Proceedings of the Eight International Workshop on Juris-Informatics (JURISIN 2014)

hosted by JSAI-isAI

Raiosha-building Keio University, Yokohama,
Kanagawa, Japan

JURISIN 2014 Chair
Satoshi Tojo

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 2014 is the 8th International Workshop on Juris-informatics, which is held in association with the International Symposia on AI by Japanese Society of Artificial Intelligence (JSAI-isAI).

Although the number of submission is not so large, each paper was reviewed by the organizing committe, plus those program committee members who have assited organizers. As a result, we have chosen thirteen papers which cover logical inference, ontology, natural language processing, information retrieval, and so on.

As guest speakers, we have invited Riichiro Mizoguchi of Japan Advanced Institute of Science and Technology, and Bart Verheij of University of Groningen. In addition, we have planned a joint guest speaker session with the workshop of Logic and Engineering of Natural Language Semantics (LENLS).

As a special event of this year, JURISIN invites participants in Competition on Legal Information Extraction/Entailment (COLIEE). Previous conferences/workshops have not conducted such a task on a large legal data collection, so we hope that the 2014 workshop will help to establish a major experimental effort in the legal information extraction/retrieval field. The motivation for the competition is to help create a research community of practice for the capture and use of legal information.

JURISIN 2014 Chair
Satoshi Tojo

Workshop Organization

Workshop Chair

Satoshi Tojo, Japan Advanced Institute of Science and Technology, Japan

Organizing Committee Members

Katsumi Nitta, Tokyo Institute of Technology, Japan
Ken Satoh, National Institute of Informatics and Sokendai, Japan
Satoshi Tojo, Japan Advanced Institute of Science and Technology, Japan
Katsuhiko Sano, Japan Advanced Institute of Science and Technology, Japan

Advisory Committee Members

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

Program Committee Members

Satoshi Tojo, Japan Advanced Institute of Science and Technology, Japan
Tokuyasu Kakuta, Nagoya University, Japan
Hanmin Jung, Korea Institute of Science and Technology Information, Korea
Minghui Xiong, Sun Yat-sen University, China
Minghui Ma, Southwest University, China
Nguyen Le Minh, Japan Advanced Institute of Science and Technology, Japan
Katsuhiko Toyama, Nagoya University, Japan
Makoto Nakamura, Nagoya University, Japan
Marina De Vos, University of Bath, UK
Guido Governatori, National ICT Australia, Australia
Paulo Novais, University of Minho, Portugal
Seiichiro Sakurai, Meiji Gakuin University, Japan
Ken Satoh, National Institute of Informatics and Sokendai, Japan
Thomas Ågotnes, University of Bergen, Norway
Robert Kowalski, Imperial College London, UK
Katsuhiko Sano, Japan Advanced Institute of Science and Technology, Japan
Katsumi Nitta, Tokyo Institute of Technology, Japan

Katie Atkinson, University of Liverpool, UK
Philip Chung, University of New South Wales, Australia
Masahiro Kozuka, Okayama University, Japan
Fumihiko Takahashi, Meiji Gakuin University, Japan
Baosheng Zhang, China University of Political Science and Law, China
Akira Shimazu, Japan Advanced Institute of Science and Technology, Japan
Trevor Bench-Capon, University of Liverpool, UK
Mi-Young Kim, University of Alberta, Canada
Randy Goebel, University of Alberta, Canada

Reviewers

Satoshi Tojo, Japan Advanced Institute of Science and Technology, Japan
Katsuhiko Sano, Japan Advanced Institute of Science and Technology, Japan
Akira Shimazu, Japan Advanced Institute of Science and Technology, Japan
Katsumi Nitta, Tokyo Institute of Technology, Japan
Seiichiro Sakurai, Meiji Gakuin University, Japan
Hanmin Jung, Korea Institute of Science and Technology Information, Korea
Randy Goebel, University of Alberta, Canada
Mi-Young Kim, University of Alberta, Canada
Ken Satoh, National Institute of Informatics and Sokendai, Japan
(Sub Reviewers)
Ryutaro Ichise, National Institute of Informatics and Sokendai, Japan
Yusuke Miyao, National Institute of Informatics and Sokendai, Japan

JURISIN 2014 Program

Nov. 23, 2014

9:20 Opening

9:30 -10:30 Invited Talk I

Ontology engineering - Theory and practice -
Riichiro Mizoguchi (Japan Advanced Institute of Science and Technology)

Coffee Break

11:00 -12:00 [Ontology]

Ontology Design Pattern Engineering For Accumulating Evidence For Legal Reasoning: Cases From Water Energy, Food and Climate Nexus,
Md Mizanur Rahman

A Hierarchy of Legal Indices
Tho Thi Ngoc Le, Minh Le Nguyen and Akira Shimazu

13:30 -14:30 [Logic and Inference I]

Generating Legal Reasoning Structure by Answer Set Programming
Duangtida Athakravi, Ken Satoh, Krysia Broda and Alessandra Russo

Classification of Precedents by DEMO
Tetsuji Goto and Satoshi Tojo

Coffee Break

15:00 -16:00 [Logic and Inference II]

Analyzing Reliability Change in Legal Case
Pimolluck Jirakunkanok, Katsuhiko Sano and Satoshi Tojo

Module Based Argumentation Framework for Analysis of Actual Discussion Records
Kei Nishina, Shogo Okada and Katsumi Nitta

16:15 -17:15 Invited talk: shared with LENLS

Nov. 24, 2014

9:30 -10:30 Invited Talk II

The Future of Argumentation Technology, As guided by the needs of the law
Bart Verheij (University of Groningen)

Coffee Break

11:00 -12:00 [Information Retrieval, Representation I]

Quantum Artificial Intelligence for Modelling Legal Knowledge of Legal Definitions: Cases from EU Water Energy and Food Legislations and Nexus
Md Mizanur Rahman

The Applicability of Nationality Extraction in Crime Domain to Construct Legal Roles Extraction in Law Domain
Masnizah Mohd, Mohammad Darwich and Kiyoaki Shirai

13:30 -14:00 [Information Retrieval, Representation II]

Recognizing logical parts in Vietnamese Legal Texts using Discriminative Sequential Learning
Son Nguyen Truong, Duyen Nguyen Thi Phuong, Quoc Ho and Nguyen Le Minh

14:10 -15:10 [COLIEE I]

Translating Simple Legal Text to Formal Representations
Shruti Gaur, Nguyen Vo, Kazuaki Kashihara and Chitta Baral

A logic-based system for recognizing textual entailment applied to the bar exam competition
Yusuke Miyao and Ken Satoh

Coffee Break

15:40 -16:40 [COLIEE II]

JAVN system for Legal Extraction and Entailment Tasks
Nguyen Le Minh

Alberta-KXG: Legal Question Answering Using Ranking SVM and Syntactic/Semantic Similarity
Mi-Young Kim, Ying Xu and Randy Goebel

16:50 Closing

Ontology engineering - Theory and practice -

Riichiro Mizoguchi

Japan Advanced Institute of Science and Technology

Abstract. Ontology engineering provides us with powerful conceptual tools/methodologies for modeling all kinds of entities in the real world. Because its origin is philosophy, it leads us to analysis of deep conceptual structure of the target world, which eventually enables us to find solid and fundamental understanding about it. In my talk, I will discuss theory and practice of ontological engineering based on my 20-year experience. Theoretical aspect of ontological engineering includes fundamental distinctions between entities with justifications. I discuss if a typhoon and an ocean current are objects or processes, how a process and an event are different, etc. As a common topic to law and language, I discuss ontology of informational objects without going into detail of semiotics. In order to demonstrate utility of ontological engineering, I also discuss a couple of application projects conducted by my group thus far. Examples include ontology of function and its deployment, ontology of learning and instructional theories and a theory-aware authoring tool, and ontology of diseases.

The Future of Argumentation Technology, As guided by the needs of the law

Bart Verheij

University of Groningen

Abstract. Information technology is significantly changing the legal profession. Still, legal information technology is not yet creating radically new levels of value, such as a significant decrease in costs of legal services, or a radical increase of access to justice. Why not?
In the talk, it will be argued that, for legal information technology to be disruptive, innovative information technology must be developed that is well-adapted to the central information processing mechanism of the law: argumentation. It will be argued that the development of argumentation technology can help overcome the long-standing technological hurdle of how to make logical and probabilistic tools mutually supportive, instead of competitors.
As an application area, we discuss the domain of forensic science, where the rise of DNA evidence has shown how hard it is to safely combine probabilistic and logical reasoning. An integrated formal perspective on reasoning with criminal evidence shows how logical reasoning with arguments and scenarios can be embedded in standard probability theory. The perspective can be the basis of argumentation technology that combines the strengths of logical and probabilistic tools.

Ontology Design Pattern Engineering For Accumulating Evidence For Legal Reasoning: Cases From Water Energy, Food and Climate Nexus*

Md Mizanur Rahman

Erasmus Mundus Doctoral Fellow, LAST-JD program, CIRSFID, University of Bologna, mdmizanur.rahman2@unibo.it and mizanur3@gmail.com

Abstract. The legal reasoning, which is an entangled form of legal rules and their associated evidences, for environmental cases primarily requires evidences of those fundamentally accumulated and verified by various environmental technologies. Therefore, ontology design pattern engineering for accumulating such evidences from environmental Big Data for legal reasoning of environmental cases might advance environmental, GIS and big data technologies one step ahead to implement legal reasoning of environmental cases in real space, time and legal setting. Under considering these perspectives, the paper briefly explains the legal reasoning process for environmental cases which is fundamentally different from criminal, civil or tort cases and how ontology pattern design can contribute in this process. Then it presents a number of computational ontology design patterns of water-energy-food-climate nexus.

Keywords: Computational Ontology. Ontology Design Patterns for Legal Reasoning. Water Energy Food nexus. AI & Law.

1 Introduction

Legal reasoning is a formal proofing process that encompassed with legal rules and their associate evidence [1][2]. Fundamentally all legal arguments can be processed and entangled[1] with following isolated essentials – (a) issue, that is a specific legal debate, (b) legal rule, what legal rules govern and/or connected with that debated issue, (c) facts, that is relevant to the legal rules, (d) legal analysis, that is the application of the legal rules to the facts, and (e) legal conclusion, that is an outcome after applying the legal rules to the facts[3][4]. Unlike legal reasoning process for criminal, civil, land and/or trot cases, legal reasoning process of environmental cases are mostly technology driven [5]. Because, all elements for legal reasoning of an environmental case, in most

* This paper is a product of a doctoral project in Erasmus Mundus Joint International Doctoral Degree in Law, Science and Technology, coordinated by CIRSFID, University of Bologna, and submitted to both conferences – JURIX 2014 and JURISIN 2014.

[1] Entangled state is known as not decomposable sate. See Timpson C.G. 2004. Quantum information theory and the foundation of quantum mechanics. Oxford: University of Oxford. 126 p. and Wichert A. 2014. Principles of quantum artificial intelligence. Singapore: World Scientific Publishing Co. 126 p.

cases if not all, can be automatized and performed in real space, time and legal setting. And throughout this process, computational ontology design pattern (ODP) engineering based on scientifically verified and tested intelligence might play a leading role. Considering these perspectives, this paper will first briefly discuss about the fundamental elements of the legal reasoning process and how ODP may play its role. Then subsequently it presents a number of computational ODPs for Water-Energy-Food (WEF) nexus[2], which is consisted with very isolated and legally distinct domains and also internationally recognized a public policy concern.

1. Legal Reasoning and Roles of Legal Ontology and Ontology Design Patterns in Environmental Cases

By and large, the legal reasoning process for all types of legal prosecutions is built upon – identification of *issue* that is legally debatable, application of *legal rules*, *facts* from evidence in order to support the legal rules, *legal analysis* through legal arguments that is consisted with legal rules and facts, and *legal conclusion* that is the outcome of the legal arguments[3]. Unlike criminal, civil, land and/or trot cases, however, in the environmental suits, environmental technologies[4] like GIS, technology for water and/or air quality assessment play very important role in the legal reasoning process by accumulating evidence in real space and time setting [5] [6]. In according to the current state of art of environmental technologies, the computational ontology design pattern (ODP) is not an architectural consideration in each of these existing technologies, but executions of ODPs and legal ontologies may automatize the entire legal reasoning process for environmental suits more effectively and effectively in not only real space and time setting, rather it may also consummate in real legal setting. The main essentials of a legal reasoning process are discussed very briefly in below –

1.1. Legal Issue

The legal issue is such an issue on what law has something to lead[5]. It may emerge from any corner of social intelligence, but it must have concreate connection with legal rules. However, detection and verification of such issues are highly intellectual task and mainly performed by human legal expert. But a well-formulated legal ontology and ontology pattern designs can play an important role in order to set up a legal setting, besides the real space and time setting. That might automatize the entire information and communication technology as to compliance with all legal requirements and rules.

1.2. Legal Definition, principle and Rule

[2] Bazillian M, Rogner H, Howells M, Hermann S, Arent D, Geilen D, Steduto P, Mueller A, Komor P, Tol RSJ. Yumkella KY. 2011. Considering the energy, water and food nexus: towards an integrated modelling approach. Energy Policy 39(12):7896-7906 p.

[3] See http://groups.csail.mit.edu/dig/TAMI/inprogress/LegalReasoning.html

[4] [WWAP] United Nations World Water Assessment Programme. 2014. The United Nations World Water Development Report 2014: Water and Energy. Paris, UNESCO.

[5] PattersonD, editor.2010. A companion to philosophy of law and legal theory. Singapore: Willy-Blackwell. 261 p.

A legal definition, principle and rule are like legal coins. One side of these coins is the legitimate source and content of it with its temporal and enforceable aspects and the other is the interpretation of it. Legal ontology and ontology pattern designs can play an important role in order to represent the content of a legal rule with its temporal and enforceable aspects in order to digitalize the legal definition, principle and rule. However, legal ontology may also contribute in a prescribed way in order to interpret the legal rule.

1.3. Fact

The fact that fits with legal rule is a fact useable in the legal reasoning process. The evidence based on those facts is the evidence plays a vital role to connect 'legal rule' with the social and environmental occurrence. In the case of environmental cases, the evidence is accumulated by the environmental technologies in real space and time setting. What is missing is a real legal setting into the execution process of environmental technological performance. This is the point where legal ontology and ontology design pattern can play its potential role.

1.4. Legal Analysis

In this stage, legal rule entangles with the evidence based facts. Legal argumentation plays its role in order to establish reasonable or legally admissible connection between legal rules and facts. Legal ontology and well-formulated ODPs can automatize this service in a prescribed way.

1.5. Legal Conclusion

This is the point of outcomes that emerges when legal analysis brings together legal rules and its admissible evidence and facts. Many environmental technologies already performs this task, but not in a legal setting. For example, measuring acidic level in water and carbon emission into the air. Therefore, the missing part is to include legal analysis features into the existing environmental technologies, this is where legal ontologies and ODPs can play its dynamic roles.

2. Laws And Policies Of WEFC Domains

In general, WEFC are isolated and domain oriented directed by different procedural and substantial laws and maintained by separated public administrations[6]. In particular, laws and policies of WEFC domains, on the one hand, are deterministic within their internal legal meta-structures[7], political administrative arrangements and programming

[6] Knoepfel P, Larrue C, Varone F, Hill M. 2007. Public policy analysis. Bristol: The Policy Press, University of Bristol. 151 p.

[7] [SEP] Stanford Encyclopedia of Philosophy. 2010 Jan 21. Causal determinism. <http://plato.stanford.edu/entries/determinism-causal/ > Accessed 2014 April 20.

with a number of uncertain and probable chaotic behaviors [7][8]. For example, European Food Safety Authority[8] only works with the legal definition of food and legal purposes of their institutional framework which is based on only food related legislations and public policies. On the other hand, they are non-deterministic outside of the body of their internal meta-structures. For example, Agency for the Cooperation of Energy Regulators (ACER)[9] is not responsible for initiating polluting measures of water, even though water pollution has been increasing due to waste release coming from energy production[10]. Therefore, legal reasoning process of WEFC nexus is by nature of law is complex as each of these sectors is fundamentally and strictly distinguished by law.

Furthermore, each of these sectors has their own set of technologies, generically known as environmental technology, designed for a specific set of purposes [9][10]. For example, technology for measuring carbon emissions of energy industry is only to measure carbon emission, not measuring of other dimensions of energy industry like water contemplation rate by energy industry related water release. However, the information and communication server might gather all information together in their data sever. For example, Water Information System for Europe contain XML based water related statistics in a place that comes from different countries of Europe and accumulated by different environmental technologies[11]. This phenomena is also known as Big Data[12]. Therefore, however, on the top of these information systems and/or Big Data of collective environmental statistics, legal ontology or ODPs of WEFC nexus might provide us a new way to accumulate evidences for legal reasoning of WEFC nexus. From this stand point, this paper presents a number of ODPs of WEFC nexus, those are mostly tested and verified by scientific communities, aiming to use these ODPs for accumulating evidence for legal reasoning of WEFC nexus.

3. Ontology Design Pattern Engineering of WEFC Nexus – General Usages

The construction of ODPs is primarily motivated by information logistics – that is about getting right information at the right time at the right place through the right channel [11][12][13]. In the context of WEFC domains, information logistics of WEFC nexus is yet unaddressed. However, besides accumulating evidences from the environmental Big Data for legal reasoning of WEFC nexus, ODPs of WEFC nexus can be also potentials for following reasons -

- To build a knowledge base of WEFC nexus in order to use it for multiple purposes like knowledge network of WEFC nexus, detecting nexus in Big Data etc.
- To provide common vocabularies and principles for understanding of WEFC nexus throughout the WEFC domains.

[8] It is key authority for assessing risks regarding food and feed safety. See http://www.efsa.europa.eu/

[9] An EU agency is to make progress on internal market of electricity and natural gas. See http://www.acer.europa.eu/The_agency/Pages/default.aspx

[10] Campbell A. 2008. Food, energy, water: conflicting in securities (and the rare win-wins offered by soil stewardship. Journal of soil and water conservation 63(5): 149-151 p.

[11] See http://water.europa.eu/links

[12] See http://www.ibm.com/big-data/us/en/

- To help managing various systematized documentation complexity of WEFC domains.
- To identify and specify abstraction that are above the level of single classes and instances of the component coming from each WEFC domains.
- To recognize the risks and trade-offs of WEFC nexus analyzing Big Data in real space, time and legal setting.
- To understand the effects of non-deterministic nature of WEFC domains and to detect the composite constitutive rules of WEFC domains from Big Data.
- The existing patterns of WEFC nexus may learn new patterns of WEFC nexus automatically from Big Data by using neural artificial intelligent. That may increase the library of knowledge patters of WEFC nexus time to time. This may also lead to understand more adequately the composite constitutive rules of WEFC nexus.
- By reusing the existing and new knowledge patterns of WEFC nexus may reduce the complexity of tasks and repetition of scientific studies, and improve the quality of WEFC nexus work.
- Utilization of knowledge pattern can also be an integral part of a bigger architecture for editing, validating and managing large volume of WEFC nexus documentation systematically.

4. Ontology Design Pattern Engineering of WEFC Nexus

4.1. Brief State Of Art – Ontology Design Pattern Engineering

The idea of utilizing pattern instigated in architecture and then followed by computer science inspired by information logistics system [14]. Computational ontology is one of the classical ways to design the knowledge base of information logistics system where pattern plays an important role [15]. In order to facilitate knowledge engineering for semantic web is further pushed by description logic based Web Ontology Language (OWL) full. However, knowledge pattern engineering or ontology design pattern (ODP[13]) is a new emerging filed with potentials [16][17][18]. Few of their benefits, under WEFC nexus context, have been mentioned in section 3.

4.2. Methodology

For knowledge pattern engineering of WEFC nexus, Graffoo, a graphical framework for OWL ontology, compatible with OWL Full, has been used for mainly two reasons – (a) it is an open source tool, and (b) it produces very clear and easy-to-understand diagram of ontologies (see figure 2 to 8). All patterns used in the paper are taken from the current start-of-art of WEFC nexus. It might be confusing that in many literatures these patterns are considered as synergies. To make it clear, it should be well-thought-out that synergies can also present a pattern that describes regularity in the WEFC domains. For example, energy is required to treat waste water in order to make it human

[13] See http://ontologydesignpatterns.org/wiki/Main_Page

consumable. Even though it may be addressed as synergies between energy and water, it itself is a pattern of energy-water nexus.

4.3. Ontology Pattern Design Of WEFC Nexus and Their Computational Ontological Architecture

In this section, six basic and fundamental, but technically complex, knowledge patterns of WEFC nexus is presented. Each of these patterns is consisted with a number of meta-patterns. All yellow boxes in the following figures (from 2 to 7) represent different concepts of WEFC domains and blue sentences in-between the concepts represents two meaning simultaneously – (a) object properties of the concepts and its flow is shown by arrow, (b) the semantic meaning of the relationship exist between the concepts. These patterns can be enriched more by putting restrictions and adding more object and data properties for each concept in order to make the functionality of these pattern more committed and straight-forward. These patterns can also be enlarged by adding more concepts and their associated properties.

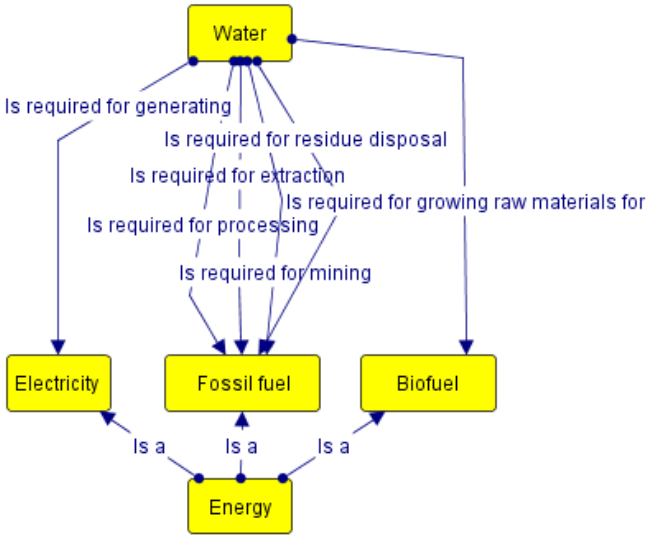


Figure 1. Pattern on water-energy nexus

4.3.1 Pattern 1 – Water is required for the extraction, mining, processing, refining, and residue disposal of fossil fuels, as well as for growing biofuels and for generating electricity [19][20][21].

In this pattern, as shown in *Figure 1*, there are two super-concepts – water and energy, and energy has three sub-concepts – electricity, fossil fuels, biofuel. There are six object properties belongs to super-concept water – 'is required for generating', 'is required for residue disposal', 'is required for extraction', 'is required for growing raw materials for', 'is required for processing', and 'is required for mining'. These object properties

make semantic links with various sub-concepts of energy. For example, water is required for generating electricity. In addition, the object property 'is a' makes clear meaning between the super-concept 'energy' with its sub-concepts. For instance, biofuel is a type of energy.

4.3.2. Pattern 2 – Energy is required for lifting, moving, distributing, and treating water and food [22][23][24].

In this pattern, as shown in *Figure 2*, there are four super-concepts – energy, water, food and 'drinking water'. There are total eight object properties belongs to energy, those describes the multiple semantic relationships flowing from energy to water and energy to food. For instance, energy is required for distributing water. The pattern also expresses that super-concept 'drinking water' is also considered as food and water.

