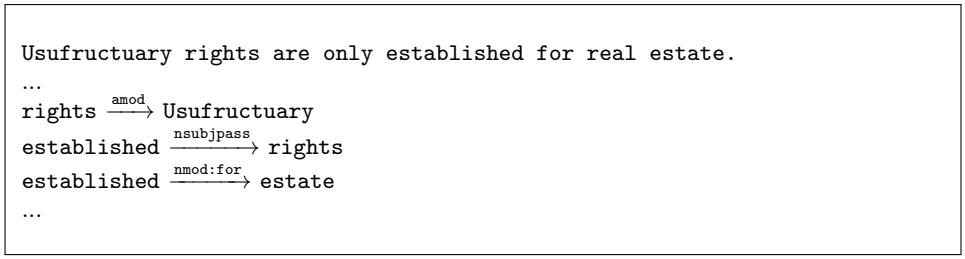


Fig. 1. A Sample of bi-gram dependencies extracted from a sentence.

2.5 Title Matching

The title of an article contains significant words, though, may not appear in the article's content or in different words . This characteristic, then, is separated into a feature describing when some queries mention the article's title and may not any words in the article's content. The matching score is a combination of uni-gram and bi-gram matching with more weighting on the longer matches (Equation 11). The score is calculated by using the same word weighting as in Equation 3 and 7.

$$\text{title-matching(d,q)} = \sum_{N=1}^{2} N \sum_{i} \sum_{n=0}^{N-1} w_{i+n}^{(d)} * w_{i+n}^{(q)} \tag{11}$$

2.6 Language Model

A document is a good match to a query if the document model is likely to generate the query, which will in turn happen if the document contains the query words often [8]. A language model is a function that puts a probability measure over strings drawn from some vocabulary.

$$P(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_1t_2)P(t_4|t_1t_2t_3) \tag{12}$$

(Special Provisions for Monetary Debt)

Article 419
(1)The amount of the damages for failure to perform any obligation for the delivery of any money shall be determined with reference to the statutory interest rate;provided, however, that, in cases the agreed interest rate exceeds the statutory interest rate, the agreed interest rate shall prevail.
(2)The obligee shall not be required to prove his/her damages with respect to the damages set forth in the preceding paragraph.
(3)The obligor may not raise the defense of force majeure with respect to the damages referred to in paragraph 1.

Fig. 2. An article with its title's words not appearing in its content.

We use unigram language model in our system.

$$P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4) \tag{13}$$

In our system, we construct from each document d in the collection a language model M_d. Our goal is to rank documents by $P(d|q)$, where the probability of a document is interpreted as the likelihood that it is relevant to the query q.

Using Bayes rule, we have:

$$P(d|q) = P(q|d)P(d)/P(q) \tag{14}$$

The retrieval ranking for a query q under the basic LM for IR is given by:

$$P(d_i|q) = \prod_{t \in q}((1-\lambda)P(t|M_c) + \lambda P(t|M_{d_i})) \tag{15}$$

Where:

- d_i: i^{th} document
- M_c: a language model built from the entire document collection
- M_d: language model of document d
- $0 < \lambda < 1$: a parameter to smooth probabilities in our document language models: to discount non-zero probabilities and to give some probability mass to unseen words

An example for applying language model in IR is showed in Figure 3.

2.7 Hidden Markov Model

We apply HMMs for retrieving relevant Civil Law articles for given questions. Given a question Q, we ranked all the articles according to the probability that D (the article) is relevant, conditioned on the fact Q, i.e. $P(D \text{ is } R|Q)$.