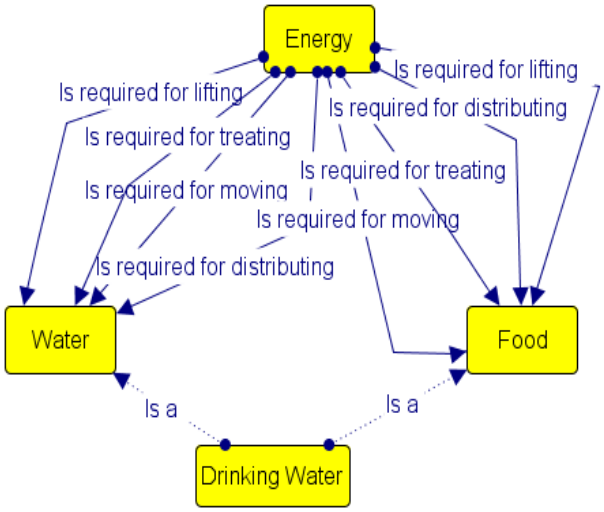


Figure 2. Pattern on Energy-water-food nexus

4.3.3. Pattern 3 – Energy is required mainly for the provision of water services and water resources are required in the production of energy [4].

There are two super-concepts in this pattern, as shown in *Figure 3*. They are energy and water. Water has two sub-concepts – 'water resources' and 'water services'. The object properties – 'is a role of' and 'is used as' establish the semantic connections between super-concept water and its sub-concepts. The object properties – 'is required for the provision of' and 'is required in the production of' bonds the semantic relationship between super-concept energy with two sub-concepts of water. The collective form of these object properties makes the pattern that exist in energy-water nexus. For instance, water service is a role of water but energy is required for the provision of water services as well as water itself is used as water resources that is required in the production of energy.

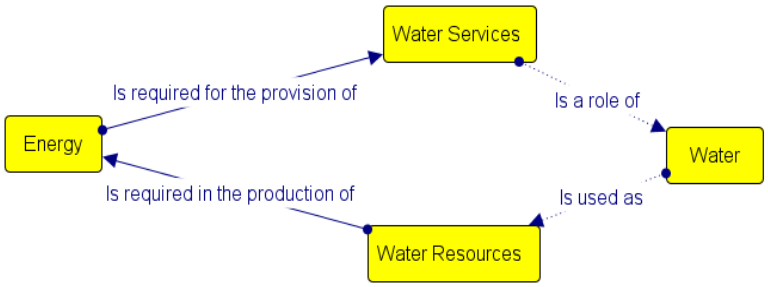


Figure 3. Pattern on energy-water nexus

4.3.4. Pattern 4 – Food production affects the quality of water bodies through agricultural runoff polluted with fertilizers, pesticides and manure from farms, fields and feedlots [25][26][27[28][29].

In this pattern, as in *Figure 4*, there are nine super-concepts – food, food farms, fields, feedlots, fertilizers, pesticides, manure, agricultural runoff and water. Super-concept water has one sub-concept water quality. One object property – 'is value of' states the semantic meaning that exist between water and water quality such as water quality is value of water. The object properties 'is prepared in' established relationship flowing from food to food farms, fields and feedlots. The object property 'comes from' describes multi-relationships between fertilizers and food farms; pesticides and food farms; fertilizers and fields; pesticides and fields; manure and feedlots. Another object properties 'pollute' describes the relationship flowing from fertilizers, pesticides and manure to agricultural runoff. In the pattern, objective property 'pollute' is same as object property 'pollutes'.

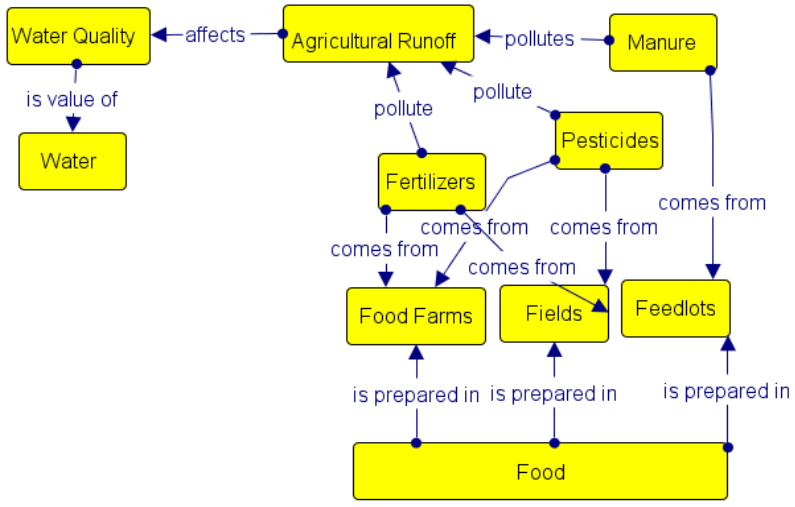


Figure 4. Pattern on water-food nexus

The other object property 'affects' states the semantic relationship between super-concept 'agricultural runoff' with the sub-concept of water that is 'water quality'. All these semantic relationships exist between super-concepts and sub-concept jointly establish multiple patterns exist between water-food nexus. For example, manure comes from feedlots pollutes agricultural runoff that affects water quality, the value of water.

4.3.5. Pattern 5 – Wasted food means wasted water, energy and agricultural resources [30] [31][32][33][34]. [Given that the water- and energy-intensive nature of growing, processing, packaging, warehousing, transporting and preparing food.]

In this pattern, as in *Figure 5*, there are three super-concepts – energy, food and water, and three corresponding sub-concepts – 'wasted energy', 'wasted food', and 'wasted water'. The object property 'count as' establish the basic pattern exist between corresponding sub-concepts – that is that 'wasted food' is equivalent to 'wasted water' and 'wasted energy'.

The object property 'is unable to be used' states the semantic meaning of corresponding sub-concepts with their associated super-concepts. For example, the food is unable to be used is wasted food. There are four objective properties gives semantic meaning of the relationship exist between two super-concepts energy and food. These properties are – 'is required for transporting', 'is required for processing', 'is required for warehousing' and 'is required for packing'. The other object property 'is required for growing' makes connection between food and water super-concepts. One of the sub-pattern from this pattern can be addressed as water is required for growing foods and energy is required for transporting food. Therefore, wasted food means wasted water and energy.

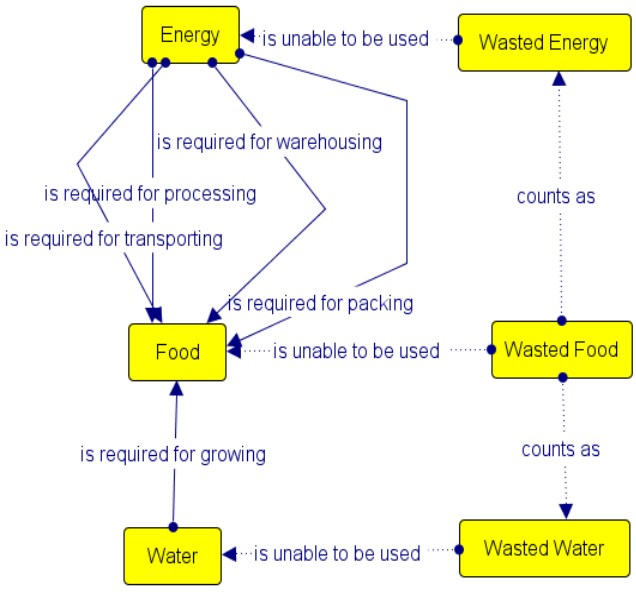


Figure 5. Pattern on energy-food-water nexus

4.3.6. Pattern 6 – Power plants can be impacted by drought when surface water levels drop, leaving them without access to cooling water and forcing the plants to reduce their operations [35][36][37][38][39][40][41].

In the pattern of water-energy-food-climate nexus, as shown in *Figure 6*, there are four super-concepts – water, energy, food and climate. There are two sub-concepts of climate – 'global warming' and 'drought'. Two objective properties 'is an expression of' and 'increases' establish relationship between climate, 'global warming' and drought such as global warming is an expression of climate that increases drought. There are four sub-concepts under the super-concept water – 'surface water', 'green water', 'blue water' and 'human consumable water'. The five object properties such as 'is a form of', 'is a kind of', 'degrades', 'depletes' and 'affects negatively' describes relationships among the super-concept water and its associated sub-concepts. Among these objective properties, the object property 'is a form of' is same as the object property 'is a kind of'.

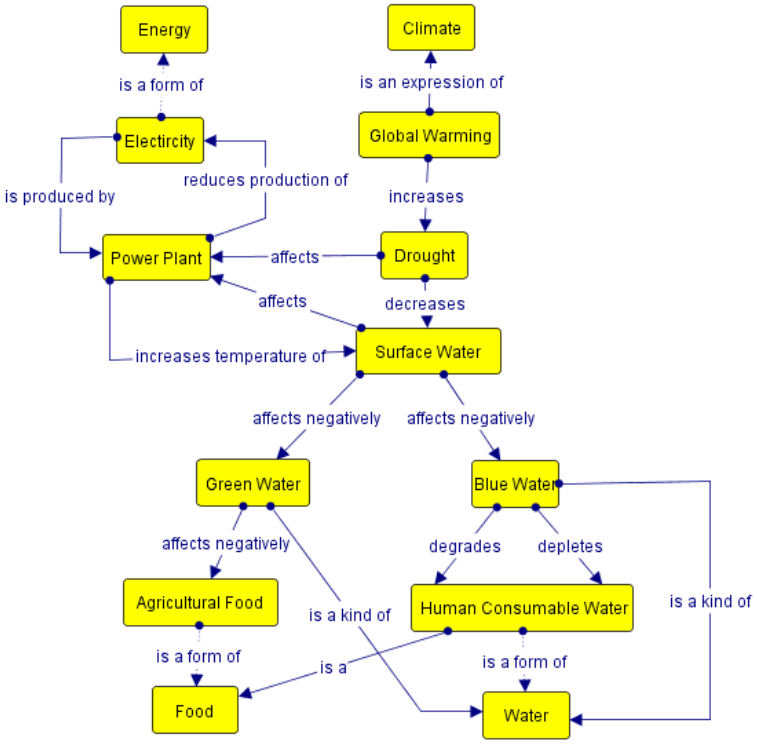


Figure 6. Pattern on water-energy-food-climate nexus

'Agricultural food' is a sub-concept of the super-concept food. The object property 'is a form of' describes the semantic relationship between these concepts. The sub-concepts – electricity and 'power plant' is associated by three objective properties – 'is produced by', 'reduces production of' and 'is a form of'. The concept 'surface water'

and ‘power plant’ is related by two object properties – ‘increases temperature of’ and ‘affects’. The another object property ‘affects’ also gives a meaning of the relationship between the concepts ‘drought’ and ‘power plant’.

One of the meta-patterns of this pattern is that increases of drought due to global warming affects power plant that reduces production of electricity and also increases temperature of surface water, which affects both green and blue water. Subsequently it affects agricultural foods and human consumable water.

5. Usefulness Of Computational Knowledge Patterns Engineering of WEFC Nexus

There are fundamentally three phases from data storage to legal reasoning of WEFC nexus, as shown in *Figure 7*. The data storage phases includes big data of WEFC nexus which is comprised with individual datasets of water, energy, food and climate. In the ontological phase, the ontology pattern designs of WEFC nexus is interweaved with ontology of legal definitions, principles and rules of WEFC domains. Then, in the legal reasoning phase, ontological contributions takes in place by identifying legal issues, applying legal definitions, principles and rules, accumulating WEFC Nexus facts, and then linking WEFC Nexus facts with legal definitions and facts.

However, the above mentioned computational ontologies of knowledge patterns of WEFC nexus can be used, in connection with computational ontologies of legal definitions of EU water, energy and food legislations, for multi-purposes using three computation platforms – classical, sub-symbolic and quantum computation.

Under the current capacities of the classical computation system based on Von Neumann architecture, there are particularly two ways can be very useful in order to utilize these knowledge patterns of WEFC nexus. They are – (a) the pattern recognition algorithm [42], which is principally designed for classical computation system, with certain manipulation, can be used to recognize these patterns of WEFC nexus in large data sets or Big Data coming from various sources like respective authorities, research institute etc. (b) machine learning algorithm [43] [44] that makes it possible to train the machine about these patterns. So that on the top of these patterns, it is possible to use numerous types of computer applications to analyze the Big Data. In addition, these patterns can also be treated as knowledge base for a number of computer-based-WEFC-nexus-application like trade-off analysis of WEFC nexus.

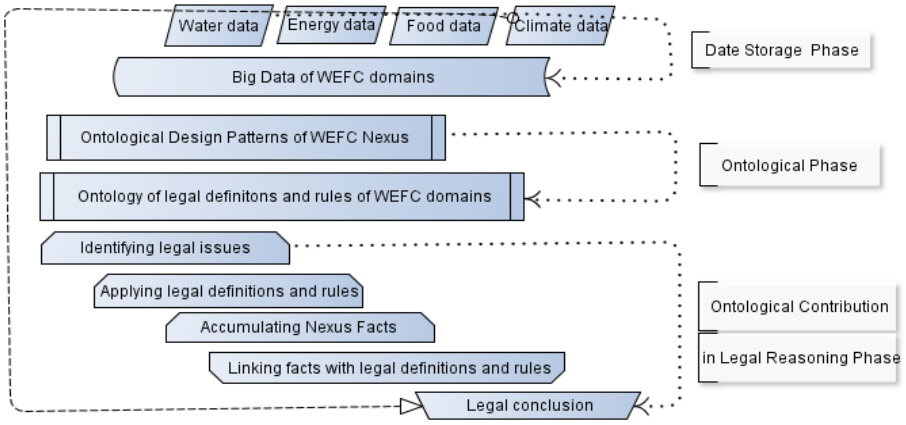


Figure 7. From Data Storage to Legal Reasoning of WEFC Nexus

Artificial neural networks inspired by animal's central nervous system is a prominent sub-symbolic computation system with the capacity of machine learning and pattern recognition [45]. The patterns of WEFC nexus can be trained to the machine in such a way that will be able to learn the new patterns from existing WEFC data set or Big Data in order to develop a comprehensive WEFC nexus's neural networks or library.

The quantum computation is an emerging computation filed inspired by the quantum mechanics with many new computation capacities like superposition of binary vector representation, entanglement, many-world theory, polynomial time performance, and iterative quantum tree search etc [46]. In the context of analyzing Big Data, quantum computer may play a big role in coming decades with its speed and new functionalities [47][48]. It is also possible to train quantum machine about the patterns of WEFC nexus [49][50] and then to use artificial neural networks in order to build more robust library of the patters of WEFC nexus automatically from the Big Data.

5. Concluding remark

By the nature of law and public administration, on one hand, WEFC sectors are isolated and domain oriented. That arises a number of complexities in the legal reasoning process of WEFC nexus. Environmental Big Data, on the other hand, is a collection of enormous vital facts and evidences of different environmental resources like water, energy, food and climate (WEFC) and accumulated by a vast number of environmental technologies. Hence, legal ontology and ODPs of WEFC nexus might help to accumulate appropriate evidences from environmental Big Data which might be legally admissible in the legal reasoning process of WEFC nexus. From this realization, the paper presents a number of ODPs of WEFC nexus.

Acknowledgement. An especial thank to Prof. Monica Palmirani, the Director of Erasmus Mundus LAST-JD program and Associate Professor at Department of Law in University of Bologna and also supervisor of this doctoral project, for her fruitful discussion on this paper.

References

[1] Levy, E.H. An introduction to legal reasoning. Chicago: University of Chicago Press, (1948)
[2] Hannu, T.K., Johanna, S. Minna, H. Evidence and legal reasoning: on the interwinement of the probable and the reasonable. Law and Philosophy, 10(1), pp 73-107, (1991)
[3] Jaap, H. A theory of legal reasoning and a logic to match. Artificial Intelligence and Law, 4(3-4), pp 199-273, (1996)
[4] Neil, M. Legal reasoning and legal theory. Oxford: Oxford University Press, (1978)
[5] Douglas, F. Legal reasoning in environmental law: a study of structure, form and language. Massachusetts: Edward Elgar Publishing Ltd, (2013).
[6] Wynne, B. Risk and environment as legitimatory discourses of technology: reflexivity inside out? Current sociology, 50: pp 459-477, (2002).
[7] Hellegers P, Zilberman D, Steduto P, McCornick P. Interactions between water, energy, food and environment: evolving perspectives and policy issues. Water Policy 10(1): pp 1-10, (2008)
[8] Shahbaz K, Hanjra MA. Footprints of water and energy inputs in food production: global perspectives. Food Policy 34(2):pp 130-140, (2009).
[9] Berger M. Development: water gap to widen dramatically by 2030, (2009 Nov 23) <http://www.global-issues.org/news/2009/11/23/3618>. Accessed 2014 April 20.
[10] Rahman M. Legal knowledge framework for identifying water, energy, food and climate nexus, (2013) <ceur-ws.org/Vol-1105/paper3.pdf> Accessed 2014 April 18.
[11] Chandrasekaran B, Josephson J.R, and Benjamins V.R. What are Ontologies and why do we need them? IEEE Intelligent Systems, pp 20–26, (1999)
[12] Fensel D, Harmelen V.F, Horrocks I, McGuinness D. L, & Patel-Schneider P. F. OIL: an ontology infrastructure for the Semantic Web". In: *Intelligent Systems*. IEEE, 16(2): pp 38–45, (2001)
[13] Yudelson M, Gavrilova T, & Brusilovsky P. Towards User Modeling Meta-ontology. Lecture Notes in Computer Science, pp 3538-3448, (2005)
[14] R. Goebel, J. Siekmann, andW.Wahlster. Knowledge Engineering: Practice and Patterns. Lecture Notes in Computer Science, Vol. 6317, (2010)
[15] Masolo, C. editor. Social Roles and their Descriptions. In: Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004), pp 267–277, (2004)
[16] Svab-Zamazal O, Svatek V, Scharffe F. Pattern-based ontology transformation service. In: International Conference on Knowledge Engineering and Ontology Development, KEOD 2009, (2009)
[17] Corcho O, Fernandez-Lopez M, Gomez-Perez A. Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46(1), pp 41–64, (2003)
[18] Harmelen V.F, Teije T.A, Wache, H. Knowledge engineering rediscovered: towards reasoning patterns for the semantic web. In: 5th International Conference on Knowledge Capture (K-CAP 2009), pp 81–88, (2009)
[19] Hoff H. Understanding the nexus: Background paper for the Bonn2011 Nexus Conference: the water, energy and food security nexus, pp 5, Stockholm, Stockholm Environment Institute, (2011)
[20] Keulertz M. The water, food, trade, energy nexus – the epitome of the next phase of green capitalism. In Kowarsch M, editor. Water management options in a globalized world: proceedings of an international scientific workshop, pp 134-142, Munich: Munich school of philosophy, Institute for Social and Development Studies, (2011)
[21] Waughray D, (eds.) Water security: the water food energy climate nexus – the World Economic Forum Water Initiative, pp 272, Washington: Island press, (2011)
[22] [NIC, USA] National Intelligence Council, US. Global trends 2030: alternative worlds, pp 5, Washington: Office of the Director of National Intelligence, (2012)
[23] [EU] European Union. European report on development 2011/2012: towards better management of water, energy and land, (2012 May 16) < http://ec.europa.eu/europeaid/what/development-policies/research-development/erd-2011-2012_en.htm>. Accessed 2014 Feb 27.
[24] The water energy and food security resource platform. EU support for water food and energy nexus, (2013 Jan 1) <http://www.water-energy-food.org/en/news/view__1191/eu-support-for-water-food-and-energy-nexus.html>. Accessed 2014 April 02.
[25] Hellegers P, Zilberman D, Steduto P, McCornick P. Interactions between water, energy, food and environment: evolving perspectives and policy issues. Water Policy 10(1): pp 1-10, (2008)
[26] Campbell A. Food, energy, water: conflicting in securities (and the rare win-wins offered by soil stewardship. Journal of soil and water conservation 63(5): pp 149-151, (2008)
[27] Shahbaz K, Hanjra MA. Footprints of water and energy inputs in food production: global perspectives. Food Policy 34(2): pp 130-140, (2009)

[28] Berger M. Development: water gap to widen dramatically by 2030, (2009 Nov 23) <http://www.global-issues.org/news/2009/11/23/3618>. Accessed 2014 April 21.
[29] Orts E, Spigonardo J. Special report: the nexus of food, energy and water, pp 5, Pennsylvania: Institute for Global Environmental Leadership, the Wharton School, University of Pennsylvania, (2013)
[30] Partzsch L, Hughes S. Food versus fuel: governance for water rivalry. In Gottwald FT, Ingensiep HW, Meinhardt M, editors. Food ethics, pp 153-166, New York: Springer, (2010)
[31] Glassman D, Wucker M, Isaacman T, Champilou C. The water energy nexus: adding water to the energy agenda, pp 6, New York: World Policy Institute, (2011).
[32] Hoff H. Translating Nexus research into nexus policy making, (2012 May 31) < http://www.water-energy-food.org/en/news/view__630/translating_nexus_research_into_nexus_policy-making.html >. Accessed 2014 April 05.
[33] Hussey K, Pittock J. The energy water nexus: managing the links between energy and water for a sustainable future. In Ecology and Society 17(1): pp 31, (2012)
[34] UNWater. Water security and the global water agenda: a UN-Water analytical brief., pp 26, Ontario: United Nations University Press, (2013)
[35] Bizikova L, Roy D, Swanson D, Venema HD, McCandless M. The water energy food security nexus: towards a practical planning and decision support framework for landscape investment and risk management, pp 14, Manitoba: International Institute for Sustainable Development, (2013)
[36] Lindstrom A, Granit J. Weinberg J. Large scale water storage in the water, energy and food nexus: perspectives on benefits, risks and best practices, pp 19, Stockholm: SIWI paper 21 (2012)
[37] Andrews-Speed P, Bleischwitz R, Boersma T, Johnson C, Kemp G, VanDeveer SD. The global resource nexus: the struggles for land, energy, food, water and minerals, pp 55, Washington: Transatlantic Academy, (2012)
[38] UN-ESCAP. The status of the water, food, energy nexus in Asia and the pacific., pp 38, Bangkok: UN publication, (2013)
[39] Fry A. Facts and trends water, pp 2, Geneva: World Business Council for Sustainable Development, (2006)
[40] [WWAP] World Water Assessment Programme. The United Nations world water development report 3: water in a changing world, pp 80-95, Paris: UNESCO Publishing, (2009)
[41] [WBCSD] World Business Council for Sustainable Development. Water, energy and climate change: a contribution from the business community, pp 16, Geneva: WBCSD, (2009)
[42] Chen C.H, editor. Handbook of pattern recognition and computer vision (4th edition). Singapore: World Scientific Publishing Co, (2010)
[43] Ethem A. Introduction to Machine Learning - Adaptive Computation and Machine Learning, MIT Press, ISBN 0-262-01211-1, (2004)
[44] Mohri M, Rostamizadeh A, Talwalkar A. Foundations of Machine Learning, The MIT Press. ISBN 9780262018258, (2012)
[45] Ferreira C. Designing Neural Networks Using Gene Expression Programming. In Abraham A. B, Baets D.M, Köppen, and NickolayB. editors. Applied Soft Computing Technologies: The Challenge of Complexity, pp 517–536, Springer-Verlag, (2006)
[46] Yanofsky N.S, Mannucci M.A. Quantum computing for computer scientists. Cambridge: Cambridge University Press, (2008)
[47] Byrne P. The many worlds of hugh everett. In Scientific American Magazine, pp 99-105, (2007)
[48] Cleve R, Gavinsky D, and Yonge-Mallo D. Quantum algorithms for evaluating min-max trees. In Kawano Y, and Mosca M. editors. Proceedings of theory of quantum computation, communication and cryptography. V. 5106, pp 11-15, (2008)
[49] Ricks, B., Ventura, D. Training quantum neural network, (2004) < http://papers.nips.cc/paper/2363-training-a-quantum-neural-network.pdf >. Accessed 2014 August 05.
[50] Behrman, E.C., Chandrashekar, V., Wang, Z., Belur, C.K., Steck, J.E., Skinner, S.R. A quantum neural network computes entanglement, (2002) < http://arxiv.org/ftp/quant-ph/papers/0202/0202131.pdf >. Accessed 2014 August 15.

A Hierarchy of Legal Indices

Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu

School of Information Science
Japan Advanced Institute of Science and Technology
{tho.le, nguyenml, shimazu}@jaist.ac.jp

Abstract. Legal documents are frequently difficult to read due to its complicated vocabulary and structures. In this work, we target a system that support the readers and the legislators to understand and edit legal documents. By visualizing the relations among legal concepts (indices), the organization of the ideas will be represented structurally to the readers. Our system provides a hierarchy of legal indices which relates to the user query. The hierarchy is extracted from a Markov network which represents the relations among legal indices in a legal document. We report the preliminary results of constructing the hierarchy of legal indices.

Keywords: hierarchy, legal index, Markov networks

1 Introduction

Legal documents play an important role in all countries and organizations in ruling the society. Since they are often updated to keep up with the changes of our society, everyone should be aware of the newest information. Therefore, managing and understanding the contents of legal documents are necessary. This work contributes to the research field of *Legal engineering* [4] which applies knowledge and techniques in computer science to store, process and retrieve information from legal documents. In the scope of this work, legal documents are corporate documents such as constitution, laws, rules, codes, ordinances or acts, which are promulgated by the government or the administrative organizations. Legal documents are also the civil documents such as contracts, agreements or wills, which are officially composed by lawyers. They describe how things work relied upon previous documents and have interrelations to other (sections of) documents. Each of them is only a part of the legal system for a country or an organization.

For a normal citizen, reading legal documents is not an easy task due to its nature. Legal documents are frequently composed by long complicated sentences with unique vocabulary from the legal field. Sentences in legal documents often make references to other sections within a document or to other legal documents. In addition, the language of legal documents is generally very formal, and style is more important than readability. Owing to the interrelation among legal concepts, the readers may be found in difficulties when reading or retrieving information from the system of legal documents.

Even a legal document editor (e.g, legislators or lawyers), who is the expert in legal domain, also need to retrieve information from legal documents to check the compatibility and the conflict when editing or modifying the documents. However, the performance on legal information retrieval (IR) is reported unsatisfactorily [8] though IR system has reported to achieve more than 70% of precision in general text. Therefore, the legislators still have to manually retrieve the relative information when editing legal documents.

Our study motivates to support the readers and the legal editors structurally understand the general ideas of legal documents and easily navigate to related information. We target a system that is able to reply to user requirements by providing them output in form of a collection of keywords. These keywords are connected together and represented as a top-down tree view format. In other words, the output represents the content of a document (or a collection of documents) in a structural condense summary, or *"hierarchical summary."* In addition, the system should have ability to show the positions in document(s) where the content is potentially affected by users' modifications.

We separate our target into two purposes. The first purpose of the study supports the readers retrieve the relative legal information by providing them a *hierarchy of indices*. Intuitively, when a user inputs keywords, the system will provide him relative information as a hierarchy of indices which suit his wishes properly. It means that the output hierarchy is flexible relying on the users' requirements. So that, the user can view the desired information in general structured view as a hierarchy and browse down for more details in the document's sections. The second purpose of the study supports the legislators pinpointing the positions where information is potentially affected by their modified text. When a legislator inserts/deletes/modifies legal sentence(s) in a document, the system should be aware of the locations within the given document (or in a collection of documents) where information can be affected.

In the current progress, we concentrate on the first purpose which creates the hierarchy of legal indices. Specifically, this work proposes a problem of constructing a hierarchy of Japanese legal indices which relates to a user query. We consider keywords in Japanese legal documents as the legal indices and assume that keyword extracting task is ready [6]. We define the characteristics of the hierarchical structure as follows:

- The root node is the query from the users;
- Other indices in the hierarchy are the keywords from legal documents;
- The level directed from the root node includes all keywords that exist in the query;
- The closer of a level to the root, the more relative of that level's indices to the query;
- Each child node in lower level may have relations to more than one parent in higher level.

Figure 1 shows an example output of hierarchical structure. In this hierarchy, each node is a keyword and relations of a node to its children are affected by its parents.

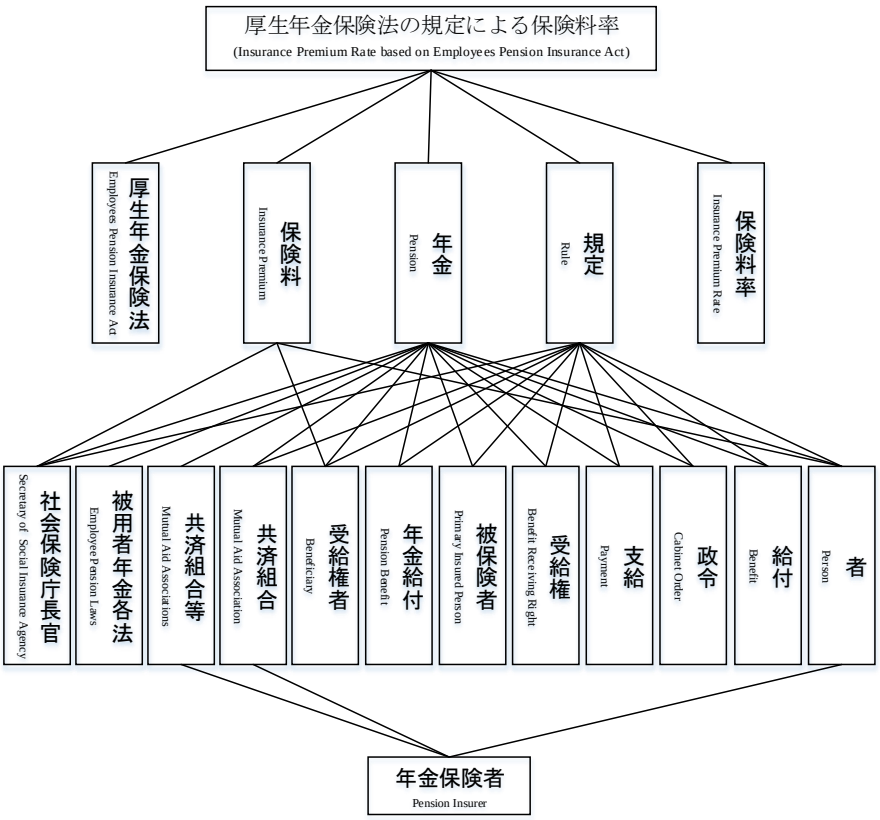


Fig. 1. An example of hierarchy of Japanese legal indices.

On the construction, we approach the hierarchy of legal indices by extracting a sub-structure from a graphical network of legal indices based on the demand of the readers. Graphical structure has been used to represent the relations among legal terms from the global view, such as semantic network [11] and context networks [12]. Our proposed hierarchy represents a local view of indices which relates to the readers' requirements. To the best of our knowledge, there is no work on constructing hierarchy of legal indices based on user query.

2 Construction of Hierarchy of Legal Indices

We approach the hierarchy by construct a graphical structure and extract sub-structure from that graph as the hierarchy. Each node in the graph is a legal keyword. Two nodes are considered to have a relation if they occur in the same

sentence. In this work, we utilize Markov networks as the graphical structure for representing the legal indices and their relations.

2.1 Markov Networks

A Markov network is a model for the joint distribution of a set of random variables. It is composed of an undirected graph $G = (V, E)$ and a set of potential functions ϕ_k. In which, V is a set of vertices such that a vertex is associated with a random variable and the graph G satisfies local Markov properties. The model for the graph has a potential function for each clique in the graph. A potential function is a non-negative real-valued function of the state of the corresponding clique.

Two important tasks of all kinds of graphical models are inference and learning. The main inference task in graphical model is to compute the conditional probability of some variables (query) given the values of some other variables (the evidence), by summing out the remaining variables. This problem is #P-completed. Thus, approximate inference techniques are required. A popular used technique is Markov chain Monte Carlo (MCMC) [2].

Learning task in graphical model includes parameter learning (or weight learning) ans structure learning. Parameter learning (or weight learning) is the task that learn the weights of features by optimizing a given objective function. The most popular objective function is the log-likelihood of the training data. In Markov networks, the log-likelihood is a convex function of the weights and learning task can be solved by convex optimization. However, this optimization requires an iterative optimization technique where log-likelihood and its gradient must be calculated at each step of the optimization. This is typically infeasible due to the requirement of computing partition function Z. In addition, an approximation may mislead the weight learning process due to reducing the expresivity of an underlying model and changing parameters [5]. Optimizing the pseudo-likelihood is an alternative efficient method that has been widely used in domains of spatial statistics, social network modeling and natural language processing.

The structures of Markov networks are usually represented as a set of features, each of which is a Boolean-valued function of a subset of the variables. The process of learning Markov network structure is to discover conditional (in)dependencies in the data such that the joint distribution can be represented more compactly. Learning an effective Markov network structure is difficulty due to the large structure space of data and the heavy computational cost of weight learning. Most approaches pose this task as a feature induction problem. There are two main approaches to automatically learn the structures of Markov networks: top-down and bottom-up approaches. Top-down approaches use greedy strategy [9] [1] to respectively add each feature to the model such that the model is improved. Bottom-up approaches [10] [7] [3] learn local models and combine them into a global model.

In this work, we apply an unsupervised structure learning approach GSSL [3] which use bottom-up strategy to construct Markov network of Japanese legal indices.

2.2 Extracting Hierarchy of Indices from the Markov network

With a query from user, we extract relative indices from the Markov network. Query Q from user is parsed to a set of terms (or indices, nodes, variables) $Q = \{q_1, q_2, ..., q_m\}$. The Markov network is represented by a feature set $\mathcal{F} = \{F_1, F_2, ..., F_n\}$, in which each feature F_i contains a set of indices. Algorithm 1 specifies how we extract a hierarchy of indices from the Markov network of legal terms. The root index is the user query. All indices in the query is the nodes of the first level. Each child level is constructed as follows:

1. Select Markov network features that cover the given query;
2. Remove the selected features the network;
3. The parent indices are those from the query;
4. The child indices are the new indices which included in the selected features;
5. Link the parent indices to child indices;
6. Make a new query with the new indices from the features;
7. Repeat this process with new query until reach desired height.

3 Experiments

The Japanese National Pension Act (JNPA) is used as the data for experiment. This data has been used for extracting Japanese keywords from legal documents [6]. There are 879 sentences in total with 208 keywords annotated manually by professionals. We consider sentences as examples and keywords as variables for training. Therefore, there are 879 examples in the training set. Each example contain 208 variables taking value from $\{0, 1\}$. The value of a variable in an example is set as 1 if the corresponding keyword appear in the corresponding sentence.

GSSL (Generate Select Structure Learning) approach [3] and Libra Toolkit[1] are employed to construct Markov network with default settings. First, 879 examples which contain binary values are The representation of Markov network follows the format in Libra toolkit.

When the Markov network of Japanese legal indices is ready. We apply Algorithm 1 to extract the hierarchy of indices with query input from the readers. Figure 1 shows the hierarchy output by our system with height of 3.

[1] http://libra.cs.uoregon.edu/

Algorithm 1 Extract Hierarchy of Legal Indices

Input: Markov network feature set $\mathcal{F} = \{F_1, F_2, ..., F_n\}$,
Query $Q = \{q_1, q_2, ..., q_m\}$, Height H
Output: A hierarchical tree R (R is root node)
Visited nodes $V \leftarrow \emptyset$
Root node$R \leftarrow Q$
Set of hierarchical sub trees $T \leftarrow \emptyset$
$T \leftarrow$ ExtractHierSubTrees(Q, $\mathcal{F}$, H)
for all tree node $t \in T$ **do**
  Sub-tree t is linked to root node R
end for

function ExtractHierSubTrees(Query Q, MN feature set $\mathcal{F}$, Height H)
  $V \leftarrow V \cup Q$
  Set of hierarchical sub trees $T \leftarrow \emptyset$
  if $H = 0$ **then**
    $T \leftarrow Q$
  else
    $Q' \leftarrow Q$
    $\mathcal{F}' \leftarrow \mathcal{F}'$
    Candidate MN features $Cans \leftarrow \emptyset$
    while $|Q'| > 0$ **do**
      $F \leftarrow \{F_i || F_i| \geq |F_j| \wedge weight(F_i) \geq weight(F_j), \forall i \neq j\}$
      $Cans \leftarrow Cans \cup F$
      $Q' \leftarrow Q' \setminus F$
      $\mathcal{F}' \leftarrow \mathcal{F}' \setminus \{F\}$
    end while
    for all candidate $c \in Cans$ **do**
      $Q' \leftarrow c \setminus (Q \cup V)$
      Parent nodes $P \leftarrow Q \cap c$
      $ST =$ ExtractHierSubTrees(Q', $\mathcal{F}'$, $H - 1$)
      for all node $p \in P$ **do**
        for all node $st \in ST$ **do**
          if $CoOccurrence(term_p, term_{st}) \geq THRESHOLD$ **then**
            Sub-tree st is linked to its parent node p
          end if
        end for
      end for
      $T \leftarrow T \cup p$
    end for
  end if
  return Set of hierarchical sub trees T ▷ $T = \{t_1, t_2, ..., t_k\}$
end function

4 Evaluation Metrics

The graphical structure is usually not easy to evaluate directly. Most previous works evaluate the structure indirectly. For example, the structure of a Markov network is evaluated by the performance of the inference task.

We plan to evaluate the effectiveness of the hierarchy of legal indices by measuring how the user gain from using our hierarchy compare with usual information retrieval. We assign the same set of queries to two groups of readers and measure the time they need to obtain the correct information.

5 Conclusion

To support the readers get the general information, we introduce a hierarchical structure that shows the relations among legal indices to show a local view to the readers based on their queries. A Markov network is used to represent the relations among legal indices. The hierarchy is constructed by extracting substructure from the Markov network based on a query given by the readers. In the next step, we are going to conduct subjective evaluations to evaluate the resulted hierarchies.

References

1. Jesse Davis and Pedro Domingos, "Bottom-Up Learning of Markov Network Structure," in Proc. ICML'10, 2010.
2. W.R. Gilks and DJ Spiegelhalter, "Markov chain Monte Carlo in practice," Chapman & Hall/CRC, 1996.
3. Jan Van Haaren and Jesse Davis, "Markov network structure learning: A randomized feature generation approach," in Proc. AAAI'12, pp. 1148-1154, 2012.
4. T. Katayama, "Legal engineering - An engineering approach to laws in e-society age," in Proc. 1st Int. Workshop JURISIN07, Japan, 2007.
5. Alex Kulesza and Fernando Pereira, "Structured Learning with Approximate Inference," in Proc. NIPS'07, pp. 785–792, 2008.
6. Tho Thi Ngoc Le, Minh Le Nguyen, Akira Shimazu, "Unsupervised Keyword Extraction for Japanese Legal Documents," in Proc. JURIX'13, pp. 97–106, 2013.
7. Daniel Lowd and Jesse Davis, "Learning Markov Network Structure with Decision Trees," in Proc. ICDM'10, pp. 334–343, 2010.
8. K. Tamsin Maxwell and Burkhard Schafer, "Concept and Context in Legal Information Retrieval, " in Proc. JURIX'08, pp. 63–72, 2008.
9. S. Della Pietra and V. Della Pietra and J. Lafferty, "Inducing features of random fields," IEEE Trans. Pattern Anal. Mach. Intell. 19(4), pp. 380–393, 1997.
10. Pradeep Ravikumar and Martin J. Wainwright and John D. Lafferty, "High-Dimensional Ising Model Selection using L1-Regularized Logistic Regression," Annals of Statistics, vol 19, pp. 1287–1319, 2010.
11. Radboud Winkels and Alexander Boer and Emile de Maat and Tom van Engers and Matthijs Breebaart and Henri Melger, "Constructing a Semantic Network for Legal Content," in Proc. ICAIL '05, pp. 125–132, 2005.
12. Radboud Winkels and Alexander Boer and Ivan Plantevin, "Creating Context Networks in Dutch Legislation," in Proc. JURIX'13, pp. 155-164, 2013.

Generating Legal Reasoning Structure by Answer Set Programming

Duangtida Athakravi[1], Ken Satoh[2], Krysia Broda[1], and Alessandra Russo[1]

[1] Imperial College London {da407,kb,ar3}@doc.ic.ac.uk
[2] National Institute of Informatics and Sokendai ksatoh@nii.ac.jp

Abstract. In legal reasoning, different assumptions are often considered when reaching a final verdict and judgement outcomes strictly depend on these assumptions. In this paper, we propose an approach for generating a declarative model of judgements from past legal cases, that expresses a legal reasoning structure in terms of principle rules and exceptions. Using a logic-based reasoning technique, we are able to identify from given past cases different underlying defaults (legal assumptions) and compute judgements that (i) cover all possible cases (including past cases) within a given set of relevant factors, and (ii) can make deterministic predictions on final verdicts for unseen cases. The extracted declarative model of judgements can then be used to make automated inference of future judgements, and generate explanations of legal decisions. The rules generated by our approach can also be automatically translated into a representation compatible with the legal reasoning system PROLEG, so making our method a useful computational mechanism for generating PROLEG models from past cases.

1 Introduction

In legal reasoning, especially in continental laws, we use written rules to make a judgement in litigation. In these written rules, principle rules and exceptions are mentioned. It is related to proof of persuasion where conditions of principle rules must be proved by the side who claims holding the conclusion of principle rules, whereas exceptions must be proved by the side who denies the conclusion. Moreover, some rules are refined by the highest court in a country (the supreme court in Japan, for example) by adding some exceptions, and exceptions of exceptions, if the current principle rules or exceptions do not capture the conclusion of the current litigation from other viewpoints such as, for instance, social welfare.

In this paper, we consider the problem of learning a set of case-rules from previous cases judged by the court. Suppose that the following cases are found in the conclusion of "*depriving the other party of what he(she) is entitled to expect under the contract*" in commercial litigation.

Case 1: The plaintiff (the buyer) showed that the goods were delivered on time but he failed to prove that there was a damage of goods. In this case, the judge decided that the seller did not deprive the buyer of what he expects.

Case 2: The plaintiff showed that the goods were delivered on time but the goods were damaged. Then, the defendant failed to prove that the trouble could be repaired. In this case, the judge decided that the buyer was deprived of what he expects.

Case 3: The plaintiff showed that the goods were delivered on time but the goods were damaged. Then, the defendant showed that the trouble could be repaired and the buyer fixed an additional period of time for repair and the repair was completed in the additional period. In this case, the judge decided that the seller did not deprive the buyer of what he expects.

Case 4: The plaintiff showed that the goods were not delivered on time. Then, the defendant showed that the buyer fixed an additional period of time but failed to prove that the goods were delivered in the period. In this case, the judge decided that the buyer was deprived of what he expects.

Case 5: The plaintiff showed that the goods were not delivered on time. Then, the defendant (the seller) showed that the buyer fixed an additional period of time and the goods were delivered in the period. In this case, the judge decided that the seller did not deprive the buyer of what he expects.

We would like to decide whether the following case satisfies the conclusion "depriving the other party of what he is entitled to expect under the contract":

New Case: *The plaintiff showed that the goods were delivered on time but the goods were damaged. The defendant showed that the trouble could be repaired and showed that the buyer fixed an additional period of time for repair, but failed to prove that the repair was not completed in the additional period.*

We can formalise all the factors mentioned in cases 1-5 as described below.

dot	The goods are delivered on time
fad	The buyer fixes an additional period for delivering the goods
dia	The goods are delivered in the above additional period
ooo	The goods are damaged
rpl	The goods are repairable
far	The buyer fixes an additional period for repair
ria	The goods are repaired in the above additional period

Past cases can then be expressed as pairs where the first argument denotes the factors mentioned the case and the second argument the positive ($+dwe$) or negative ($-dwe$) conclusion "depriving the other party of what he is entitled to expect under the contract". We would like to predict what the conclusion would be for the new case, given the past cases:

Case 1	$\langle\{dot\}, -dwe\rangle$
Case 2	$\langle\{dot, ooo\}, +dwe\rangle$
Case 3	$\langle\{dot, ooo, rpl, far, ria\}, -dwe\rangle$
Case 4	$\langle\{\neg dot, fad\}, +dwe\rangle$
Case 5	$\langle\{\neg dot, fad, dia\}, -dwe\rangle$
New Case	$\langle\{dot, ooo, rpl, far\}$,**??**$\rangle$

In order to make a judgement for a new case, the history of judgement revisions occurred in past cases needs to be taken into account. Let us assume in our example that if no factor is proved, the conclusion dwe will not be approved by judges (that is $-dwe$ is a conclusion in an empty case - a case with an empty

set of factors). Then, Case 1 has the same conclusion as the initial empty case, so the factor *dot* is irrelevant to the judges. Then Case 2 comes and the conclusion is reversed (judgement revision is made). We conclude that a combination of *dot* and *ooo* does affect the conclusion and therefore we understand that the simultaneous existence of *dot* and *ooo* is an exceptional situation. Then Case 3 comes and the conclusion is reversed again into the original judgement. Then, we could say that the combination of rpl, far, ria is an exception to the exceptional situation in which *dot* and *ooo* exist. For Case 4, the conclusion differs from the initial empty case so the combination of $\neg dot$ and fad represents an exceptional situation. Then Case 5 comes and the conclusion is reversed again into the original judgement. Therefore, *dia* constitutes an exception for exceptional situation in which $\neg dot$ and fad exist.

Finally, for the new case, given the factors $\{dot, ooo, rpl, far\}$, we should conclude $+dwe$, since *dot* and *ooo* hold in this case (and Case 2 was considered to be an exceptional situation with these factors), and it cannot be considered to be an exception for this exception since the only such exception known is the combination of rpl, far, ria, and in the new case ria does not hold.

In this paper, we aim to formalise the above reasoning and extract a set of rules that capture the past judgements and allow prediction of judgements for new cases. We view the cases and their judgements as outcomes of reasoning using an argumentation model [4], where the factors are arguments presented to the judge and the judgement is the conclusion after applying the attacks between the factors. Hence, attacks can be inferred from past cases that share some common factors but have opposing judgements, and consequently arguments are inferred from the factors that are uncommon between the two cases. Then, the purpose of relevant attacks is to identify only the necessary factors that affect the judgement of a new case. Consider the above cases. Case 1 does not contain any relevant information for deciding the outcome of future cases. Similarly, suppose Case 6 exists where the buyer was deprived of what he expects as the goods are delivered on time, but are damaged and repairable $\langle\{dot, ooo, rpl\}, +dwe\rangle$. This Case 6 will also be deemed irrelevant, due to the existence of Case 2 which tells us that $\{dot, ooo\}$ are already sufficient for overturning the judgement. Thus, by extracting information about the relevant attacks and their arguments we can define rules reflecting the reasoning applied by the judge. These rules are generated using the ASP solver Clingo [5] and an appropriate meta-level representation. The outcome is a logic program that is equivalent to the PROLOG representation of PROLEG [11], making the generated program representable in PROLEG as well.

The paper is structured as follows. In Section 2 we provide the formal definition of our computational model, introduce the notions of relevant attack, prediction of judgement and define the type of legal reasoning structures that we are able to extract from past cases. In Section 3 we show how the approach is implemented in Answer Set Programming (ASP), illustrate its execution in Section 4 through the legal reasoning example described above and discuss in

Section 5 how the outcome of our approach can be translated back into PROLEG rules. Sections 6 and 7 discuss, respectively, related work and conclusion.

2 Formalisation

In this section we give the formal definition of our computational model. We first define the formal representation of past cases and related judgements. We then define, using this representation, the concepts of relevant attack and prediction of judgements, and formalise the type of legal rules our approach is able to compute.

Definition 1 (Casebase). *Let F be a set of elements called* factors*. A* case *is a subset of F. A* case with judgement *is a pair $cj = \langle c, j\rangle$, where c is a case and $j \in \{+, -\}$. The set c is also referred to as the set of factors included in a case with judgement. Given a case with judgement cj, $case(cj)$ denotes the set of factors included in cj and $judgement(cj) = j$ denotes the judgement decision taken in the case. A* casebase*, denoted with CB, is a set of cases with judgements, namely a subset of $F \times \{+, -\}$.*

Given a casebase CB we impose the restriction that for every $cj_1, cj_2 \in CB$, if $case(cj_1) = case(cj_2)$ then $judgement(cj_1) = judgement(cj_2)$. This avoids inconsistent casebases. We also assume that all casebases contain an element $\langle \emptyset, j_0\rangle$ representing the empty case and *default judgement*, the assumed judgement in the absence of any factor, of the casebase.