Applying the Bayes' rule, we may obtain a more easily estimated formula:

The document collection contains two documents:
d_1: `Xyzzy reports a profit but revenue is down`
d_2: `Quorus narrows quarter loss but revenue decreases further`

For a query:
q: `revenue down`

For unigram language model and $\lambda = 1/2$
Then:
$P(d_1|q) = [(1/8 + 2/16)/2] * [(1/8 + 1/16)/2] = 1/8 * 3/32 = 3/256$
$P(d_2|q) = [(1/8 + 2/16)/2] * [(0/8 + 1/16)/2] = 1/8 * 1/32 = 1/256$
So, the ranking is $d_1 > d_2$

Fig. 3. An example of applying language model in IR [8]

$$P\left(D\ is\ R \mid Q\right) = \frac{P\left(Q \mid D\ is\ R\right).P(D\ is\ R)}{P(Q)} \tag{16}$$

where:

+ $P\ (Q|D$ is $R\ )$ is the probability of the question, under the hypothesis that the article is relevant.

+ $P\ (D$ is $R\ )$ is the prior probability that article D is relevant.

+ $P\ (Q)$ is the prior probability of question Q.

We assume that $P\ (D$ is $R\ )$ and $P\ (Q)$ are constant across all articles. Therefore, we only focus our attention on the remaining term $P(Q|D$ is $R\ )$ when sorting articles. We constructed a model to calculate the generation probability of a question depended on an article as a discrete hidden Markov.

A discrete hidden Markov model is defined by a set of output symbols, a set of states, a set of probabilities for transitions between the states, and a probability distribution on output symbols for each state. An observed sampling of the process (i.e. the sequence of output symbols) is produced by starting from some initial state, transitioning from it to another state, sampling from the output distribution at that state, and then repeating these latter two steps. The transitioning and the sampling are non-deterministic, and are governed by the probabilities that define the process. The term of "hidden" refers to the fact that an observer sees only the output symbols, but does not know the underlying sequence of states that generated them (since the same symbol could come from any of the states) [15].

In the present application, we collected all articles to construct the corpus. After that, we take the union of all words in the corpus as the set of output symbols and introduce a separate state for each of several mechanisms of question word generation. There is a separate process for each individual article which generates the words of the question by traversing a random sequence of states, and at each state producing a word according to the output distribution of the state. For each question, we can easily compute the probability of its being

produced by each of the articles in the corpus. This is the $P(Q|D \text{ is } R)$ term that appears in Equation 16.

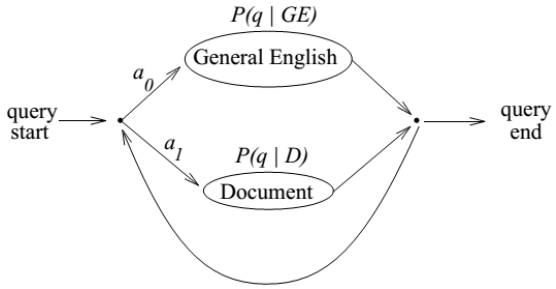


Fig. 4. A simple two-state HMM

In our application, we used a simple but powerful two-state HMM shown in Figure 4. The first state, labelled "Document", represents choosing a word directly from the article. The second state, labelled "General English", represents choosing a word that is unrelated to the article, but that occurs commonly in corpus. After observing the data, we found that the forms and words in user's questions might be different from real laws. Therefore, the statistical methods using exact word will not have high accuracy. Thus, it is necessary to expand the questions by applying word generation mechanisms involving synonyms, topic lexicons, or proper names (Figure 5)[9].

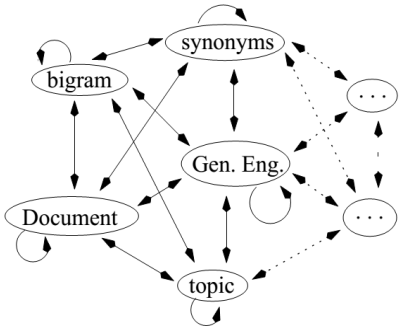


Fig. 5. An expanded multi-state HMM

The two-state HMM model is simplified that there are only two transition probabilities, a_0 and a_1 instead of the possible four. This corresponds to an assumption that the choice of which kind of word to generate next is independent of the previous such choice. This assumption allows us to introduce two

null states (states producing no output) and simplify the drawing of the model. In order to use the proposed HMM model, we have to estimate the transition probabilities and the output distributions for every article in the corpus, since we have a separate HMM for each article. We ranked all articles for each query based on generation values calculated by HMM. Relevant articles are selected by two criteria including a threshold and top k relevance.