Definition 2 (Raw Attack). *Let CB be a casebase. A* raw attack *relation is a set $RA \subseteq CB \times CB$ defined as follows: $\langle cj_1, cj_2\rangle \in RA$ if $case(cj_1) \supset case(cj_2)$ and $judgement(cj_1) \neq judgement(cj_2)$. For every pair $\langle cj_1, cj_2\rangle \in RA$, we say cj_1* raw attacks *cj_2 and we write $cj_1 \rightarrow_r cj_2$.*

Example 1. Let us consider the set of factors F given by $\{a, b, c, d, e, f\}$ and a casebase CB given by the following named cases with judgements:

$\{\ c0 : \langle\{\}, -\rangle,$
$\ \ c1 : \langle\{A\}, +\rangle,$
$\ \ c2 : \langle\{A, B\}, +\rangle,$
$\ \ c3 : \langle\{A, B, C\}, -\rangle,$
$\ \ c4 : \langle\{A, B, C, D\}, -\rangle,$
$\ \ c5 : \langle\{A, B, C, D, E\}, +\rangle,$
$\ \ c6 : \langle\{A, B, C, D, E, F\}, -\rangle,$
$\ \ c7 : \langle\{D\}, +\rangle\}.$

Then, the raw attack relation over CB is given by the following set $\{c1 \rightarrow_r c0$, $c2 \rightarrow_r c0$, $c5 \rightarrow_r c0$, $c7 \rightarrow_r c0$, $c3 \rightarrow_r c1$, $c4 \rightarrow_r c1$, $c6 \rightarrow_r c1$, $c3 \rightarrow_r c2$, $c4 \rightarrow_r c2$, $c6 \rightarrow_r c2$, $c5 \rightarrow_r c3$, $c5 \rightarrow_r c4$, $c6 \rightarrow_r c5$, $c4 \rightarrow_r c7$, $c6 \rightarrow_r c7\}$.

Definition 3 (Relevant Attack). *Let CB be a casebase. A* relevant attack *relation is a set $A \subseteq RA$ defined as follows: $\langle cj_1, cj_2\rangle \in A$, and written as $cj_1 \rightarrow cj_2$, if*

(i) there is no $cj_3 \rightarrow_r cj_2$ in RA such that

– $case(cj_1) \supset case(cj_3) \supset case(cj_2)$,

(ii) *there is no* $cj_1 \rightarrow_r cj_4 \in RA$ *such that*

– $case(cj_1) \supset case(cj_2) \supset case(cj_4)$ *and,*

– *there is no* $cj_5 \rightarrow_r cj_4 \in RA$ *such that*

• $case(cj_2) \supset case(cj_5) \supset case(cj_4)$

A relevant attack $cj_1 \rightarrow cj_2$ is between a case cj_1 that overturns the judgement of another case cj_2, with $case(cj_1) \supset case(cj_2)$, and such that i) there isn't another attack against cj_2 whose factors are a subset of $case(cj_1)$, implying that $case(cj_1)$ contains the relevant factors for overturning the judgement of cj_2; and ii) for any other case that cj_1 raw attacks there is an intermediate judgement revision whose factors are included in $case(cj_2)$. The second condition implies the existence of a chain of judgement revisions from the default case to a given attack case. In summary, for a case to be relevant it must either be the default case, or it must be involved in a relevant attack against other relevant cases.

Example 2. Let us consider the set RA of raw attacks defined in Example 1, The set of relevant attacks is given by the following subset:

$\{c1 \rightarrow c0,\ c7 \rightarrow c0,\ c3 \rightarrow c1,\ c5 \rightarrow c3,\ c6 \rightarrow c5,\ c4 \rightarrow c7\}$.

So not all raw attacks are relevant attacks. Consider for instance the pair $\langle cj_2, cj_0 \rangle$. This is a raw attack but it is not a relevant attack because the first condition in Definition 3 is not satisfied, as there exists another case cj_1 which is the raw attack $\langle cj_1, cj_0 \rangle$ where $case(cj_1)$ is a subset of $case(cj_2)$. For each relevant attack the set of factors, called *argument*, responsible for overturning the judgement can be deduced by comparing the factors in the two cases.

Definition 4 (Argument). *Let CB be a casebase and let A be the set of relevant attacks relation with respect to CB. For each pair $\langle cj_1, cj_2 \rangle \in A$, the set of factors $\alpha(cj_1, cj_2)$, representing the attack from cj_1 to cj_2, is given by $\alpha(cj_1, cj_2) = case(cj_1) - case(cj_2)$. Given a relevant attack $cj_1 \rightarrow cj_2$, for any case c such that $c \supset case(cj_2)$, the attack is said to be* irrelevant for c *if the argument $\alpha(cj_1, cj_2) \not\subset c$.*

By considering relevant attacks and arguments of past cases, we can infer the judgement of future cases.

Definition 5 (Predicted judgement). *Let CB be a casebase and c be an unseen case. Its predicted judgement, denoted with $pj(c)$, is equal to the default judgement j_0 unless:*

– *There exists a sequence of cases with judgements in CB such that $cj_n \rightarrow \ldots \rightarrow \langle \emptyset, j_0 \rangle$, where $n \geq 1$ and,*

– $judgement(cj_n) \neq j_0$ *and,*

– $case(cj_n) \subseteq c$ *and,*

– *any other relevant attack $\langle cj_m, cj_n \rangle \in A$ is irrelevant for c.*

If such cj_n exists then $pj(c) = judgement(cj_n)$

The aim of this work is to generate a theory T from a given casebase CB and default judgement j_0 such that, given a new case c it is possible to predict the judgement for c.

3 Generating Case-rules by ASP

In this section we describe how we can reason about past cases to generate the case-rules, namely legal reasoning structures expressed in terms of principle rules and exceptions. This is done by using Answer Set Programming (ASP) and the Clingo 3 ASP solver [5]. We begin by describing how relevant attacks can be inferred from examples of past cases with judgements. Then we describe how, using the relevant attacks, case-rules can be generated in the form of a meta-level representation that can be automatically translated into PROLEG.

3.1 Extracting Relevant Attacks and Arguments from a Casebase

Each casebase can be seen as a definite clause.

Definition 6 (Rule representation of a case with judgement). *Let CB be a casebase. Each $cj \in CB$ can be expressed as a definite clause $r(cj)$ of the form:*

$$judgement(cj) \leftarrow f_1 \wedge \ldots \wedge f_n.$$

where $f_i \in case(cj)$, for $1 \leq i \leq n$. We refer to such clause as a rule.

Inferring relevant attacks and arguments means reasoning about the structure of the set of rules that express a given casebase. To do so, we make use of a meta-level representation of our cases with judgement, that describes the syntactic structure of each case, in terms of the literals (i.e. factors) that appear in each specific case.

Definition 7 (Casebase meta-level representation). *Let CB be a casebase, its meta-level representation $meta(CB)$ is defined as:*

$$meta(CB) = \bigcup_{cj \in CB} \mu(r(cj)) \cup \tau(CB) \cup \delta(CB)$$

where $\tau(CB) = \{factor(f_i) | f_i \in F\}$, $\delta(CB) = \{default\_head(judgement(cj_0)), default\_id(id(r(cj_0)))\}$ defining the meta-information about the default case, and the function μ is defined as follows:

$$\mu(r(cj)) = \begin{cases} cb\_id(id(r(cj))). \\ is\_rule(id(r(cj)), judgement(cj)). \\ in\_rule(id(r(cj)), judgement(cj), f_i) \qquad for\ each\ f_i \in case(cj) \end{cases}$$

This meta-level representation can be used to express the notion of raw attack ($\rightarrow_r$) given in Definition 2. The second and third rules below capture the condition $case(cj_1) \supset case(cj_2)$ in Definition 2, where ID_1 and ID_2 are the IDs of cases with judgement cj_1 and cj_2 respectively; the condition $H_1 \neq H_2$ in the first rule below captures instead the condition $judgement(cj_1) \neq judgement(cj_2)$ of Definition 2.

$$\begin{aligned} raw\_attack(ID_1, ID_2) &\leftarrow factor\_subset(ID_2, ID_1), is\_rule(ID_1, H_1), \\ &\quad is\_rule(ID_2, H_2), H_1 \neq H_2. \\ factor\_subset(ID_1, ID_2) &\leftarrow cb\_id(ID_1), cb\_id(ID_2), \\ &\quad not\ not\_factor\_subset(ID_1, ID_2). \\ not\_factor\_subset(ID_1, ID_2) &\leftarrow cb\_id(ID_1), cb\_id(ID_2), factor(B), \\ &\quad in\_rule(ID_1, H_1, B), is\_rule(ID_2, H_2), \\ &\quad not\ in\_rule(ID_2, H_2, B). \end{aligned}$$

The computation of the relevant attack relation (called *attack* in the program) uses the ASP's choice operator (see first rule below) to select from the inferred raw attacks those relations that satisfy the constraints (i) and (ii) given in Definition 3. These constraints are captured by the second and third rules given below.

$$0\ \{attack(ID_1, ID_2)\}\ 1 \leftarrow raw\_attack(ID_1, ID_2).$$

$$\begin{aligned}\bot \leftarrow\ & attack(ID_1, ID_2), raw\_attack(ID_3, ID_2),\\ & factor\_subset(ID_2, ID_3), factor\_subset(ID_3, ID_1).\end{aligned}$$

$$\begin{aligned}\bot \leftarrow\ & attack(ID_1, ID_2), raw\_attack(ID_1, ID_4),\\ & factor\_subset(ID_4, ID_2), not\ intermediate\_attack(ID_2, ID_4).\end{aligned}$$

$$\begin{aligned}intermediate\_attack(ID_2, ID_4) \leftarrow\ & factor\_subset(ID_4, ID_5),\\ & factor\_subset(ID_5, ID_2),\\ & raw\_attack(ID_5, ID_4).\end{aligned}$$

To generate all relevant attacks in a given casebase, we use the following Clingo optimisation expression, which guarantees the maximum number of instances of the *attack* relation to be computed in a given solution.

$$\#maximise\{attack(ID_1, ID_2)\}.$$

Finally, using the inference of relevant attack instances and the meta-level representation of given cases, we can compute *arguments* of relevant attacks:

$$\begin{aligned}argument(ID_1, ID_2, Arg) \leftarrow\ & attack(ID_1, ID_2), in\_rule(ID_1, H_1, Arg),\\ & is\_rule(ID_2, H_2), not\ in\_rule(ID_2, H_2, Arg).\end{aligned}$$

3.2 Generating Meta-level Information of Case-rules

The above set of rules helps extracting, from a given set of cases with judgements, information about relevant attacks and their arguments. But to be able to perform prediction on unseen cases, our approach has to learn the set of legal rules underpinning the casebase that reflect the underlying legal reasoning applied by the judges throughout the past judged cases. As mentioned in Section 1, the legal reasoning structure is normally composed of exceptions to a given default assumption (e.g. default case) and exceptions of exceptions, i.e. when the current principle rules (or exceptions) do not capture the conclusion of a current litigation).

Using an argumentation reasoning model, we learn abnormality rules that capture arguments of relevant attacks inferred from a given casebase. These rules are of the form:

$$ab_m \leftarrow f_{m1}, \ldots, f_{mk_m}, not\ \ ab_{m+1}, \ldots, not\ \ ab_p$$
$$ab_q\ \leftarrow f_{q1}, \ldots, f_{qk_q}$$

where $f_{m1}, \ldots, f_{mk_m}$ are factors involved in the argument that makes ab_m an exception to either a default principle or another exception. The conditions $not\ \ ab_{m+1}, \ldots, not\ \ ab_p$ express the fact that ab_m has itself counter arguments, one for each abnormality ab_j, for $m+1 \leq j \leq p$. The rule is read as "exception ab_m is true provided that factors $f_{m1}, \ldots, f_{mk_m}$ can be demonstrated to hold

and each counter argument $not\ \ ab_j$ cannot be proved to hold". Counter arguments ab_j are in turn learned as rule with head predicate ab_j. Exceptions that have no counter arguments will instead have no negated abnormalities in their conditions. This is the case for the second type of rule given above, that represents the case of an exception ab_q given by an argument $f_{q1}, \ldots, f_{qk_q}$ without any counter argument. Our computational approach learns the above type of legal rules (or case-rules) in terms of their equivalent meta-level representation. Automatic rewriting of our meta-level learned theory into rules of the types given above provide a logic program that is equivalent to the PROLOG representation of PROLEG [11] (see Section 5).

To learn a legal reasoning structure each relevant attack inferred from the past cases has to be linked to a unique abnormality name. This is computed using the ASP choice rule given below, which associates the attack identifier A_ID of a given attack $a(ID_1, ID_2)$ with an abnormality name Ab. An integrity constraint guarantees that attack identifiers are unique.

$$1\ \{id\_attack\_link(A\_ID, a(ID_1, ID_2)) : abnormal(A\_ID, Ab)\}\ 1 \leftarrow attack(ID_1, ID_2).$$
$$\bot \leftarrow id\_attack\_link(A\_ID_1, Atk), id\_attack\_link(A\_ID_2, Atk), A\_ID_1 \neq A\_ID_2.$$

Abnormality names are defined by the basic types $abnormal$ and $negated\_abnormal$ as illustrated below:

$gen\_id(r0).$
$gen\_id(r1).\ abnormal(r1, ab1).\ negated\_abnormal(r1, not\_ab1).$
$gen\_id(r2).\ abnormal(r2, ab2).\ negated\_abnormal(r2, not\_ab2).$
$gen\_id(r3).\ abnormal(r3, ab3).\ negated\_abnormal(r3, not\_ab3).$
$gen\_id(r4).\ abnormal(r4, ab4).\ negated\_abnormal(r4, not\_ab4).$
$gen\_id(r5).\ abnormal(r5, ab5).\ negated\_abnormal(r5, not\_ab5).$

where $r0$ is assumed to be a reserved identifier for the default judgement.

Once appropriate links between relevant attacks and abnormal identifiers are learned, through the choice rule given above, the meta-level representation of the rules defining these abnormalities can be inferred using the following two sets of rules. The first rule expresses the fact that the default judgement is the judgement of an empty factor set, thus it can only be applied on future cases if all attacks against cannot be proved to hold. The second rule captures the meta-level representation of an principle rule of the form $j_0 \leftarrow not\ ab_1, \ldots, not\ ab_n$, defining the absence of exceptions to the default judgement.

$$is\_rule(r0, Def) \leftarrow default\_head(Def).$$
$$in\_rule(r0, Def, Neg\_Ab) \leftarrow default\_head(Def), default\_id(ID2), id\_attack\_link(A\_ID, a(ID_1, ID_2)), negated\_abnormal(A\_ID, Neg\_Ab).$$

For each learned link between relevant attack and abnormality names, the following set of rules allows the inference of the meta-level representation of rules of the form $ab_m \leftarrow f_{m1}, \ldots, f_{mk_m}, not\ \ ab_{m+1}, \ldots, not\ \ ab_p$ and $ab_q\ \leftarrow f_{q1}, \ldots, f_{qk_q}$.

$$
\begin{aligned}
&is\_rule(A\_ID, Ab) && \leftarrow id\_attack\_link(A\_ID, Atk), abnormal(A\_ID, Ab).\\
&in\_rule(A\_ID, H, Arg) && \leftarrow is\_rule(A\_ID, H),\\
&&& id\_attack\_link(A\_ID, a(ID_1, ID_2)),\\
&&& argument(ID1, ID2, Arg).\\
&in\_rule(A\_ID_1, H, Neg\_Ab) && \leftarrow is\_rule(A\_ID_1, H),\\
&&& id\_attack\_link(A\_ID_1, a(ID_2, ID_1)),\\
&&& id\_attack\_link(A\_ID_2, a(ID_3, ID_2)),\\
&&& negated\_abnormal(A\_ID_2, Neg\_Ab).
\end{aligned}
$$

The first two rules allow the inference of the definition of predicate head abnormality name Ab that corresponds to the correct linked attack, for which the appropriate factors involved in the argument of this attack are inferred to be conditions in the body of such abnormality rule. The third rule instead captures the abnormality identifiers of subsequent attacks that invalidate this current exception Ab. The steps of our approach are shown in Figure 1, and the form of the output program corresponding to the example discussed in Section 1 is given in the next section.

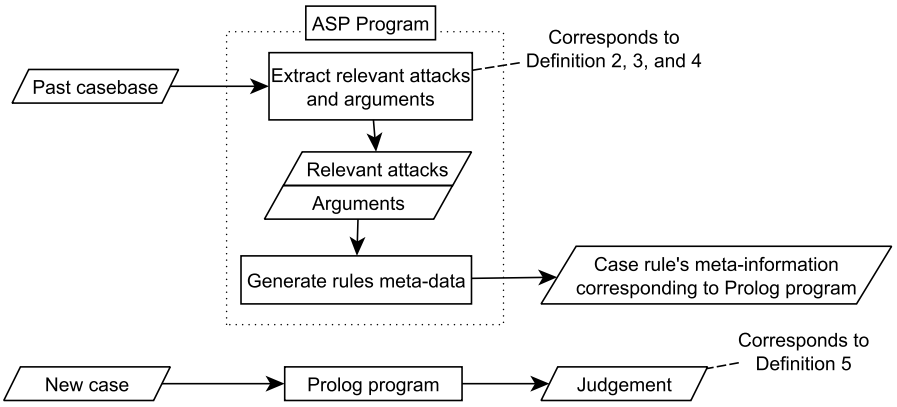


Fig. 1. Work flow for generating case rules by ASP

The output meta-representation can be transformed into a set of rules T by matching the id ID in the inferred instances $is\_rule(ID, H)$ and $in\_rule(ID, H, B)$, where H and B indicate the head and body of the rule respectively. Informally, the following set of instances

$$
\begin{aligned}
&is\_rule(id, h).\\
&in\_rule(id, h, b_1).\\
&\ldots\\
&in\_rule(id, h, b_n).
\end{aligned}
$$

will be translated into the rule $h \leftarrow b_1, \ldots, b_n$.

4 Learning Example in Legal Reasoning

In this section we illustrate our approach by showing in detail how the various steps are applied to the past cases with judgements described in Section 1.

Firstly, all the given past cases are expressed, using our meta-level representation, by the following set of facts:

```
cb_id(c0).          factor(ria).                in_rule(c3,neg_dwe,ooo).
cb_id(c1).          factor(neg_dot).            in_rule(c3,neg_dwe,rpl).
cb_id(c2).          default_head(neg_dwe).      in_rule(c3,neg_dwe,far).
cb_id(c3).          default_id(c0).             in_rule(c3,neg_dwe,ria).
cb_id(c4).          is_rule(c0,neg_dwe).        is_rule(c4,dwe).
cb_id(c5).          is_rule(c1,neg_dwe).        in_rule(c4,dwe,neg_dot).
factor(dot).        in_rule(c1,neg_dwe,dot).    in_rule(c4,dwe,fad).
factor(fad).        is_rule(c2,dwe).            is_rule(c5,neg_dwe).
factor(dia).        in_rule(c2,dwe,dot).        in_rule(c5,neg_dwe,neg_dot).
factor(ooo).        in_rule(c2,dwe,ooo).        in_rule(c5,neg_dwe,fad).
factor(rpl).        is_rule(c3,neg_dwe).        in_rule(c5,neg_dwe,dia).
factor(far).        in_rule(c3,neg_dwe,dot).
```

We were able to extract the following raw attacks ($raw\_attack/2$), relevant attacks $attack/2$, and arguments ($argument/3$):

```
raw_attack(c2,c0).      attack(c4,c0).          argument(c4,c0,neg_dot).
raw_attack(c4,c0).      attack(c3,c2).          argument(c3,c2,ria).
raw_attack(c2,c1).      attack(c5,c4).          argument(c3,c2,far).
raw_attack(c3,c2).      argument(c2,c0,ooo).    argument(c3,c2,rpl).
raw_attack(c5,c4).      argument(c2,c0,dot).    argument(c5,c4,dia).
attack(c2,c0).          argument(c4,c0,fad).
```

Using our ASP program, we generate then answer set solution containing the following meta-level representation of the case-rules:

```
is_rule(r0,neg_dwe).            in_rule(r5,ab1,fad).
in_rule(r0,neg_dwe,not_ab0).    in_rule(r5,ab1,not_ab3).
in_rule(r0,neg_dwe,not_ab1).    is_rule(r2,ab2).
is_rule(r4,ab0).                in_rule(r2,ab2,rpl).
in_rule(r4,ab0,dot).            in_rule(r2,ab2,far).
in_rule(r4,ab0,ooo).            in_rule(r2,ab2,ria).
in_rule(r4,ab0,not_ab2).        is_rule(r3,ab3).
is_rule(r5,ab1).                in_rule(r3,ab3,dia).
in_rule(r5,ab1,neg_dot).
```

This corresponds to the program:

```
neg_dwe :- not ab0, not ab1.
ab0 :- dot, ooo, not ab2.
ab1 :- fad, neg_dot, not ab3.
ab2 :- far, ria, rpl.
ab3 :- dia.
```

4.1 Correctness of the Generated Program

Here we reason that the generated program behaves as intended according to Definition 5.

First we note that as the program is stratified and each abnormality has only one definition, the generated program will have one answer set. Also, from Definition 3 and 5 we can reason that: the sequence of relevant attack cases with judgements $cj_n \rightarrow \ldots \rightarrow cj_0$, where cj_0 is the default casebase, $\langle \emptyset, j_0 \rangle$ can be represented by the sequence of their arguments $\alpha(cj_n, cj_{n-1}), \ldots, \alpha(cj_1, cj_0)$. The conditions on the sequence implies that:

- Every argument $\alpha(cj_i, cj_j)$ in this sequence is a subset of the new case $\alpha(cj_i, cj_j) \subset c$.
- Any argument representing an attack on cj_n is irrelevant for c.
- This sequence must be of odd length, since each attack overturns the judgement of the attacked case.

We outline a proof of the following property:

"The program does not derive the default judgement if, and only if, there exists a sequence of arguments as described above."

First, suppose the default judgement is not derived. By definition at least one relevant attack to the default case must have succeeded, corresponding to one abnormality in the body for the default judgement rule. Either that abnormality does not contain further abnormalities (as they only occur negatively as body literals), in which case the length of the sequence is one, or if they do contain further abnormalities then none of them hold. This may be because the arguments are irrelevant for c or there must be a successful counter attack. Similar reasoning can be applied to a finite sequence of pairs of subsequent abnormalities.

On the other hand, if there exists a sequence with the required properties then the default judgement must not be derived.

Suppose for contradiction that such a sequence exists, where the default judgement is derived. If the length of the sequence is one then there is a direct contradiction. For longer sequences, if any sequence of odd length n derives the default judgement then the the sequence of length $n + 2$ should also derive the default judgement. As sequence of length one gives a contradiction, all odd sequences will result in a contradiction.

5 Translating to PROLEG

PROLEG is used to represent rules and exceptions [11]. It would therefore be interesting to investigate a relationship between the generated program that our approach is able to compute and PROLEG.

We define a PROLEG program as follows. A rule is a horn clause of the form $H \Leftarrow B_1, ..., B_n$[3] where $H, B_1, ..., B_n$ are atoms. We denote the head of the rule R as $head(R)$. An exception is an expression of the form $exception(H, E)$ where H, E are atoms. A PROLEG program P is a pair $\langle \mathcal{H}, \mathcal{E} \rangle$ where $\mathcal{H}$ is a set of rules and $\mathcal{E}$ is a set of exceptions.

We can show that an automated translation from our generated program into PROLEG is possible using the following equivalent translation between PROLOG and PROLEG [12]. Let P be the following PROLOG program with

[3] B_i part can be empty. In this case, we write it as $H \Leftarrow$.

rules generally of the form:
$C : -B_1, \ldots B_n,$ `not` $E_1, \ldots,$ `not` E_m where the B_i or E_j part of the rule may be empty. Then, a translation of a PROLOG program P of the above form, $tr_o(P)$ into a PROLEG program $\langle \mathcal{H}_o, \mathcal{E}_o \rangle$ is as follows.

- A rule of the form C: − `not` $E_1, \ldots,$ `not` E_m, is represented by the PROLEG rules:
  $C \Leftarrow .$
  $exception(C, E_1).$
  $\vdots$
  $exception(C, E_m).$
- Rules of the form C: $-B_1, \ldots, B_n$, are represented by the PROLEG rule:
  $C \Leftarrow B_1, \ldots, B_n.$
- Rules of the form C: $-B_1, \ldots, B_n,$ `not` $E_1, \ldots,$ `not` E_m, are represented by the PROLEG rules:
  $C \Leftarrow B_1, \ldots, B_n.$
  $exception(C, E_1).$
  $\vdots$
  $exception(C, E_m).$

Then for our example of legal reasoning in Section 4, we can get the following PROLEG program.

```
neg_dwe<=.
exception(neg_dwe, ab0).
exception(neg_dwe, ab1).
ab0 <= dot, ooo.
exception(ab0, ab2).
ab1 <= fad,neg_dot.
exception(ab1, ab3).
ab2 <= far, ria, rpl.
ab3 <= dia.
```

Therefore, this work can be regarded as generating PROLEG program as well.

6 Related Work

In this work we have presented how meta-level reasoning can be used for extracting information from past legal cases for generating case rules for deciding the judgement of future cases. We have used notions of argumentation and ASP for computation, and while there have been many recent works [13] in representing argumentation framework and computing argumentation extensions using ASP, we are more concerned with the extraction of information about the arguments and attacks from examples rather than computing extensions of a given framework. This is in contrast to work such as [14] where the information extraction is from legal documents.

Similar to legislators using past legal cases for creating or revising legislations, past work in legal reasoning has explored how formal reasoning can be applied for reasoning from past cases [9, 10]. The system HYPO [1] also analyses factors in the form of dimensions, where a dimension is a structure containing a factor and the party it favours, to suggest the arguments and counter examples that plaintiff and defendant may use to further their objectives. It sorts case relevance using a claim lattice, a directed acyclic graph of dimensions and cases relevant to the set of dimensions. This differs from our approach where we sort cases using the relevant attacks, arranging them according to their ability to overturn judgements of other cases.

This work can also be seen as a method for generating exceptions for default logic [8]. Inductive learning has often been used for learning such rules, using multiple learning phases to learn exceptions. For instance in [6] and [7], the learning is split into two phases, the first phase for learning the overly general rules from the examples, and the second phase for specialising the general rules using exceptions. Our work is similar to the second learning phrase in these works, but with an assumed over-general rule (default judgement) to be given from the legal specification. Other similar work such as [3] uses prioritised logic programming to express preferences between the default rules.

7 Conclusion

We have presented a method for reasoning and extracting information from past cases to infer the arguments and attacks present in the decision for the judgement of a new case. We have defined the notion of minimal attacks for identifying the factors relevant to the judgements of cases, and describe how ASP can be used to generate these minimal attacks as well as how these attacks can be used for inferring rules for modelling the judgement. This is achieved by abstracting the casebase into the meta-level information, which is also used for representing the generated rules. These rules are compatible with an existing legal reasoning system PROLEG, which can be seen by applying a simple translation from our PROLOG rules into PROLEG's representation.

For future work, it would be interesting to expand the approach to consider dependencies between factors, by extending the meta-level representation and enhancing the reasoning approach for handling more complex casebases. Furthermore, for new cases that have contradictory judgement to the generated program, instead of rerunning our ASP program to generate a new set of case rules, theory revision could be applied to revise the existing case rules [2].

Acknowledgment. The authors would like to thank Robert Kowalski for his valuable comments.

References

1. Kevin D. Ashley. Reasoning with cases and hypotheticals in hypo. *In: Int. J. ManMachine Studies*, 34:753–796, 1991.

2. Duangtida Athakravi, Domenico Corapi, Alessandra Russo, Marina De Vos, Julian A. Padget, and Ken Satoh. Handling change in normative specifications. In Matteo Baldoni, Louise A. Dennis, Viviana Mascardi, and Wamberto Vasconcelos, editors, *Declarative Agent Languages and Technologies X - 10th International Workshop, DALT 2012, Valencia, Spain, June 4, 2012, Revised Selected Papers*, volume 7784 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2012.
3. Yannis Dimopoulos and Antonis Kakas. Learning non-monotonic logic programs: Learning exceptions. In Nada Lavrac and Stefan Wrobel, editors, *Machine Learning: ECML-95*, volume 912 of *Lecture Notes in Computer Science*, pages 122–137. Springer Berlin Heidelberg, 1995.
4. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
5. Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Thomas Schneider. Potassco: The potsdam answer set solving collection. *AI Commun.*, 24(2):107–124, 2011.
6. Katsumi Inoue. Learning extended logic programs. In *In Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 176–181. Morgan Kaufmann, 1997.
7. Kouzou Ohara, Hideyuki Taka, Noboru Babaguchi, and Tadahiro Kitahashi. Determination of general concept in learning default rules. In Riichiro Mizoguchi and John Slaney, editors, *PRICAI 2000 Topics in Artificial Intelligence*, volume 1886 of *Lecture Notes in Computer Science*, pages 104–114. 2000.
8. R. Reiter. A logic for default reasoning, 1980.
9. Ken Satoh. Translating case-based reasoning into abductive logic programming. In Wolfgang Wahlster, editor, *12th European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings*, pages 142–146. John Wiley and Sons, Chichester, 1996.
10. Ken Satoh. Statutory interpretation by case-based reasoning through abductive logic programming. *JACIII*, 1(2):94–103, 1997.
11. 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 Takashi Onada, Daisuke Bekki, and Eric McCready, editors, *New Frontiers in Artificial Intelligence - JSAI-isAI 2010 Workshops, LENLS, JURISIN, AMBN, ISS, Tokyo, Japan, November 18-19, 2010, Revised Selected Papers*, volume 6797 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2010.
12. Ken Satoh, Takamune Kogawa, Nao Okada, Kentaro Omori, Shunsuke Omura, and Kazuki Tsuchiya. On generality of proleg knowledge representation. In *JURISIN 2012*, 2012.
13. Francesca Toni and Marek Sergot. Argumentation and answer set programming. In Marcello Balduccini and TranCao Son, editors, *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, volume 6565 of *Lecture Notes in Computer Science*, pages 164–180. Springer Berlin Heidelberg, 2011.
14. Radboud Winkels and Rinke Hoekstra. Automatic extraction of legal concepts and definitions. In Burkhard Schäfer, editor, *Legal Knowledge and Information Systems - JURIX 2012: The Twenty-Fifth Annual Conference, University of Amsterdam, The Netherlands, 17-19 December 2012*, volume 250 of *Frontiers in Artificial Intelligence and Applications*, pages 157–166. IOS Press, 2012.

Classification of Precedents by DEMO

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. To determine if the crime is really caused by the accused, the judge examines the correspondence of the actions by the accused to the external elements defined in the Penal Code. The judgement is greatly influenced by the predictability of results and recognition of actions. DEL (Dynamic Epistemic Logic) can formalize reasoning about information change and it's extension Action Model can treat epistemic actions by multi agents. The knowledge or belief of the accused and the judge can be modeled by using this language. This paper classifies some typical precedents and represent them by using DEL and Action Model. Also the process of these precedents in the trial is modeled and implemented on DEMO (Dynamic Epistemic MOdeling) which can specify epistemic models and action models, and the change of the judge's epistemic states during the trial in the court is observed.

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

1 Introduction

For the decision of a crime specified in the Penal Code, first of all it is necessary that the accused's action comes under the external elements (*Actus reus*)[1]. If the action matches to that elements, the crime is determined further if there is neither justifiable cause for noncompliance with the code[2], nor justifiable cause for responsibility[3] [9]. Note that, in this paper, we deal mainly with the process of verifying the correspondence. (We take the position that the external element includes the intent and the negligence. [3])

In general, as external elements for crime, "Action", "Result" and "Causal relationship" are required. The evaluation of "Action" by negligence[4] or intent (*Mens rea*)[5] is greatly influenced by the predictability of results and the recognition of the action by the accused. For example, the intent of external elements is determined based on the recognition of a fact and the predictability of a result. For negligence, the predictability and the result avoidance obligation that is evaluated by the relation to the predictability are the issue. Of actual crimes, there are so many cases[1] in which an unexpected situation happens between the action by the accused and the result, or the action based on a uncertain intent by the accused causes the criminal result, or the duty of care are claimed because in general the result of the action by the accused is predictable.

It is the common understanding that the individual consideration about each crime on a case by case basis is important. This situation make it difficult to determine a crime

in the uniform process and it is difficult to evaluate the judicial decision automatically by using a computer. If there were some common patterns in that process, it should be more easy to handle the evaluation of the judgement in a mechanical manner. That may be done by categorizing the judge's epistemic states during the process of the trial. So we tried to classify some typical precedents, to represent them by using the DEL (Dynamic Epistemic Logic)[4], because Action Model in DEL can represent the individual epistemic states of the accused and the judge, can update the states by various epistemic actions and can express the difference of epistemic states between the accused and the judge. Finally we implemented these models for model checking by using DEMO (Dynamic Epistemic MOdeling)[5].

2 DEL and Action Model

2.1 DEL and Action Model

Knowledge and belief are not static because of the communication of information. Dynamic Epistemic Logic is an extension of epistemic logic[6] with dynamic operators '[]', and $[\pi]\varphi$ is read as "successfully executing program π yields a φ state".[4] Public Announcement Logic[7] is an example of DEL where the epistemic action is only restricted to public announcement.

Action Models[11] are semantic objects and used to describe epistemic actions. The actual Action and it's precondition is described as a point (action point).If two actions are distinguishable, the results are also distinguishable(perfect recall). The factual states are represented as pairs consisting of the factual state and the action executed in that state. The expression (s, s) indicates that action s is executable in state s.

$M, s \models \mathrm{pre(s)}$

The two factual states are indistinguishable, if the following relation exists.

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

For example, 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 didn't know whether the value of P was true or false and there is a link between 0 and 1 for a and b. A letter came to a that told p and a read it and knew that but b couldn't distinguish an action p (a reads a letter which tells p) from an action np (a reads a letter which tells $\neg p$). (But b knew that a knew p or $\neg p$.) This can be described as updating the epistemic state by the action "Read, p". (refer to Fig.1) [4]

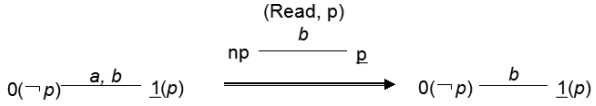


Fig. 1. state transition by an action Read

[1] the guilty acts or typified criminal acts, sometimes called as the objective element of a crime

[2] a reason that there is no illegality about the act that illegality is usually estimated

[3] a reason to deny the responsibility of the act that responsibility is accepted as a general rule

[4] a failure to take reasonable care when they act by taking account of the potential harm to other people

[5] a guilty mind or an intention to commit a crime

Definition 1 (Action model).
Let $\mathscr{L}$ be any logical language for given parameters agents A and atoms P. An $S5$ action model M is a structure $\langle \mathrm{S}, \sim, \mathrm{pre} \rangle$. pre:S $\to \mathscr{L}$ is a preconditions function. A pointed $S5$ action model is a structure (M, s) with s $\in$ S

The syntax of the Action model Language is as follows below.

Definition 2 (Language of Action model).
The language of action model logic $\mathscr{L}_{KC\otimes}(A,P)$ is the union of the formulas
$\varphi ::= p|\neg\varphi|(\varphi\wedge\varphi)|K_a\varphi|C_B\varphi|[\alpha]\varphi$
$\alpha ::= (\mathrm{M},\mathrm{s})|(\alpha\cup\alpha)$

The semantics is as follows.

Definition 3 (Semantics of formulas and actions).
$M,s \models p$ 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 $M',s' \models \varphi$
$(M,s)[\![\mathrm{M},\mathrm{s}]\!](M',s')$ iff $M,s \models \mathrm{pre(s)}$ and $(M',s') = (M\otimes \mathrm{M},(s,\mathrm{s}))$
$[\![\alpha\cup\alpha]\!] = [\![\alpha]\!]\cup[\![\alpha']\!]$

2.2 DEMO

DEMO[5] is a modeling tool for Dynamic Epistemic Logic and programmed in Haskell[13]. 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.

3 Classification of Precedents

3.1 Handling of Recognition and Predictability in the Penal Code

External Elements defined in the Penal Code In general, external elements are roughly divided into subjective one and objective one[3]. (There are also some opposite theory[2] against this, such that both intent and negligence are regarded as the responsibility element and are not included in the external elements.) The objective element contains action, result, some object or entity and causal relationship between an action and a result. This paper focuses on treating these major elements such as “action”, “result”, “causal relationship”. The subjective elements are comprises of intent, negligence.

Intent in the Penal Code Intent is ”an intention to commit a crime”. (The Penal Code Article 38 paragraph 1[8]). At least, a recognition of the objective external element such as an action, a result and predictability for causality are needed. Further, in general, the

probability of occurrence of the result and acceptance of the results by the accused[12] are taken into account by the judge

Intent is roughly divided into two types such as deterministic one which is a representation of ensuring the generation of results and the uncertain one which is a foreseen of an indeterminate result and intent negligence[6] is classified as this type.

Negligence in the Penal Code The negligence is defined in Penal Code Article 38 paragraph 1 as ”The action without the intention 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.”[8]. The negligence is applied in the case of a psychological state where the accused did not foresee the result which might have been able to predict. In recent years, the duty of the accused to avoid such results is more emphasized.[10]

Causal Relationship For a consequent offense[2], a relationship is required between an action and a result. and this is called causal relationship.The causal relationship is the objective requirements for the offender to accept the responsibility for the criminal result. In order to affirm the causal relationship, there should be not only a conditional relationship (without that there should not be this) between the action and the result, but also it is required to be regarded reasonable from the experience of the social life of ordinary people. (Legally sufficient cause[2])

Validation Process of a Crime To determine whether the accused's action conforms to the external elements, the objective and the subjective elements are examined[3]. (refer to Fig.2)

1. Recognition about the objective facts constituting the offense At first, the accused's recognition about the objective external elements such as an action, a result and a causal relationship between them is verified. If the recognition is different from the actual fact, it is considered that a mistake in interpretation of facts[2] has occurred.

2. Recognition about subjective elements Secondly, the intent (the recognition of the predictability, the possibility of occurrence of the result, etc.) and the negligence (the breach of the duty of predicting the result, the breach of the duty of avoiding the result) are examined.

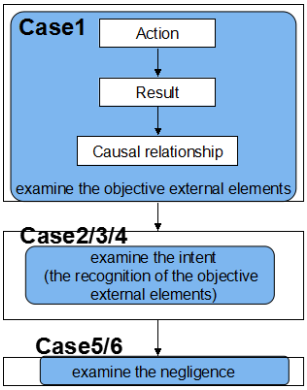


Fig. 2. validation process

3.2 Description of the Issues in the Precedents by DEL

From the above, in order to represent a process of deciding a judgement of precedents dealt in this paper, the followings descriptions are needed.

[6] The accused is uncertain about the realization of crime but even though knowing that crimes may be implemented, he has accepted it.

– Description of facts (action, result, causal relationship) constituting an offense
– Description of recognition about the fact
– Description of predictability and intent

So we define issues in the process of a trial as follows.

Action α_a : an action point of Action Model by agent a
Result φ : a proposition of a possible world (epistemic state)
Causal relationship $[\alpha]\varphi$: a relation between an action and a state in DEL, if there is a causal relationship between an action and a result.
Predictability The predictability is described as a link cut between epistemic states. We describe that agent a can't predicate the state at the state s as a's link between s and t.
Possibility of avoidance This is represented as a link to the state of precondition for the alternative action in Action Models of DEL.
Intent This is represented as a link cut between states. We regard the intention of an action as no link between taking an action and no action ($\top$).

3.3 Classification of Precedents

According to the validation process written in the previous section, there are three main points where predictability or recognition can be the issue in the judgement. The first point is recognition of the objective facts which includes the problem of intervention of unexpected actions and a mistake in recognizing causal relationship. The second is the intent of the accused and the problem of intent negligence appears at this point. The third is the negligence which includes the problem of predictability and possibility of avoiding results. From precedents often cited [1], some typical examples are listed below and they are classified into 6 cases from the point of view of recognition and predictability (refer to Table 1) and these case are mapped to the three points above (refer to Fig.2).

4 Implementation and Result

The model checking for these cases is implemented by using DEMO.[5] In this implementation, the accessibility relations are restricted to KD45 relation, because the accused and the judge may make a mistake in recognizing the facts.

Updating the states is executed in two steps. The first step is by the accused's actions from the beginning of a crime to the end (or the start of a trial where the judge has no prejudice.) and the second step is by the judge's actions from the start of a trial to the final judgement. In updating, the epistemic actions as follows are used.

message which notifies propositions to particular persons and the others doesn't know whether the message has reached nor the proposition's value. This corresponds to the situation that a prosecutor gives new evidence and the judge examines this evidence and the others can't know the determination in the judge's mind.
secret which inform propositions to the judge or the accused and the others can't be aware of the change in the accused's mind or the judge's mind.

Table 1. classification of precedents

Precedents/**Issue**	Outline	Classification
Defendant of injury resulting in death case (No. A35, 2003) **Intervention of the action**	4 defendants commit the outrage repeatedly. The victim ran away to broke into the highway nearby and was run over by a car and died. The judge applied injury resulting in death because there is a causal relationship between the outrage and the death.	**Case1** (Objective) The intervention of the unexpected action by the victim or the third person. (legally sufficient cause, the degree of the contribution to the result)
Defendant of unlawful arrest and illegal confinement resulting in death case (No. A2901, 2005) **Intervention of the action**	The accused and the accomplice imprisoned the victim in a rear trunk of a passenger car and parked by the street. Then a passenger car driven by a third person bumped into the rear trunk and the victim died. The judge found the accused guilty of illegal confinement resulting in death.	
Defendant of indecent document sale (No. A1713, 1953) **the recognition of the meaning**	The accuseds who translated and published the "Lady Chatterley's Lover", were charged with selling obscene document	**Case2** (Normative) The recognition of meaning (No fitting to the legal concept exactly)
Defendant of murder case (No. RE 517, 1923) **mistake of causal relationship** similar case: defendant of fraud case (No. A1625, 2003) **the result happened too early**	The accused tried to murder the victim by strangulation and then made an attempt to conceal the crime by burying him in the sand of the coast, but he did not die at that time and died because of sand absumption. The judge determined there was an intention because the accused make a mistake of the causal relationship.	**Case3** (Descriptive) The mistake of a process (the second action can be regarded as a part of the first act)
Dealing with stolen goods case (No. RE238, 1947) **intent negligence** similar case: The Stimulant Drug Control Law violations (No. A1038, 1998)	The accused bought stolen clothes without deterministic recognition of their being stolen. The judge apply the crime of paid acquisition of stolen goods which were stolen by another	**Case4** (Descriptive) Intent negligence (an acceptance of a result or high probability)
Hokkaido University electric scalpel case (No. U219, 1974) **An offense against the duty of care**	The doctor and the nurse made a mistaken of connecting the cable of the scalpel incorrectly, and the patient's right foot below the knee was damaged. The judge ruled professional negligence resulting in bodily injury to the nurse and the innocence to the doctor.	**Case5** (Negligence) Predictability (the offense against the duty of care)
The use of HIV contaminated blood in Teikyo University hospital (No. WA1879, 1996) **An offence against the avoidance obligation**	The doctor used unheated blood and the patient become HIV positive. The judge applied the innocence to the doctor.	**Case6** (Negligence) Avoidance (the offense against the duty of avoiding the result)

public which is the same as public announcement. This is used to describe the common sense which influence the criminal actions.

For example (message b q), this demonstrates that the judge knows q but the accused doesn't know that as the result of updating by the action of message. (refer to Fig. 3) When we define the propositions, we take "p" as the primary (or the external element)

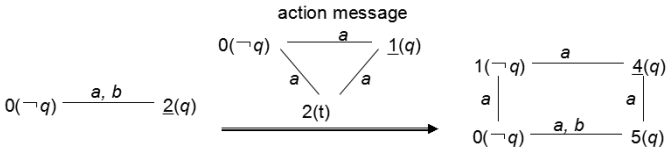


Fig. 3. update by message

and "$q, r, ...$" as the secondary (the extraneous element).

4.1 Case 1 : The intervention of the unexpected action (Objective Case)

The accused commit the outrage repeatedly. So the victim ran away to break into the highway nearby, was run over by a car and died. In this case, the accused is on the crime of inflicting injury and the cause of the victim's death is an issues.

- Proposition p: The victim is injured.
- Proposition q: The victim is driven to the emotional corner.
- Agent a:The accused (in the following all cases)
- Agent b:The judge (in the following all cases)

We set the proposition p (external element) as the injury and q (substantial element) as the cause of a successive action. Four epistemic states are set where the valuation depends on this two propositions and it's true/false binary values respectively.

As the crime proceeds, the accused or the plaintiff takes actions. We interprets these non-epistemic actions as corresponding (whose preconditions are equivalent) epistemic actions which are points of Action Model. In this case there are two non-epistemic actions, "p" and "q".

"harm the victim" whose precondition is $\neg p$ (for harming)

"run into highway" whose precondition is q (for breaking into the highway)

It is necessary to update the epistemic states by two corresponding epistemic actions concerning two non-epistemic actions described above. The accused can recognize his own action of harming the victim, so we update the states by the action of sending a secret message to himself whose precondition is the same as of the precondition of his action "harm". On the other hand, the victim's action "breaking into the highway" can't be predicted by the accused, so no additional information is send to himself and the links are remain unchanged.That is the state at the beginning of the trial. A part of program code is as follows.

```
intervent  = initE[P 0, Q 0]
initInt = upds intervent [secret [a] (Neg p)]
```

The Agent a's links between states where q is true or $\neg q$ is true. (which are the agent a's world where he believes $\neg p$, but he can't predict the victim's action q(running into the highway).

Then we updates this epistemic state by the action of the judge. This process is regarded as the proceeding of the trial at the court. The actions for the final state are as follows. (refer to Fig. 4)

message b (Neg p) The judge knows the accused harmed the victim.
secret [b] (Disj [p, q]) The judge thinks the victim was cornered by the accused's action.

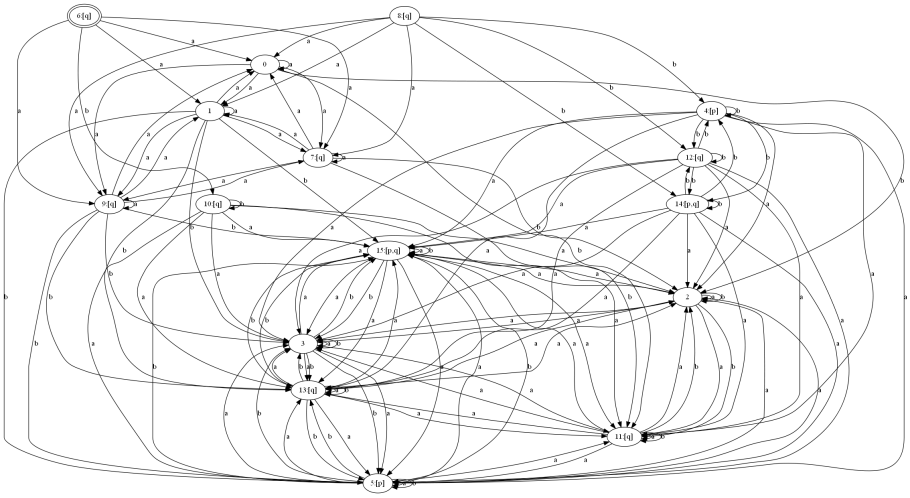


Fig. 4. Case1 the state after the trial

4.2 Case2：Recognition of the meaning (Normative Case)

The accused who translated and published the novel "Lady Chatterley's Lover" were charged with obscene. document sale.
The propositions and the states are as follows.