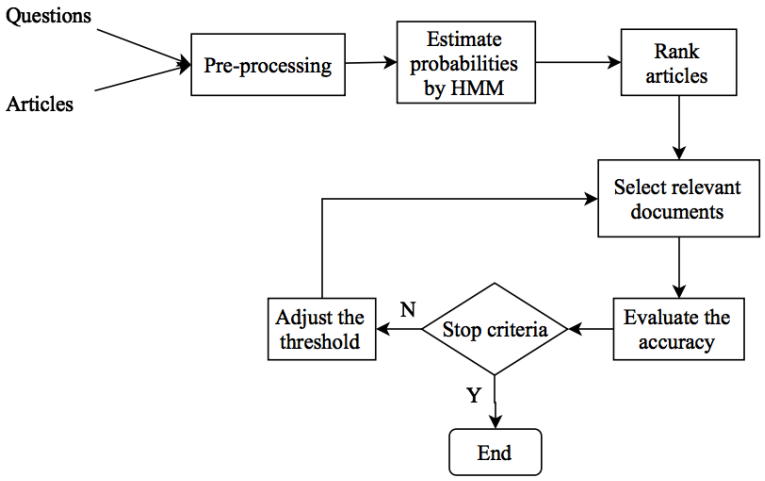


Fig. 6. A proposed framework to retrieve relevant articles for Legal QA System Using HMM

3 Applying Hidden Markov Model In Legal IR

HMM is used to calculate conditional probability of $P(D \text{ is Relevant}|\text{given } Q)$ for selecting relevant articles for each user question. We estimated probabilities for not only individual articles but also combination of pairs of articles to enhance the accuracy. Firstly, each question is separated into words and removed stop word. Probability estimation, $P(D_k \text{ is } R|Q)$ is then calculated for all articles and the question Q. Some candidate articles are also combined to calculate probabilities and perform re-ranking the list of sorted articles. Finally, relevant articles for each question are chosen by a given threshold (obtained from training stage).

3.1 Pre-processing

The main purpose of this part is preparing data for further computation. We use Stanford Tokenizer tool[1] for word segmentation. We indexed each articles

[1] http://nlp.stanford.edu/software/corenlp.shtml

separately to create inverted index files recording the number of times word appears in each article. In addition, we created and indexed a set of union of all words in a corpus. The high frequency words are extracted from the corpus to create a list of "stop" words. We ignore such words in both articles and questions when estimating probabilities.

3.2 Probability Estimation

The output distribution for the "Document" state, $P(q|D)$, is set to be the sample distribution on words appears in the article. For article D_k we set:

$$P\left(q \mid D_k\right) = \frac{number\ of\ times\ q\ appears\ in\ D_k}{length\ of\ D_k} \tag{17}$$

The probability of the second state is estimated by the sample distribution of the entire articles in the corpus:

$$P\left(q \mid GE\right) = \frac{\sum_k number\ of\ times\ q\ appears\ in\ D_k}{\sum_k length\ of\ D_k} \tag{18}$$

Where the sum is taken over all articles in the corpus.

With these parameters, we may compute the generation probability $P(Q|D_k$ is $R)$ corresponding to proposed HMM model by the formula as follows:

$$P\left(q \mid D_k\ is\ R\right) = \prod_{q \in Q} \left(a_0 P\left(q|GE\right) + a_1 P(q|D_k)\right) \tag{19}$$

This expression is used in Equation 16 to compute $P(D_k$ is $R|Q)$ values, which are used to rank articles for each query.