- Proposition p: The novel is obscene. q: The public order is violated.
- Four states (0..3), 0:$\neg p, \neg q$, 1:p, 2:q, 3:p, q:

The accused's actions which update the epistemic states through the crime

- "It is known that the violation of public order is a crime."' public (Disj[Neg q, p])
- "The accused knows publishing this novel violates the public order." secret [a] q

The judge's actions which update the epistemic states during the trial

- "The judge thinks the novel is obscene in the legal meaning." secret [b] p

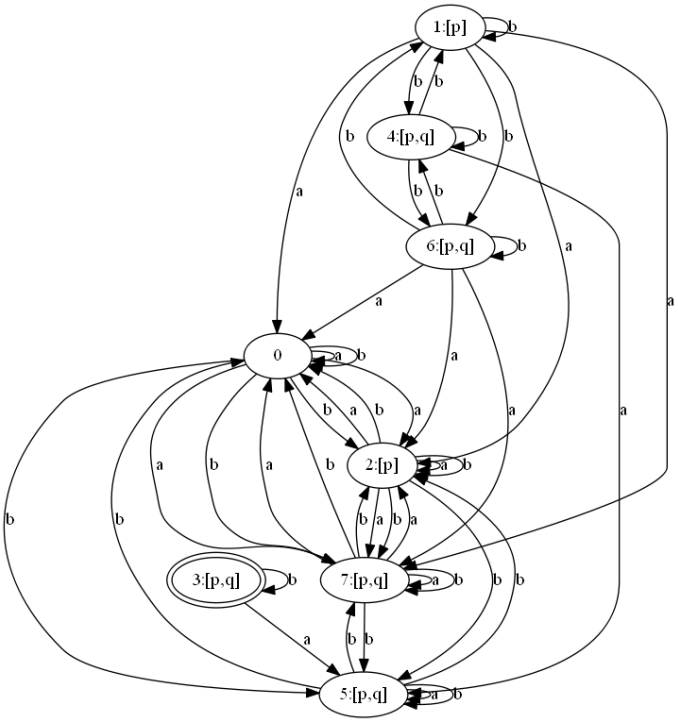


Fig. 5. Case2 the state after the trial

– “The judge thinks the accused knows the violation of the public order.” secret [b] (K a q)

After these updates, the judge knows that the accused knows the illegality and he can inflict the punishment based on the accused’s intention. This can be checked in DEMO as follows.

```
*DEMO> isTrue (upds initMeaning [secret [b] p, secret [b] (K a q)])
(K b (K a p))
True
```

4.3 Case3：Mistake of the causal relationship (Descriptive Case)

The accused tried to murder the victim by strangulation and then made an attempt to conceal the crime by burying him in the sand of the coast, but he did not die at that time and died because of sand absumption.
The propositions and the states are as follows.

– Proposition *p*: The victim is dead. *q*: The victim survives. the accused’s strangulation.

The accused’s actions which update the epistemic states through the crime

– “The accused recognizes his own action strangulation” secret [a] (Neg p) (But the accused does not know the victim survives.)

The judge's actions which update the epistemic states during the trial

- "The judge knows the accused strangulated the victim." message b (Neg p)
- "The judge thinks the concealment is a part of the process of strangulation." secret [b] (Disj [p, q])

The the final state is the same as Case1.

4.4 Case4 : The Intent negligence

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

- Proposition p: The goods are stolen. q:The probability of being stolen is high.
- State four states (0..3) 0:$\neg p, \neg q$, 1:p, 2:q, 3:p, q Agent a, b can not distinguish these states.

The accused's actions which update the epistemic states through the crime

- "'It is known that the probability is high then the goods are perhaps stolen." public (Disj[Neg q,p])
- "The accused thinks that the probability of being stolen is high." secret [a] q

The judge's actions which update the epistemic states during the trial

- "The judge thinks the goods are stolen."secret [b] p
- "The judge thinks the accused knows the probability is high." secret [b] (K a q)

The result is the same as case2. After these updates, the judge knows that the accused knows the illegality and he can inflict the punishment based on the accused's intention.. This can be checked by DEMO, too.

```
*DEMO> isTrue (upds initWilneg [secret [b] p, secret [b] (K a q)])
(K b (K a p))
True
```

4.5 Case5 : The offense against the 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 negligence resulting in injury.
The propositions and the states are as follows.

- Proposition p: The cables are connected wrongly. q: the patient needs the operation.
- State four states (0..3), 0:$\neg p \wedge \neg q$ 1:p 2:q 3:p,q (Agent a,b can not distinguish these states.)

The accused's actions which update the epistemic states through the crime

– “The accused thinks he has to operate. on the patient for treatment.” secret [a] q

The judge’s actions which update the epistemic states during the trial

– “’The judge thinks the result can be predicted generally.” secret [b] p
– “’The judge knows the accused has to operate.” message b q

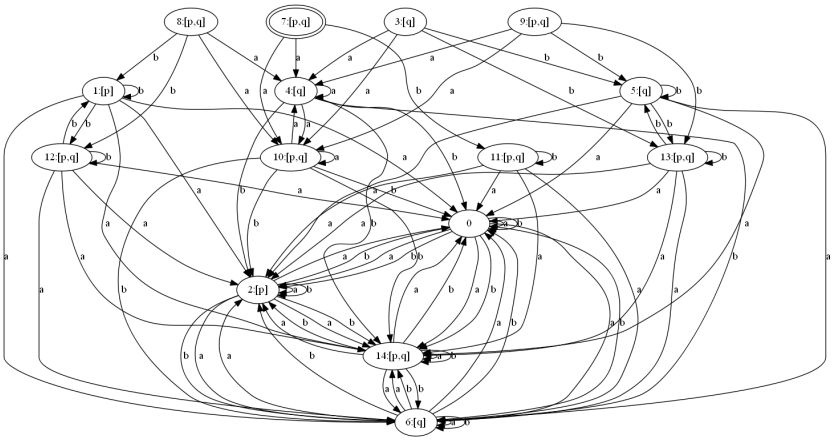


Fig. 6. Case5 the state after the trial

4.6 Case6：The offense against the duty of the avoidance

The doctor used an unheated blood and the patient become HIV positive. The doctor is innocent.
The propositions and the states are as follows.

– Proposition p: The medicine is infected with HIV. q: The patient is hemophilic . r: Cryoprecipitate is hard to obtain
– Action medicine 0: do nothing 1: give unheated blood normally 2: give unheated blood (HIV) 3: give cryoprecipitate Agent a can’t distinguish 2 from 1. The preconditions are $\neg q$ for 0, $p \wedge \neg q$ for 1, $p \wedge q$ for 2, $p \wedge r$ for 3.

The accused’s actions which update the epistemic states through the crime

– “The accused thinks he has to treat the patient.” secret [a] p
– “The accused thinks it is better and easy to give unheated blood than giving cryoprecipitate.” secret [a] r

The judge’s actions which update the epistemic states during the trial

– “The judge knows the patient is infected by HIV.” message b p
– “The judge knows the accused has to treat the patient.” message b q “The judge knows it is easy and better to give unheated blood than to give cryoprecipitate generally.” secret [b] r

the final state after the trial
Model is consists of 75 states and the actual state is one where the accused can't distinguish p from $\neg p$ and r is monolithic (he doesn't have the possibility of taking the other aciton.). The judge can distinguish p, q, r.

4.7 Patterns of the final states

The final states of six cases after updating can be categorized into three graphical patterns according to the actual state and the links from this state. For the case where the crime is commit by the accused intentionally, the final state can be summarized to the pattern in which p is monolithic (distinguishable from the other). This pattern can be categorized into two kinds, one is concerning the judge's recognition about facts (Case1,3) where the secondary elements are important for the judge, and the other is concerning the accused's recognition of the meaning of his actions (Case2,4) where the judge can find the accused's intention based on the common sense.

On the other hand, the case where the crime is commit by the accused's negligence, the final state can be summarized to the pattern in which p and $\neg p$ are reachable from the actual world by the accused's relation (he can't distinguish these states). (refer to Table 2 p :primary proposition, q: secondary proposition)

- Intentional case (Case1,2,3,4)
  - The primary proposition p is distinguishable (monolithic) for both the accused and the judge. The secondary proposition q is not distinguishable for the accused. (Case1,3) (Fig. 7)
  - The primary proposition p and the secondary proposition q are distinguishable for both the accused and the judge. (Case2,4) (Fig. 8)
- Negligent case (Case5,6)
  - The primary proposition p is distinguishable for the judge. But not for the accused. The secondary propositions q/r are distinguishable for the both. (Case5,6) (Fig. 9) In case6, the probability of p is low for the accused (almost distinguishable).

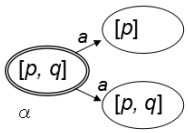


Fig. 7.

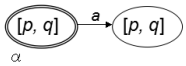


Fig. 8.

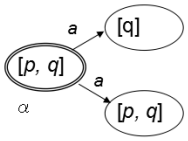


Fig. 9.

[7] He can deduce from the common sense.
[8] The probability is very low.
[9] He has no possibility of taking the other action.

Table 2. Patterns of final states

case	int/negl	*p* for *a*	*p* for *b*	*q* for *a*	*q* for *b*
Case1	Int	D	D	N	D
Case2	Int	D[7]	D	D	D
Case3	Int	D	D	N	D
Case4	Int	D	D	D	D
Case5	Neg	N	D	D	D
Case6	Neg	N[8]	D	D[9]	D

Int: intention Neg: negligence D: distinguishable N: not distinguishable

5 Conclusion and Further Directions

5.1 Conclusion

The process of making judgement according to the Penal Code is examined and it is found that recognition and predictability are the main factors of the complicated process of verifying the correspondence of the accused's action to the external elements. We classify the examples into six cases by the issues during the process above.

The predictability and the recognition in the precedents can be regarded as the link between epistemic states. If the states or the action is predictable from a state, there is no link between the two states.

These precedents are modeled and implemented on DEMO. In this model, the epistemic states are updated by epistemic actions which simulates the change of the judge's epistemic state. Finally these result states after updating can be categorized into three cases focusing on the actual state and it's links of the accused and the judge from this state. We find that typical cases where recognition and the predictability are the issue in judgements according to the Penal Code can be described in these basic patterns.

5.2 Further Directions

There are many cases in which the accused's actions change the facts. In this paper, we maintain the assumption about the perfect recall and distinguish the preconditions of the same actions which leads to the different results. if we can describe the post conditions of an action, it becomes easier to describe these cases.

To describe negligence, it is needed to describe the probability regarding to the predictable degree. To do that, the link between states should be directional and prioritized.

Finally, in implementing these precedents, the number of the states and the points of actions, the time of calculation increased rapidly and we have to examine the use of these patterns to reduce the number of states in calculation.

References

1. Noriyuki Nishida, Atsushi Yamaguchi, Hitoshi Saeki: *The 100 selected precedents of the Penal Code 1, 6th edit.* Yuhikaku (2008) (in Japanese)
2. Atsushi Yamaguchi: *Criminal law general part* Yuhikaku (2006) (in Japanese)

3. Minoru Oya: *Lecture note on Criminal law, general part* Seibundo (2007) (in Japanese)
4. Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi: *Dynamic Epistemic Logic* Springer (2008)
5. J. van Eijck: Dynamic episteimc modeling. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam 2004
6. R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi: *Reasoning about Knowledge* The MIT Press (1995)
7. J.A. Plaza: Logics of public communications, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems* (1989)
8. The Penal Code. `http://www.japaneselawtranslation.go.jp/law/detail/?id=1960`
9. M. E. Mayer: *Der allgemeine Teil des deutsch Strafrechts, 2. Aufl.* (1923)
10. Hideo Fujiki (eds.): *Criminal Negligence, a dispute on the old and new negligence theory* Gakuyoushobou (1975) (in Japanese)
11. A. Baltag, L.S. Moss: Logics for epistemic program, *Synthese, vol.139 Issue 2* (2004)
12. Shigemitsu Dandou: *The criminal law essentials, general remarks, 3rd edit.* Soubunsya (1990) (in Japanese)
13. Simon Peyton Jones (eds.): Haskell 98 Language and Libraries. `http://www.haskell.org/onlinereport/`(2002)

Analyzing 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. This paper proposes a logical framework for analyzing reliability change of an agent. Firstly, we formalize a source of information by using the notion of signed statement within logic of agents' beliefs. Secondly, we introduce two dynamic operators into our syntax: downgrade and upgrade operators. The downgrade operator allows an agent to downgrade some specified agents with respect to the degree of reliability, while the upgrade operator allows an agent to upgrade them. Finally, we demonstrate our formalization by a legal case in Thailand. In order to derive an agent's belief from received signed statements, we employ a policy of aggregating signed statements.

Keywords: reliability change, belief, legal case, modal logic, signed information

1 Introduction

In agent communication, an agent has to decide what she should believe. In order to decide which information should believe, the agent needs some criteria. A common criterion is to consider the reliability of an information source. The agent may accept or reject the received information depending on the reliability of sources. If the agent considers that a source of received information is not reliable, she would reject the received information. On the other hand, the agent may accept the received information if she considers that the source is reliable. This shows that the notion of reliability plays an important role in agent communication.

In the process of the trials, when a judge received a piece of information, the judge has to decide that she should believe the received information or not. This is the same problem as in agent communication. Thus, the judge also needs the concept of reliability of the information source, i.e., when the judge received a piece of information from a witness, the judge should consider if the witness is reliable or not.

Recently, many studies [1–3] described the use of logical approaches in the legal systems. There are also several works that studied on modal logic and dynamic epistemic logic (DEL) in [4, 5]. Moreover, many works [6–8] proposed a logical framework for formalize the reliability of the information source. Among of them, Lorini et al. [8] introduced a modal framework for reasoning about signed information. In this framework, the agents can keep track of information source by using the notion of signed statement. They also considered the notion of reliability over the information sources. However, they did not deal with dynamics of the reliability relation of agents.

For this reason, we propose to formalize reliability change of an agent. First, we apply a concept of signed statement based on [8] to formalize the source of information. Then, we introduce two dynamic operators, i.e. downgrade and upgrade operators. The downgrade operator is used for downgrading the reliability of agents, while the upgrade operator is used for upgrading. With these operators, we can capture the change of reliability ordering of agents. Finally, we adopt a careful policy based on [8] in order to aggregate the received signed statements and decide which signed information should believe. Moreover, we demonstrate our formalization in an example of legal case in Thailand.

The remainder of this paper is organized as follows. Section 2 describes the details of the target legal case. Then, a logical formal tool for analyzing this legal case is presented in Section 3. In Section 4, we propose a dynamic logical analysis of the target legal case. Finally, our conclusion and future works are stated in Section 5.

2 Target Legal Case

Firstly, we summarize a story of our target legal case that occurred on 26th January 2003 in Trang province, Thailand [1] as follows.

> One day, a victim v had a drink with his friends f_1, f_2 and d at f_2's house. After that, v was punched and stabbed with a hand scraper in the back by an offender, and as a result, v was bleeding at the lung. However, v was still alive.

In the inquiring stage, a police po, who is an inquiry official, was interviewing four witnesses v, f_1, f_2, mo that gave the following statements.

(I_1) v tells that d is the offender who punched and stabbed v.
(I_2) f_1 also tells that d is the offender who punched and stabbed v.
(I_3) f_2 states that v and d had a dispute, but did not have any fighting. The more details can be shown as follows.
> Firstly, f_2 had a drink with v, f_1, and d at f_2's house. Around 21 o'clock, v and d had a dispute. f_2 came to forbid them from fighting. After that, v, f_1 and d left f_2's house without any fighting.

(I_4) mo, who is v's mother, tells that d is the offender according to v's saying. The more details can be shown as follows.
> At night of the accident, mo visited v in the hospital. Then, v told her that v went to have a drink with d, f_1 and f_2 at f_2's house. During drinking, v and d had a dispute, then d punched v and stabbed with a hand scraper in the back of v.

From the interview, po accused d of attempting to kill v, but d refused this accusation.

In the Civil Court, v and f_1 changed their statements as follows.

v tells that one of a group of unknown teenagers is the offender who punched and stabbed v by a knife. The more details can be shown as follows.

[1] This legal case can be referred from `http://deka2007.supremecourt.or.th/deka/web/search.jsp` (in Thai).

> At 19 o'clock, v and f_1 were invited to drink by x who was their neighbor. After drinking, v and f_1 went to a market. While f_1 was riding a motorcycle from x's house, a group of unknown teenagers came to punch v. Then, one of them armed with a knife stabbed in the back of v.

f_1 can only state that v was punched by d, but cannot state that v was stabbed by d or not. The more details can be shown as follows.

> At 18 o'clock, v and f_1 were invited to drink by d who was their friend. v and f_1 went to f_2's house by their motorcycle (v was a rider and f_1 was a passenger), and d also followed them. Then, v had a drink with f_1, f_2, d and two other friends at f_2's house. Around 21 o'clock, v and d had a dispute and then d punched v. f_2 came to forbid them from fighting, while f_1 went to bring the motorcycle. After that, v came to sit behind f_1's motorcycle and said that he was stabbed.

Moreover, po was called to be a witness for testifying all statements in the inquiring stage. Thus, there are six testimonies in the Civil Court as follows.

(T_1) v tells that one of a group of unknown teenagers is the offender who punched and stabbed v by a knife.
(T_2) f_1 can only state that v was punched by d, but cannot state that v was stabbed by d or not.
(T_3) po states that v tells that d is the offender who punched and stabbed v.
(T_4) po states that f_1 tells that d is the offender who punched and stabbed v.
(T_5) po states that f_2 states that v and d had a dispute, but did not have any fighting.
(T_6) po states that mo tells that d is the offender according to v's saying.

From the above testimonies, testimonies of v and f_1 in the inquiring stage (T_3 and T_4) are more reliable than that in the Civil Court (T_1 and T_2) because of the following reasons: First, the judge believed that po and f_2 had never had any arguments against the defendant. So, there is no reason that they will allege or testify against the defendant to be punished. Second, according to T_1 and T_2, the judge believed that v and f_1 tried to distort the facts in order to prevent d who is their friend from the punishment.

Therefore, the judge decided that d is the offender and intend to kill v by the following reasons.

- Since the hand scrapper belongs to a dangerous weapon, the defendant uses it for attacking that would be lethal. This shows that the defendant intends to kill v.
- The defendant stabs v while v is turning back. At that time, the defendant can choose other alternative positions for attacking. Nevertheless, the defendant strongly stabs v in the lung that is a vital organ. It is obvious that the defendant intends to kill v.
- From the statement of doctor, v is seriously injured, i.e., there is air leaking and bleeding in the chest cavity and lung, and would be dead unless v gets the treatment in time. This shows that the attack of the defendant is such dangerous resulting in death.

For this reason, the Civil Court judged the defendant to be sentenced to ten years' imprisonment by Article 288 and Article 80 of Penal Code: [2]

> *Article 288* (offence causing death): Whoever, murdering the other person, shall be imprisoned by death or imprisoned as from fifteen years to twenty years.
>
> *Article 80* (commitment): Whoever commences to commit an offence, but does not carry it through, or carries it through, but does not achieve its end, is said to attempt to commit an offence. Whoever attempts to commit an offence shall be liable to two-thirds of the punishment as provided by the law for such offence.

In the Appeal Court and the Supreme Court, the defendant appealed that he did not intend to kill v; in fact, he only intended to attack v. However, the judge agreed with the decision of the Civil Court and adopted the result, i.e., the defendant is imprisoned for ten years by Articles 288 and 80 of Penal Code.

3 Formal Tool for Analyzing Target Legal Case

3.1 Static Logic of Agents' Beliefs for Signed Information

To analyze the previous legal case from the logical point of view, we introduce a modal language, based on the previous work [8], which enables us to formalize the 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:

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

where $p \in \mathsf{Prop}$ and $a, b, c \in G$. For intuitive readings of formulas, the reader can refer to Table 1. 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 equally reliable can be defined as $(b \leqslant_a c) \wedge (c \leqslant_a b)$. We define $\vee$, $\rightarrow$, $\leftrightarrow$ as ordinary abbreviations. Our syntax is different from [8] in at least two respects. First, we do not introduce the universal quantifier for agents. This is because we realized that most of the ideas in [8] are done without quantifiers for agents when the set of agents are finite, i.e., the universal quantifier for a finite domain is just reduced to the conjunction of finite conjuncts. Second, we relativize the notion of reliability ordering $\leqslant$ to each agent. In order to analyze our example in logical perspective, we need to formalize belief change of a judge of the Civil Court and we regard that belief change is induced by reliability change. However, there is no need for us to change reliability ordering of the other agents other than a judge of the Civil Court. This is why we propose the notion of reliability between agents depending on an agent.

[2] An English translation of the articles can be referred from `http://www.thailaws.com/`.

$\mathsf{Bel}(a, \varphi)$	: agent a believes that φ.
$\mathsf{Sign}(a, \varphi)$	: agent a signs statement φ.
$b \leqslant_a c$	: from agent a's perspective, agent b is at least as reliable as agent c.
$\mathsf{Sign}\big(a, \mathsf{Sign}(b, \varphi)\big)$	: agent a signs statement that agent b signs statement φ.
$\mathsf{Bel}\big(a, \mathsf{Sign}(b, \varphi)\big)$	: agent a believes that agent b signs statement φ.
$\mathsf{Bel}\big(a, b \leqslant_a c\big)$	: agent a believes that from agent a's perspective, agent b is at least as reliable as agent c.

Table 1. Examples of Static Logical Formalization

Let us provide Kripke semantics with 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 *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 mapping from W to $\mathcal{P}(G \times G)$ intended to capture the reliability ordering between agents for each agent, and $V : \mathsf{Prop} \to \mathcal{P}(W)$ is a valuation. In what follows, we simply write $b \preccurlyeq_a^w c$ for $(b, c) \in \preccurlyeq_a (w)$. 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.

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

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

Definition 1. *We say that a model* $\mathfrak{M} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq_a)_{a\in G}, V)$ *is a si-model* (*a model for signed information*) *if the following conditions are satisfied:*

(i) R_a *is transitive* (wRv *and* vRu *jointly imply* wRu *for all states* w, v, u) *and Euclidean* (wRv *and* wRu *jointly imply* vRu *for all states* w, v, u).

(ii) S_a *is serial* (*for any state* w*, there is some state* v *such that* wS_av), *transitive and Euclidean.*

(iii) $\preccurlyeq_a^w \subseteq G \times G$ *is a total pre-ordering between agents, i.e.,* $\preccurlyeq_a^w$ *is reflexive* ($b \preccurlyeq_a^w b$ *for all agents* b), *transitive and comparable* (*for any agents* b *and* c, $b \preccurlyeq_a^w c$ *or* $c \preccurlyeq_a^w b$).

(iv) ($R_a(w) \subseteq \{\, v \in W \mid b \preccurlyeq_a^v c \,\}$ *or* $R_a(w) \subseteq \{\, v \in W \mid c \preccurlyeq_a^v b \,\}$) *for all agents* b *and* c.

The first and second items of this definition ensure us that we never sign a contradiction (due to seriality of S_a), and $\mathsf{Bel}(a,\cdot)$ and $\mathsf{Sign}(a,\cdot)$ are both positively and negatively introspective. The reader may wonder the forth item, but it implies that there is some total pre-ordering between agents such that agent a believes that it is her reliability ordering (cf. [8]). Corresponding to these constraints, we obtain the following validites.

Proposition 1. *The following are all valid in all si-models.*

(i) $\mathsf{Bel}(a,p) \to \mathsf{Bel}(a,\mathsf{Bel}(a,p))$ *and* $\neg\mathsf{Bel}(a,p) \to \mathsf{Bel}(a,\neg\mathsf{Bel}(a,p))$.
(ii) $\neg\mathsf{Sign}(a,\bot)$, $\mathsf{Sign}(a,p) \to \mathsf{Sign}(a,\mathsf{Sign}(a,p))$, *and* $\neg\mathsf{Sign}(a,p) \to \mathsf{Sign}(a,\neg\mathsf{Sign}(a,p))$.
(iii) $b \leq_a b$, $(b \leq_a c \wedge c \leq_a d) \to b \leq_a d$, *and* $b \leq_a c \vee c \leq_a b$.
(iv) $\mathsf{Bel}(a, b \leq_a c) \vee \mathsf{Bel}(a, c \leq_a b)$
(v) $b \leqslant_a b$, $(b \leqslant_a c \wedge c \leqslant_a d) \to b \leqslant_a d$, *and* $b \leqslant_a c \vee c \leqslant_a b$.
(vi) $\mathsf{Bel}(a, b \leqslant_a c) \vee \mathsf{Bel}(a, c \leqslant_a b)$

Based on Definition 1 and [8], we can rank agents as follows. The agents who are equally reliable are categorized in the same group. We define such ranking by giving a partition $(\mathsf{C}_i^a)_{i\leq M}$ to G, where M is a natural number representing the maximum rank. First, we define C_1^a which stands for 'a group of agents which is the most reliable from a's perspective' by the following formula.

$$c \in \mathsf{C}_1^a =_{def} \bigwedge\nolimits_{b\in G}(c \leqslant_a b),$$

where we recall that G is a finite set of agents, and $a, b, c \in G$. Note that we relativize the notion C_1^a to a specified agent a, because the notion of reliability ordering $\leqslant_a$ depends on a specified agent a. This is a difference from [8]. Then, we can rank the groups of agents C_i^a such that $i > 1$ as follows.

$$c \in \mathsf{C}_i^a =_{def} \left(\Big(\bigwedge_{1\leq j\leq i-1} \neg(c \in \mathsf{C}_j^a)\Big) \wedge \Big(\bigwedge_{b\in G}\Big(\big(\bigwedge_{1\leq j\leq i-1} \neg(b \in \mathsf{C}_j^a)\big) \to (c \leqslant_a b)\Big)\Big)\right).$$

This implies that all agents in C_i^a are equally reliable, and if $i <_{\mathbb{N}} j$ then $c <_a b$ for all agent $c \in \mathsf{C}_i^a$ and agent $b \in \mathsf{C}_j^a$.

3.2 Downgrade and Upgrade Operations for Agents

In order to change a reliability ordering between agents from a particular agent's perspective, we introduce two dynamic operators, i.e., the downgrade operator $[H \Downarrow_\varphi^a]$ and the upgrade operator $[H \Uparrow_\varphi^a]$, where $H \subseteq G$ is a set of agents. Our intended reading of $[H \Downarrow_\varphi^a]\psi$ is 'after the agent a downgraded the agents in *H* who sign statement φ, ψ', and we can read $[H \Uparrow_\varphi^a]\psi$ as 'after the agent a upgraded the agents in *H* who sign statement φ, ψ'. Semantically speaking, $[H \Downarrow_\varphi^a]$ makes all agents in *H* who sign φ less reliable than all the other agents, and $[H \Uparrow_\varphi^a]$ makes all agents in *H* who sign φ more reliable than all the other agents.

Before giving a detailed semantics, let us demonstrate the effect of $[H \Downarrow_\varphi^a]$ and $[H \Uparrow_\varphi^a]$ by figures. Firstly, we assume that rectangle G of Fig. 1(i) represents a fixed

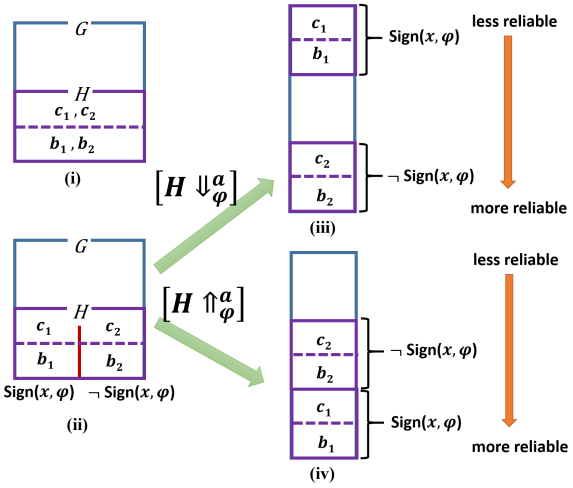


Fig. 1. Reliability Downgrade by $[H \Downarrow^a_\varphi]$ and Reliability Upgrade by $[H \Uparrow^a_\varphi]$

finite set of agents. Secondly, we will select a specified set of agents in order to change their reliability that can be represented by a rectangle H, and we assume that $b_1 \approx_a b_2 <_a c_1 \approx_a c_2$ hold, i.e., agents b_1 and b_2 are equally reliable and more reliable than agents c_1 and c_2 that are equally reliable from agent a's perspective. In this sense, b_1, b_2, c_1 and c_2 are situated as shown in Fig. 1(i). Then, if we focus on the agents who sign statement φ, H is divided into two equal vertical parts, i.e., $\mathsf{Sign}(x, \varphi)$ and $\neg\mathsf{Sign}(x, \varphi)$ as shown in Fig. 1(ii), namely by the set $\{x \in H \mid \mathfrak{M}, w \models \mathsf{Sign}(x, \varphi)\}$ and the set $\{x \in H \mid \mathfrak{M}, w \models \neg\mathsf{Sign}(x, \varphi)\}$. Next, if agent a downgrades all the agents signing statement φ, we downgrade all of them less reliable than the other agents as in Fig.1(iii). On the other hand, if agent a upgrades all the agents signing statement φ, we upgrade all of them more reliable than the other agents as in Fig.1(iv).

Definition 2. *Given 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* $[H \Downarrow^a_\varphi]$ *on* $\mathfrak{M}$ *and* $w \in W$ *is defined as follows.*

$$\mathfrak{M}, w \models [H \Downarrow^a_\varphi]\psi \;\text{ iff }\; \mathfrak{M}^{H\Downarrow^a_\varphi}, w \models \psi,$$

where $\mathfrak{M}^{\Downarrow^a_\varphi} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq'_d)_{d\in G}, V)$ *and* $\preccurlyeq'_d$ *is defined as:*

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

$\big(b, c \in H$ *and* $\mathfrak{M}, w \models \mathsf{Sign}(b, \varphi) \wedge \mathsf{Sign}(c, \varphi)$ *and* $b \preccurlyeq^w_a c\big)$ *or*

$\big(b, c \in (G \setminus H) \cup \{x \in H \mid \mathfrak{M}, x \models \neg\mathsf{Sign}(x, \varphi)\}$ *and* $b \preccurlyeq^w_a c\big)$ *or*

$\big(b \in (G \setminus H) \cup \{x \in H \mid \mathfrak{M}, x \models \neg\mathsf{Sign}(x, \varphi)\}$ *and* $c \in H$ *and* $\mathfrak{M}, w \models \mathsf{Sign}(c, \varphi)\big)$

Definition 3. *Given 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* $[H \Uparrow^a_\varphi]$ *on* $\mathfrak{M}$ *and* $w \in W$ *is defined as follows.*

$$\mathfrak{M}, w \models [H \Uparrow^a_\varphi]\psi \;\text{ iff }\; \mathfrak{M}^{H\Uparrow^a_\varphi}, w \models \psi,$$

where $\mathfrak{M}^{\Uparrow^a_\varphi} = (W, (R_a)_{a\in G}, (S_a)_{a\in G}, (\preccurlyeq'_d)_{d\in G}, V)$ *and* $\preccurlyeq'_d$ *is defined as:*

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

$$\big(b, c \in H \text{ and } \mathfrak{M}, w \models \mathsf{Sign}(b,\varphi) \wedge \mathsf{Sign}(c,\varphi) \text{ and } b \preccurlyeq^w_a c\big) \text{ or}$$
$$\big(b, c \in (G \setminus H) \cup \{x \in H \mid \mathfrak{M}, x \models \neg\mathsf{Sign}(x,\varphi)\} \text{ and } b \preccurlyeq^w_a c\big) \text{ or}$$
$$\big(c \in (G \setminus H) \cup \{x \in H \mid \mathfrak{M}, x \models \neg\mathsf{Sign}(x,\varphi)\} \text{ and } b \in H \text{ and } \mathfrak{M}, w \models \mathsf{Sign}(b,\varphi)\big)$$

Since the space is limited, we focus the recursive validities of our downgrade operator $[H \Downarrow^a_\varphi]$, though the recursive validities of the upgrade operator are given similarly.

Proposition 2 (Recursive Validities). *The following are valid on all models. Moreover, if* ψ *is valid on all models, then* $[H \Downarrow^a_\varphi]\psi$ *is also valid on all models.*

$$\begin{array}{lcll}
[H \Downarrow^a_\varphi]p & \leftrightarrow & p & \\
[H \Downarrow^a_\varphi](b \leqslant_d c) & \leftrightarrow & b \leqslant_d c & (d \neq a) \\
[H \Downarrow^a_\varphi](b \leqslant_a c) & \leftrightarrow & b \leqslant_a c & (b, c \in G \setminus H) \\
[H \Downarrow^a_\varphi](b \leqslant_a c) & \leftrightarrow & \big(\mathsf{Sign}(b,\varphi) \wedge \mathsf{Sign}(c,\varphi) \wedge (b \leqslant_a c)\big) \vee & \\
 & & \big(\neg\mathsf{Sign}(b,\varphi) \wedge \neg\mathsf{Sign}(c,\varphi) \wedge (b \leqslant_a c)\big) \vee & \\
 & & \big(\neg\mathsf{Sign}(b,\varphi) \wedge \mathsf{Sign}(c,\varphi)\big) & (b, c \in H) \\
[H \Downarrow^a_\varphi](b \leqslant_a c) & \leftrightarrow & \mathsf{Sign}(c,\varphi) \vee \big(\neg\mathsf{Sign}(c,\varphi) \wedge (b \leqslant_a c)\big) & (c \in H, b \in G \setminus H) \\
[H \Downarrow^a_\varphi](b \leqslant_a c) & \leftrightarrow & \neg\mathsf{Sign}(b,\varphi) \wedge (b \leqslant_a c) & (b \in H, c \in G \setminus H) \\
[H \Downarrow^a_\varphi]\neg\psi & \leftrightarrow & \neg[H \Downarrow^a_\varphi]\psi & \\
[H \Downarrow^a_\varphi](\psi_1 \wedge \psi_2) & \leftrightarrow & [H \Downarrow^a_\varphi]\psi_1 \wedge [H \Downarrow^a_\varphi]\psi_2 & \\
[H \Downarrow^a_\varphi]\mathsf{Sign}(b,\psi) & \leftrightarrow & \mathsf{Sign}(b, [H \Downarrow^a_\varphi]\psi) & \\
[H \Downarrow^a_\varphi]\mathsf{Bel}(b,\psi) & \leftrightarrow & \mathsf{Bel}(b, [H \Downarrow^a_\varphi]\psi) &
\end{array}$$

3.3 Private Announcement: Tell-action

This section reviews *tell*-action $[\mathsf{Tell}(b, a, \varphi)]$ in [8]: 'agent b tells to agent a that a certain statement φ is true'. So, we expand our syntax with the operator $[\mathsf{Tell}(b, a, \varphi)]$. An underlying idea of this action is that agent b *privately* tells φ to agent a, that is, the other agents than a will not notice this action. As a result, agent a will change her belief by φ but the other agents than a will not change their beliefs. After the action, agent a will update her belief not only by the statement φ but also by the signed statement $\mathsf{Sign}(b, \varphi)$. In order to capture this private action, we introduce the following special structure (called *action model* in dynamic epistemic logic, the reader may find the similar structure in [9]).

Definition 4. Tell-action model *is* $(E, (D_c)_{c\in G}, (U_a)_{a\in G}, \mathrm{pre})$ *such that* E *consists of two actions: tell-action* $!_b$ *to* a *and non-telling action* $\top$*, and* $D_a = \{(!_b, !_b), (\top, \top)\}$ *and* $D_c = \{(!_b, \top), (\top, \top)\}$ *if* $c \neq a$, $U_c = \{(!_b, \top), (\top, \top)\}$ *for all* $c \in G$, *and* pre *assigns* preconditions *to each action, i.e.,* $\mathrm{pre}(!_b) = \mathsf{Sign}(b, \varphi)$ *and* $\mathrm{pre}(\top) = \top$.

Let us introduce our semantic clause to $[\mathsf{Tell}(b, a, \varphi)]\psi$.

Definition 5. *Given a Kripke model* $\mathfrak{M} = (W, (R_c)_{c\in G}, (S_c)_{c\in G}, (\preccurlyeq_c)_{c\in G}, V)$*, a semantic clause for* $[\mathsf{Tell}(b, a, \varphi)]\psi$ *on* $\mathfrak{M}$ *and* $w \in W$ *is defined as follows.*

$$\mathfrak{M}, w \models [\mathsf{Tell}(b, a, \varphi)]\psi \;\text{ iff }\; \mathfrak{M}^{\mathsf{Tell}(b,a,\varphi)}, (w, !_b) \models \psi,$$

where $\mathfrak{M}^{\mathsf{Tell}(b,a,\varphi)} = (W', (R'_c)_{c\in G}, (S'_c)_{c\in G}, (\preccurlyeq'_c)_{c\in G}, V')$ *is the updated model by Tell-action model of Definition 4, i.e.,*

- $W' := W \times E = W \times \{\, !_b, \top \,\}$.
- $(w, e)R'_c(v, f)$ *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* $c \in G$*).*
- $d \preccurlyeq_c^{\prime(w,e)} d'$ *iff* $d \preccurlyeq_c^{w} d'$.
- $(w, e) \in V'(p)$ *iff* $w \in V(p)$.

Proposition 3 (Recursive Validities [8]). *The following are valid on all models. Moreover, if* ψ *is valid on all models, then* $[\mathsf{Tell}(b, a, \varphi)]\psi$ *is also valid on all models.*

$$\begin{array}{ll}
[\mathsf{Tell}(b, a, \varphi)]p & \leftrightarrow p \\
[\mathsf{Tell}(b, a, \varphi)]d \leqslant_c d' & \leftrightarrow d \leqslant_c d' \\
[\mathsf{Tell}(b, a, \varphi)]\neg\varphi & \leftrightarrow \neg[\mathsf{Tell}(b, a, \varphi)]\varphi \\
[\mathsf{Tell}(b, a, \varphi)](\psi \wedge \theta) & \leftrightarrow [\mathsf{Tell}(b, a, \varphi)]\psi \wedge [\mathsf{Tell}(b, a, \varphi)]\theta \\
[\mathsf{Tell}(b, a, \varphi)]\mathsf{Bel}(a, \psi) & \leftrightarrow \mathsf{Bel}(a, \mathsf{Sign}(b, \varphi) \rightarrow [\mathsf{Tell}(b, a, \varphi)]\psi) \\
[\mathsf{Tell}(b, a, \varphi)]\mathsf{Bel}(c, \psi) & \leftrightarrow \mathsf{Bel}(c, \psi) \quad (a \neq c) \\
[\mathsf{Tell}(b, a, \varphi)]\mathsf{Sign}(c, \psi) & \leftrightarrow \mathsf{Sign}(c, \psi)
\end{array}$$

Proposition 4 (Successful Telling [8]). $\mathsf{Tell}(b, a, \varphi)]\mathsf{Bel}\big(a, \mathsf{Sign}(b, \varphi)\big)$ *is valid in all si-models.*

This prposition is the essential aspect of tell-action. That is, after agent b tells to agent a information φ, agent a believes that agent b signs φ.

4 Dynamic Logical Analysis of Target Legal Case

In this section, we will focus on the inquiring stage and the Civil Court in order to analyze reliability change from the judge's perspective. We will not consider the Appeal Court and the Supreme Court because they only adopted the result of the Civil Court.

4.1 Inquiring Stage

Let us suppose a set of agents $G = \{po, v, f_1, f_2, mo\}$, where we recall that po is an agent of police and v, f_1, f_2, mo are agents of four witnesses. We have four statements involving the legal case as Table 2.

Firstly, we assume that po believes that all witnesses are equally reliable and more reliable than himself because po was not present in the circumstances. So, the reliability ordering in po's perspective can be represented as follows.

$$\mathfrak{M}, w \models \mathsf{Bel}(\; po,\; v \approx_{po} f_1 \approx_{po} f_2 \approx_{po} mo <_{po} po)$$

$p : d$ is the offender.	$r : v$ is punched by an offender.
$q : v$ has a drink at f_2's house.	$u : v$ is stabbed by a hand scraper.

Table 2. Logical Formalization for Statements in Legal Case

In the inquiring stage, four witnesses v, f_1, f_2, mo told the information to po that can be represented by tell-action as follows.

$$I_1 := \mathsf{Tell}\,\big(v, po, p \wedge r \wedge u\big), \qquad I_2 := \mathsf{Tell}\,\big(f_1, po, p \wedge r \wedge u\big),$$
$$I_3 := \mathsf{Tell}\,\big(f_2, po, \neg p \wedge q \wedge \neg r \wedge \neg u\big), \; I_4 := \mathsf{Tell}\,\big(mo, po, p \wedge q \wedge r \wedge u\big).$$

Then, po has to gather all pieces of information from the witnesses. However, po could not know if the witnesses tell the truth or not. po just believes that four witnesses sign the above statements. For example, when v told the statement to po (in I_1), po cannot believe the received statement, i.e., $\mathsf{Bel}(po, p \wedge r \wedge u)$, but po just believes the signed statement of v, i.e., $\mathsf{Bel}(po, \mathsf{Sign}(v, p \wedge r \wedge u))$. Thus, after the above tell-actions, po will believe the following by Proposition 4.

$$\mathfrak{M}, w \models [I_1][I_2][I_3][I_4]\ \mathsf{Bel}\begin{pmatrix} po, \ \mathsf{Sign}\big(v, p \wedge r \wedge u\big) \wedge \mathsf{Sign}\big(f_1, p \wedge r \wedge u\big) \ \wedge \\ \mathsf{Sign}\big(f_2, \neg p \wedge q \wedge \neg r \wedge \neg u\big) \wedge \mathsf{Sign}\big(mo, p \wedge q \wedge r \wedge u\big) \end{pmatrix}$$

4.2 Civil Court

In the Civil Court, a set of agents $G = \{po, v, f_1, f_2, mo, j\}$, where po, v, f_1, f_2, mo are agents of five witnesses, and j is a judge of the Civil Court. Suppose at first, j believes that all witnesses, i.e., po, v, f_1, f_2, mo are equally reliable that can be represented as follows.

$$\mathfrak{M}, w \models \mathsf{Bel}(\ j,\ v \approx_j f_1 \approx_j f_2 \approx_j mo \approx_j po)$$

In the trial, two witnesses v and f_1 told different information to j as follows.

$$T_1 := \mathsf{Tell}\,\big(v, j, \neg p \wedge \neg q \wedge r \wedge \neg u\big), \; T_2 := \mathsf{Tell}\,\big(f_1, j, \neg p \wedge q \wedge r \wedge \neg u\big).$$

Then, po was called to be a witness and told the received information in the inquiring stage to j that can be represented by the following tell-actions.

$$T_3 := \mathsf{Tell}\,\big(po, j, \mathsf{Sign}(v, p \wedge r \wedge u)\big), \qquad T_4 := \mathsf{Tell}\,\big(po, j, \mathsf{Sign}(f_1, p \wedge r \wedge u)\big),$$
$$T_5 := \mathsf{Tell}\,\big(po, j, \mathsf{Sign}(f_2, \neg p \wedge q \wedge \neg r \wedge \neg u)\big), \; T_6 := \mathsf{Tell}\,\big(po, j, \mathsf{Sign}(mo, p \wedge q \wedge r \wedge u)\big).$$

After that, j will believe the following information by Proposition 4.

$$\mathfrak{M}, w \models [T_1][T_2][T_3][T_4][T_5][T_6]\mathsf{Bel}\begin{pmatrix} j, \ \mathsf{Sign}\big(v, \neg p \wedge \neg q \wedge r \wedge \neg u\big) \ \wedge \\ \mathsf{Sign}\big(f_1, \neg p \wedge q \wedge r \wedge \neg u\big) \ \wedge \\ \mathsf{Sign}\big(po, \mathsf{Sign}(v, p \wedge r \wedge u)\big) \ \wedge \\ \mathsf{Sign}\big(po, \mathsf{Sign}(f_1, p \wedge r \wedge u)\big) \ \wedge \\ \mathsf{Sign}\big(po, \mathsf{Sign}(f_2, \neg p \wedge q \wedge \neg r \wedge \neg u)\big) \ \wedge \\ \mathsf{Sign}\big(po, \mathsf{Sign}(mo, p \wedge q \wedge r \wedge u)\big) \end{pmatrix}$$

Based on these pieces of information alone, j cannot decide which pieces of information she should believe. This is firstly because (P1) if j considers the reliability of the information sources, j cannot distinguish the reliability ordering between all witnesses because they are equally reliable. Moreover, (P2) there is contradicting information about p, q, r and u in the signed information from all witnesses. So, j cannot decide which signed information should be in j's belief, i.e., p or $\neg p$, q or $\neg q$, r or $\neg r$, u or $\neg u$. We use the following two ideas: (i) reliability change, and (ii) aggregation policy, to resolve the above problems (P1) and (P2).

(i) Reliability change: The downgrade and upgrade operators of Section 3 are applied in order to simulate the effect of reliability change of the judge in the Civil Court.[3] This allows us to solve the above problem (P1). We also note that, if we apply a framework based on [8], a reliability relation between agents is *fixed*, i.e., the reliability relation between agents cannot be changed.

(ii) Aggregation policy: In [8], Lorini et al. introduced several policies, as meta-logical principles, in order to decide which pieces of information an agent should believe. A common and rational policy is called a *careful policy*, which can be also used to solve the above problem (P2). An idea of this policy is to accept the statements which are universally signed by a group of agents who are equally reliable as beliefs. Firstly, we define $\mathsf{Sign}(\mathsf{C}_i^a, \varphi)$ which stands for 'all agents who are in the set C_i^a sign statement φ' as follows:

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

Then, we can describe the careful policy in the following formal definition.

$$\mathsf{Careful}(a, \varphi) =_{def} \bigvee_{i \leq M} \Bigg(\mathsf{Bel}\big(a, \mathsf{Sign}(\mathsf{C}_i^a, \varphi)\big) \wedge \mathsf{Bel}\big(a, \bigwedge_{1 \leq j \leq i-1} \neg \mathsf{Sign}(\mathsf{C}_j^a, \neg\varphi)\big) \Bigg) \rightarrow \mathsf{Bel}(a, \varphi),$$

where M is the maximum rank, i.e., the maximum natural number of $\{i \leq \#G \mid \mathsf{C}_i^a \neq \emptyset\}$, and $\mathsf{Careful}(a, \varphi)$ stands for 'agent a aggregates information about φ by the careful policy'. The agent a will believe statement φ if (i) agent a believes that statement φ is universally signed by a group of agents C_i^a, and (ii) agent a believes that statement $\neg\varphi$ is not signed by all other agents which are more reliable than agents in C_i^a.

Now let us apply our two ideas to disolve the judge's difficulty of deciding which pieces of information she should believe. In what follows, we assume that the judge j in the Civil Court. From Section 2, j believes that the signed information of v and f_1 in the Civil Court is less reliable than that in the inquiring stage. Thus, we can regard that j changes her reliability ordering of v and f_1.

Focusing on the Civil Court, j believes that v and f_1 are less reliable. From the tell-actions $T_1 - T_6$, there is conflicting information about p and u. That is, both v and f_1

[3] In this work, we will not analyze how an agent decides to change the reliability ordering between the other agents.