3.3 Collecting Relevant Articles

For each query, all articles in corpus are sorted based on generation values. After that, we put top k ranked articles into the set of candidates. In practice, an user's question may be answered by several articles. With expectation of getting information for users as much as possible, we tried to find combination of pairs of articles which give higher generation probabilities and performed re-ranking the articles. We only performed this work in the candidate set in order to reduce computation time by ignoring low probability articles. To retrieve, we use a threshold value to determine which articles are selected.

In the training stage, we set $k = 10$ and initialize the threshold value is the maximum value of generation probability in the candidate set. Using labelled training data, we evaluate the retrieval result for each value of the threshold and adjust it in order to maximize the objective function ($F - measure$).

4 Experiments

4.1 Evaluation measure

We use the $F-measure$ to evaluate the performance of the retrieval system. It considers both the precision P and the recall R of the test to compute the score. In retrieval system with binary classification (relevant and not relevant), R is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents, while precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search. R, P and $F-measure$ are calculated as follows:

$$P = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{20}$$

$$R = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}}{|\{relevant\ documents\}|} \tag{21}$$

$$F-measure = \frac{2 * P * R}{P + R} \tag{22}$$

4.2 Results

The labelled training data (all pair of queries and their relevant documents) is used for training HMM model to find the value $a_1 = 0.3$, $a_0 = 1$ - a_1 and an appropriate threshold . We ranked articles for each query using the HMM measure of Equation 19. Top n articles and their values for each query are collected into candidate and threshold set, respectively. In the candidate set, we try to merge some pair of articles and performed re-ranking them. To find the suitable threshold value, we traversed all possible values in the threshold set and get one which has greatest value of objective function(Equation 22).

From the experimental results, the training set includes 267 questions and the total of 325 their relevant articles. We used all of them and above settings to conduct the experiment. The best value of $F-measure$ is 52.0 %.

4.3 Combining Features in Legal IR

We try to find a better combination of methods by conducting experiments on different feature sets. Each combination is a subset of fully combined feature set generated by removing one feature (Table 3). In the experiments, we treat the problem of relevant article retrieval as a classification problem with relevant articles being the positive instances. Support Vector Machine [4] is selected as the classification model. Before supplying the training set, each feature is standardized with 0 mean and 1 variance. In the training phase, in order to deal with the imbalanced dataset (Table 1), we adjust the cost matrix by increasing the cost of misclassifying positive instances (Table 2). We evaluate the combination by applying the models back on the training data.

Table 1. Training set statistic

Label	No. Instances
Positive	324
Negative	200,506

Table 2. Cost Matrix

Predict/Actual	Negative	Positive
Negative	0	7.0
Positive	1.0	0.0

When we conduct experiments of combining features on training dataset (not cross-validation), experimental results are shown in Table 4. The combinations show better performance than the baseline ($> 60\%$ vs. 42.8%). On top of that, the fully combined feature set performs the best (73.0%). In other evaluating aspect, the combination of word based features gives an improvement in which the $F - measure$ increases from 42.8% to 63.5%, and the addition of bi-gram dependency based features further improves up to 73.0% which are 36.2% and 26.0% error reductions respectively. The error reduction shows the effectiveness of using semantic relation as text representation.

Table 3. Feature Set

Feature	**Description**
F1	word based Vector Space Model
F2	N-gram
F3	Title Matching
F4	word based Topic Modelling
F5	HMM
F6	word based Word Alignment
F7	bi-gram dependency based Vector Space Model
F8	bi-gram dependency based Topic Modelling
F9	bi-gram dependency based Word Alignment
All	combination of features from F1 to F9
All-Fi	combined feature set with removal of Fi

Table 4. Performance on training set

Feature Set	**F-Score (%)**
F1	42.8
All-F1	62.7
All-F2	69.3
All-F3	64.5
All-F4	69.3
All-F5	70.9
All-F6	69.7
All-F7	71.4
All-F8	70.5
All-F9	70.5
All-F7-9	63.5
All	73.0

However, the strategy of combining methods show a worse result on 10-fold cross-validation training dataset, in which F-score gets 23.2%. A better set of features should be explored in our future works.

5 Related Work

There are many QA studies in the legal field. The first one is ResPubliQA2009 [12]. There is another system for QA of legal documents reported by Monroy et al. [11]. In [14], Paulo et al. present a QA system for Portuguese juridical documents. M.-Y. Kim et al. [7] have proposed a method to answer yes/no questions from legal bar exams related to civil law. O.T. Tran et al. [19] present a study on exploiting reference information to build a question answering system restricted to the legal domain.

Question answering can be viewed as a sophisticated information retrieval (IR) task [18]. The three most used models in IR research are the vector space model, the probabilistic models, and the inference network model [17]. In the Vector Space Model, text is represented by a vector of terms (typically words and phrases) [17]. The family of Probabilistic Models models is based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. In the model of Inference Network Model, document retrieval is modeled as an inference process in an inference network [17]. The practical pursuit of computerized information retrieval began in the late 1940s [8]. Earlier studies have concluded that simple word overlap measures (e.g., bag of words, n-grams) have a surprising degree of utility [6]. Stoyanchev et al. [18] use phrases automatically identified from questions as exact match constituents to search queries. Ponte and Croft [13] proposed to used Language Modelling for this task, which involves a two-step scoring procedure: (1) Estimate a document language model for each document; (2) Compute the query likelihood using the estimated document language model (directly). Berger and Lafferty [1] use methods from statistical machine translation to incorporate synonomy into the document language model. Blei et al. [2] described using Latent Dirichlet Allocation, a flexible generative probabilistic model for collections of discrete data. The task of IR is also solved by using HMMs as described in [9], [10].

6 Conclusion

This work gives a survey of some methods used for information retrieval and analyses of applying these methods for legal texts. We have proposed a system for legal information retrieval by applying HMM. We also investigate a combination of various features and show the contribution of each feature for this task. Experimental results show that using HMM give us promising results in solving the task of information retrieval for the specific domain - legal documents.

We plan to do more experiments and analyses on this set of features (discussed in Section 4.3) to improve our work. We also plan to explore other features for the task.

Acknowledgments. This work is supported partly by the grant of NII Research Cooperation and AIST's Research grant.

References

1. Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229. ACM, 1999.
2. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
3. Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.
4. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
5. Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.
6. Valentin Jijkoun and Maarten de Rijke. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 73–76, 2005.
7. Mi-Young Kim, Ying Xu, Randy Goebel, and Ken Satoh. Answering yes/no questions in legal bar exams. In *New Frontiers in Artificial Intelligence*, pages 199–213. Springer, 2014.
8. C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. An Introduction to Information Retrieval. Cambridge University Press, 2008.
9. David RH Miller, Tim Leek, and Richard M Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221. ACM, 1999.
10. Elke Mittendorf and Peter Schäuble. Document and passage retrieval based on hidden markov models. In *SIGIR94*, pages 318–327. Springer, 1994.
11. Alfredo Monroy, Hiram Calvo, and Alexander Gelbukh. Nlp for shallow question answering of legal documents using graphs. In *Computational Linguistics and Intelligent Text Processing*, pages 498–508. Springer, 2009.
12. Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forăscu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, and Petya Osenova. Overview of respubliqa 2009: question answering evaluation over european legislation. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 174–196. Springer, 2010.
13. Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.
14. Paulo Quaresma and Irene Rodrigues. A question-answering system for portuguese juridical documents. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 256–257. ACM, 2005.
15. Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
16. Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

17. Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.
18. Svetlana Stoyanchev, Young Chol Song, and William Lahti. Exact phrases in information retrieval for question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 9–16. Association for Computational Linguistics, 2008.
19. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. Answering legal questions by mining reference information. In *New Frontiers in Artificial Intelligence*, pages 214–229. Springer, 2014.

Author Index

ISBN 978-4-915905-66-7 C3004(JSAI)JURISIN