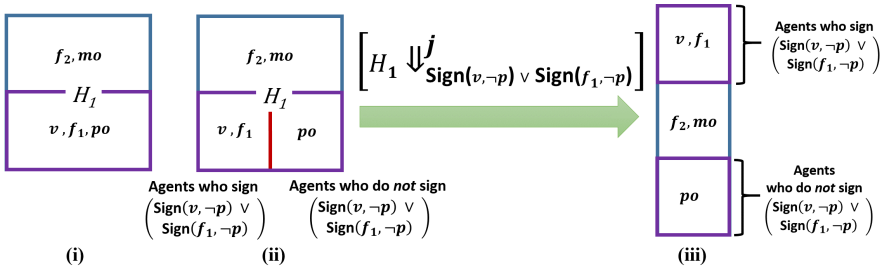


Fig. 2. Downgrading by $[H_1 \Downarrow^{j}_{\mathsf{Sign}(v,\neg p) \vee \mathsf{Sign}(f_1,\neg p)}]$

told statements $\neg p$ and $\neg u$ (in T_1 and T_2), while po told signed statement of v and f_1 p and u (in T_3 and T_4). Since p is an important statement providing the information who is the offender, j will focus on the statement p. Therefore, j downgrades all agents in H_1 who sign statement $\mathsf{Sign}(v, \neg p) \vee \mathsf{Sign}(f_1, \neg p)$ by $[H_1 \Downarrow^{j}_{\mathsf{Sign}(v,\neg p) \vee \mathsf{Sign}(f_1,\neg p)}]$. Let us see a process of the downgrade step by step (Fig. 2). We first define H_1 by $\{v, f_1, po\}$ which is a set of agents who told information in the Civil Court (see Fig. 2(i)). Then, when we consider the agents who sign statement 'agent v signs information $\neg p$ or agent f_1 signs information $\neg p$', i.e., $\mathsf{Sign}(v, \neg p) \vee \mathsf{Sign}(f_1, \neg p)$, H_1 is divided into two equal vertical parts, i.e., $\mathsf{Sign}\big(x, \mathsf{Sign}(v, \neg p) \vee \mathsf{Sign}(f_1, \neg p)\big)$ and $\neg\mathsf{Sign}\big(x, \mathsf{Sign}(v, \neg p) \vee \mathsf{Sign}(f_1, \neg p)\big)$ as Fig. 2(ii). Next, j downgrades all agents in H_1 who sign $\mathsf{Sign}(v, \neg p) \vee \mathsf{Sign}(f_1, \neg p)$, and the result can be shown as Fig. 2(iii). That is, agents v and f_1 are less reliable than the other agents. We assume that the agents who are in the same set are equally reliable. After that, j changes her belief about the reliability ordering between all witnesses as follows.

$$\mathfrak{M}, w \models [T_1][T_2][T_3][T_4][T_5][T_6]\ \mathsf{Bel}(\ j,\ po <_j f_2 \approx_j mo <_j v \approx_j f_1)$$

By a careful policy, j aggregates information and will believes as follows.

$$\mathfrak{M}, w \models [T_1][T_2][T_3][T_4][T_5][T_6]$$
$$\mathsf{Bel}\begin{pmatrix} j,\ \mathsf{Sign}\big(v, p \wedge r \wedge u\big) \wedge \mathsf{Sign}\big(f_1, p \wedge r \wedge u\big)\ \wedge \\ \mathsf{Sign}\big(f_2, \neg p \wedge q \wedge \neg r \wedge \neg u\big) \wedge \mathsf{Sign}\big(mo, p \wedge q \wedge r \wedge u\big) \end{pmatrix}$$

From the above result, we can regard that j believes the signed information of witnesses in the inquiring stage. When j focuses on the inquiring stage, j believes that v and f_1 are more reliable. Thus, j upgrades all agents who sign statement $p \wedge r \wedge u$ by $[H_2 \Uparrow^{j}_{p \wedge r \wedge u}]$. At first, we regard Fig. 3(i) which is the result of the downgrade as a model of the beginning of this upgrade. Since j focuses on the inquiring stage, a new set H_2 of agents is defined by $\{v, f_1, f_2, mo\}$ as Fig. 3(ii). Then, when we consider the statement $p \wedge r \wedge u$, H_2 is divided into two equal vertical parts, i.e., $\mathsf{Sign}(x, p \wedge r \wedge u)$ and $\neg\mathsf{Sign}(x, p \wedge r \wedge u)$ as shown in Fig. 3(iii). By $[H_2 \Uparrow^{j}_{p \wedge r \wedge u}]$, agents v, f_1 and mo who sign statement $p \wedge r \wedge u$ are upgraded to be more reliable than the other agents as shown in Fig. 3(iv). Consequently, j changes her reliability ordering between all witnesses as

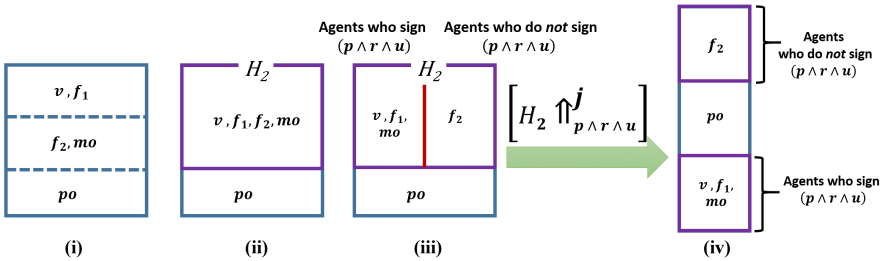


Fig. 3. Upgrading by $[H_2 \Uparrow^j_{p \wedge r \wedge u}]$

follows.

$$\mathfrak{M}, w \models [T_1][T_2][T_3][T_4][T_5][T_6]\ \mathsf{Bel}(\ j,\ v \approx_j f_1 \approx_j mo <_j po <_j f_2)$$

After the upgrade, j aggregates information by the careful policy again and will believe that d is the offender (p), v is punched by the offender (r), and v is stabbed by a hand scraper (u) as follows.

$$\mathfrak{M}, w \models [T_1][T_2][T_3][T_4][T_5][T_6]\ \mathsf{Bel}(\ j, p \wedge r \wedge u)$$

5 Conclusion

This work has proposed logical analysis for formalizing reliability change of a judge in a court. We introduced two dynamic operators: downgrading $[H \Downarrow^a_\varphi]$ and upgrading $[H \Uparrow^a_\varphi]$. The first operator downgrades all agents in H who sign φ, while the second operator upgrades them. Based on these operators, we have formalized an example of legal case in Thailand. In the trials, the judge first believed that all witnesses are equally reliable. Then, the judge changed her belief about the reliability ordering between witnesses. We can successfully analyze this process by downgrading and upgrading the reliability of the witnesses. Our contribution is to formalize the change of the reliability ordering between the other agents depending on an agent's perspective.

In this work, we only capture the effect of reliability change on belief change, i.e., when a judge changed her reliability ordering between some witnesses, she may change her beliefs about information from those witnesses. On the other hand, belief change may affect reliability change. For example, consider a legal case consisting of several accused persons. At first, a judge cannot decide who is the defendant. Then, when the judge received new information, she may change her belief about some accused persons. After that, the judge may change her reliability ordering between accused persons. Our work only supposes that the judge changes the reliability based on her belief change. Therefore, we plan to formalize the effect of belief change on reliability change by applying the notion of preference upgrade in [10]. Furthermore, this work only formalize the reliability of agents, but does not consider the reliability of statements. That is, this work assumes that when agent a received a statement φ from agent b, agent a has already decided that the statement φ is reliable or not. If agent a considers that the

statement φ is not reliable, then she believes that agent b who gives the statement φ will be unreliable. However, we can analyze such reliability change of statements by employing a preference modality based on [10].

References

1. Prakken, H., Sartor, G.: The role of logic in computational models of legal argument - a critical survey. In: Computational Logic: Logic Programming and Beyond. Volume 2408 of Lecture Notes in Computer Science. (2001) 342–381
2. Bench-Capon, T.J.M., Prakken, H.: Introducing the logic and law corner. J. Log. Comput. **18**(1) (2008) 1–12
3. Grossi, D., Rotolo, A.: Logic in the law: A concise overview. Logic and Philosophy Today, Studies in Logic **30** (2011) 251–274
4. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer (2008)
5. van Benthem, J.: Dynamic logic for belief revision. Journal of Applied Non-Classic Logics **14**(2) (2004) 129–155
6. Liau, C.J.: Belief, information acquisition, and trust in multi-agent systemsa modal logic formulation. Artificial Intelligence **149**(1) (2003) 31 – 60
7. Perrussel, L., Lorini, E., Thévenin, J.: From signed information to belief in multi-agent systems. In: Proceedings of The Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2010), Lyon, France, August 30 - September 2, 2010. (2010)
8. Lorini, E., Perrussel, L., Thévenin, J.: A modal framework for relating belief and signed information. In: Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17-18, 2011. Proceedings. (2011) 58–73
9. 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
10. van Benthem, J., Liu, F.: Dynamic logic of preference upgrade. Journal of Applied Non-Classical Logics **17**(2) (2007) 157–182

Module Based Argumentation Framework for Analysis of Actual Discussion Records

Kei Nishina[1], Shogo Okada[1], and Katsumi Nitta[1]

[1] Department of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology

Abstract. To analyze real complex discussion records containing a lot of arguments, theory of argumentation framework isn't sufficient because it lacks in ability of dividing a large argument structure into several smaller structures. In this paper, we showed the motivation of introducing a module structure, and then, we proposed Module Based Argumentation Framework (MBAF) which regards an argumentation framework as a set of modules. In each module, a smaller argumentation framework exists. We showed the definition of MBAF's module structure, and defined the MBAF semantics which is an integrated semantics of more than one modules.

1 Introduction

Discussions are conducted by exchanging participants' arguments of different opinions. It is important for participants to judge which arguments are accepted or rejected after the discussion is over. One of the ways to judge them is to use a theory of mathematical argumentation such as Abstract Argumentation Framework (AF) [1]. AF is a useful tool to analyze the argument structure of a discussion and to calculate its semantics.

Based on the AF theory, we have developed a discussion support tool by which a discussion record is transformed into a set of logical formula, then the logical formula are translated into an AF structure and the conclusion of the discussion is obtained by AF semantics. Using this tool, we have analyzed several negotiation records from the Inter Collegiate Negotiation Competition. As the negotiation is conducted for a long time (up to 3 hours), and about 10 issues are argued and more than 70 arguments are included in the record.

By this experience, we found that actual complex discussion records have the following properties. Firstly, some of the arguments

may be based on unreliable assumptions. In such case, concerning the reliability of assumptions, another discussion records may be referred to. Secondly, topics of the discussion may consist of several issues from different domains such as engineering, economics, ecology and so on. Thirdly, when a discussion is of a complex nature, a lot of arguments appear and the argument structure becomes too complex to recognize the key arguments which affect the final outcome of the discussion.

Therefore, when we analyze such discussion records by AF, following problems occurs:

(i) AF has no way to represent the reliability of an argument directly. Therefore, to show the reliability of an argument, another AF which decides the reliability of the argument may be referred. However, the current AF theory doesn't have semantics which treats two AFs at once.

(ii) AF has no way to discriminate arguments of a specific domain from those of other domains. Therefore, arguments from different domains appear in one AF which reduces the clarity of the structure.

(iii) As AF represents the structure of arguments in the form of a simple graph structure, if the discussion contains a lot of issues, the graph becomes too large to grasp which argument affects the final conclusion.

To solve these problems, employing a module structure to AF is a promising consideration. In previous works of AF, Modular Assumption Based Argumentation (MABA) [2] and hierarchical Extended Argumentation Framework (EAF) [3] have a module structure. However, MABA doesn't have semantics which uses two AFs at once, and hierarchical EAF employs extended attack mechanism to introduce the concept of preference among arguments which doesn't solve above problems of AF.

Therefore, the objective of our research is to develop a novel AF theory which provides semantics of two AFs at once and treats a huge graph structure as a set of sub graphs. In this paper, we introduce Module Based Argumentation Framework (MBAF) which employs the module structure, and shows how the above problems are solved in MBAF.

In Section 2, the motivation of employing a module structure is shown using a simple example. In Section 3, the concept of MBAF is introduced, and in Section 4, properties of MBAF are described. And in

Section 5, the conclusion of this research is outlined.

2 Motivation of Employing Module Structure

In this paper, we show why we employ module structure in AF using a simple example case. Figure 1 shows the AF structure of the discussion about restart of nuclear reactors in Japan. There are 9 arguments {A, B, C, …, I} and 11 attacks { A→B, B→A, C→A, …, I→E }, and they forms a graph structure. In AF theory, a set of arguments {*D*, *H*, *I*} is the ground extension, and a set {*A*, *F*} is one of complete extensions.

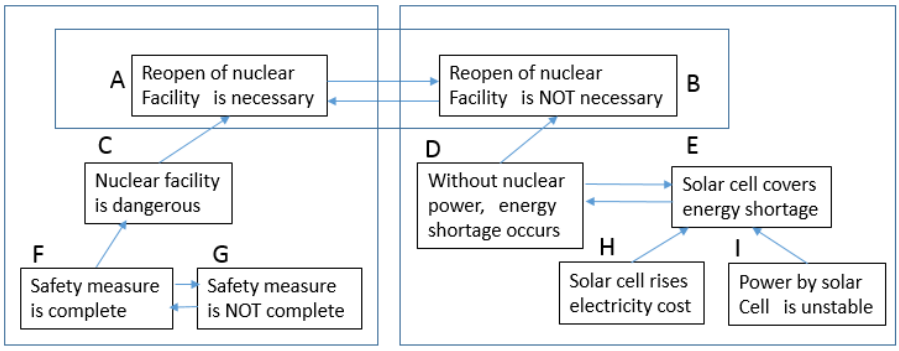


Fig. 1. An example of AF represents real complex debates

This example contains three kinds of discussion – industry policy issue {A, B}, security issue {C, F, G} and economy issue {D, E, H, I}. The conclusion of discussion of security issue affects the reliability of A, and the conclusion of discussion of economy issue affects the reliability of B. Furthermore, this example shows the conclusion of discussion of security issue and that of economy issue don't affect each other.

This information is represented by three boxes which surround {A, B}, {A, C, F, G} and {B, D, E, H, I}. When the graph is huge, this box notation is useful to divide a huge graph into small graphs for each issue. However, the original AF theory doesn't have a function of such box notation. Therefore, we employed module structure to AF theory, and constructed Module Based Argumentation Framework (MBAF).
In MBAF, we think AF structure in Figure 1 consists of three modules – M_1, M_2 and M_3. Followings are arguments and attack relations of each

module.

M_1: Arguments = {A, B}
attacks = {(A, B), (B, A)}
M_2: Arguments = {A, C, F, G}
attacks = {(C, A), (F, C), (F, G), (G, F)}
M_3: Arguments = {B, D, E, H, I}
attacks = {(D, B), (D, E), (E, D), (H, E), (I, E)}

An argument A is contained in two modules, M_1 and M_2, and an argument B is contained in two modules in M1 and M_3. We think M_1 is higher than M_2 and M_3 because M_1 discusses the main issue and M_2 and M_3 discuss sub issues which decide the reliability of arguments of main issue. Therefore, the semantics of total AF may be obtained by calculating semantics of M_2 and M_3 at first and then by calculating that of M_1.

3 Definition of Module Based Argumentation Framework

A Module Based Argumentation Framework consists of Module Structure and a set of Modular Argumentation Frameworks.

Definition 1. Module Based Argumentation Framework
A module based argumentation framework is 2-tuple (*MS*, *MAFs*), where *MS* is a module structure, and *MAFs* is a set of Modular Argumentation Frameworks.

3.1 Definition of MS

A Module Structure consists of a set of modules and access relation among them.

Definition 2. Module Structure
Module Structure is a 2-tuple, (*Modules*, *Access*), where *Modules* is a set of modules and *Access* is a set of hierarchical relations between two modules. If (M, M') $\in$*Access*, module M is immediately higher than module M', which means the reliability of some arguments in M is decided by the semantics of argumentation framework in M'. We assume that *MS* forms an acyclic graph.

Furthermore, the function *Ihm* and the function *Ilm* are defined as follows.

For a module M $\in$*Modules*, *Ihm*(*M*) $\equiv$$\{M_i | (M_i, M) \in Access\}$, and *Ilm*(M) = $\{ M_i | (M, M_i) \in Access \}$.

Figure 2 shows an example of Module Structure. In this figure, *MS*= $(\{M_0, M_1, M_2, M_3\}, \{(M_0, M_1), (M_0, M_2), (M_2, M_3)\})$. and *Ihm*($M_0$) = $\emptyset$ (M_0 is a top module of MS), $Ihm(M_1) = Ihm(M_2) = \{M_0\}$, $Ihm(M_3) = \{M_2\}$, $Ilm(M_0) = \{M_1, M_2\}$, $Ilm(M_1) = Ilm(M_3) = \emptyset$ (M_1 and M_3 are bottom modules of *MS*), and $Ilm(M_2) = M_3$.

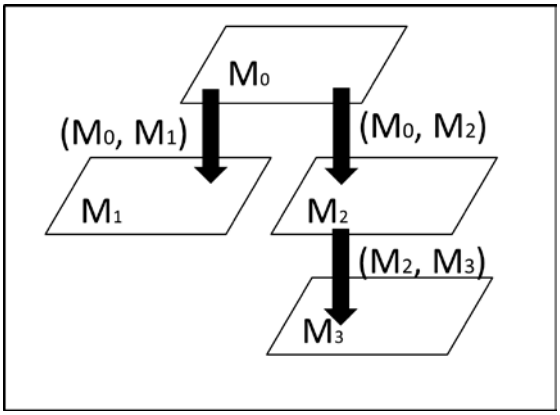


Fig. 2. Structure of MS

3.2 Definition of MAF

In each module, as there are arguments and attack relations, an AF is constructed. Instead of original AF, we introduce a concept of "Modular Argumentation Framework (MAF)," because the semantics of an AF of a module M is affected by semantics of AF of *Ilm*(M).

Definition 3. Modular Argumentation Framework (MAF)
MAF is a 4-tuple *MAF* = (*Ar, attacks*, *Status*, *ST*). *Ar* is a set of arguments, and *attacks* is a set of attack relation which satisfiy *attacks* $\subseteq$ *Ar*$\times$*Ar. Status* is a set of statuses of reliabilities. In this paper, we

define *Status* = {*sk, cr, unc, def*} where *sk* means skeptical reliability, and *cr* means credulous reliability, *unc* means uncertain, and *def* means defeated.

ST is the function which allocate Status to each argument: $Ar \rightarrow Status$. *How to decide ST* is described in section 3.4.

If a MAF is in the lowest module M_l, which means $Ilm(M_l) = \emptyset$, to all arguments, a status label *sk* is allocated. If a MAF is not in the lowest module, module M_i, status of arguments is defined by the MBAF semantics of $Ilm(M_i)$.

3.3 MAF semantics

Let $MAFi = (Ar_i, att_i, Status_i, ST_i)$ is MAF in a module M_i. The extensions in the MAF semantics are defined using the notion of labelling just as described in [4], [5], and [6].

Definition 4. *MAF Labelling*
A MAF labelling in a module M_i, L_i is a total function.

L_i: $Ar_i \rightarrow \{in, out, undec\}$,
$in(L_i) = \{A \mid L_i(A) = in\}$, $out(L_i) = \{A \mid L_i(A) = out\}$,
$undec(L_i) = \{A \mid L_i(A) = undec\}$.

Definition 5. *Conflict-free MAF Labelling*
L_i is a conflict-free labelling if no *in*-labelled argument attacks an (other or the same) *in*-labelled argument.

Let $cMAF_i labelling$ is a set of all *conflict-free* MAF_i *labelling*, let *cMAFset* be the set of all conflict-free sets in MAF_i, then a function $MAF_i Lab2Ext$ is defined as follows.

$MAF_i Lab2Ext : MAF_i clabelling \rightarrow MAF_i csets$
such that $MAF_i\, Lab2Ext(L_i) = in(L_i)$

Definition 6. *MAF Complete Labelling*
L_i is a MAF_i *complete labelling* iff for each argument $A \in Ar_i$ it holds that:

(i) If $ST(A) = sk$ and A isn't attacked by any argument in MAF_i then

A must be labelled *in*.

(ii) If *ST*(A) = *cr*, then A must be labelled *in* or *out* or *undec*.

(iii) If *ST*(A) = *unc*, then A must be labelled *undec*.

(iv) If *ST*(A) = *def*, then A must be labelled *out*.

Furthermore, each argument which doesn't correspond to any of the condition (i), (ii), (iii) and (iv) follows any one of the followings.

(v) If argument A is labelled *in*, then all its attackers are labelled *out*.

(vi) If argument A is labelled *out*, then it has at least one attacker that is labelled *in*.

(vii) If argument A is labelled *undec*, then it has at least one attacker that is labelled *undec* and it does not have an attacker that is labelled *in*.

If L_i is a MAF_i *complete labelling*, MAF_i *Lab2Ext*(L_i) is always a member of MAF_i *complete extension*.

Definition 7. *MAF Stable Labelling*

A labelling L_i is a MAF_i *stable labelling* iff L_i is a MAF_i *complete labelling* and *undec*(L_i) = ∅.

If L_i is a MAF_i *stable labelling*, MAF_i *Lab2Ext*(L_i) is always a member of MAF_i *stable extension*.

Definition 8. *MAF Grounded Labelling*

The following statements are equivalent:

L_i is the MAF_i *grouded labelling*.

L_i is a *strict* MAF_i *complete labelling* where *in*(L_i) is minimal (w.r.t. set inclusion) among all *skeptical* MAF_i *complete labelling*.

L_i is a *strict* MAF_i *complete labelling* where *out*(L_i) is minimal (w.r.t. set inclusion) among all *skeptical* MAF_i *complete labelling*.

L_i is a *strict* MAF_i *complete labelling* where *undec*(L_i) is maximal (w.r.t. set inclusion) among all *skeptical* MAF_i *complete labelling*.

If L_i is unique MAF_i *grounded labelling*, MAF_i *Lab2Ext*(L_i) is always

unique MAF_i *grounded extension*.

Definition 9. *Strict MAF Complete Labelling*
L_i is a s*trict* MAF_i *complete labelling* iff for each argument $A \in Ar_i$, it holds that:

(i) If *ST*(A) = *s*k and A isn't attacked by any argument in MAF_i, then A must be labelled *in*.
(ii) If *ST*(A) = *cr*, then A must be labelled *out* or *undec*.
(iii) If *ST*(A) = *unc*, then A must be labelled *undec*.
(iv) If *ST*(A) = *def*, then A must be labelled *out*.

Furthermore, each argument which doesn't correspond to (i), (ii), (iii) and (iv) follows any one of (v), (vi) and (vii) in Definition 6.

Definition 10. *MAF Preferred Labelling*
The following statements are equivalent:

L_i is a MAF_i *preferred labelling*.

L_i is a MAF_i *complete labelling* where *in*(L_i) is maximal (w.r.t. set inclusion) among all MAF_i *complete labelling*.

L_i is a MAF_i *complete labelling* where *out*(L_i) is maximal (w.r.t. set inclusion) among all MAF_i *complete labelling*.

If L_i is a MAF_i *preferred labelling*, MAF_i *Lab2Ext*(L_i) is always a member of MAF_i *preferred extension*. Therefore, MAF_i semantics is defined by MAF_i *labellings* and reliability of arguments defined by ST_i. Then, we will explain of the detail of *ST*.

3.4 Reliability of argument in MAF

ST of MAF assigns the degrees of reliability to each argument in Ar of MAF.

Definition 11. *ST*
Let $MAF_i = (Ar_i, attacks_i, Status_i, ST_i)$ in a module M_i., and let $A \in Ar_i$, then *ST*(A) is defined as follows.

(1) *ST*(A) = *sk*

ST(A) = *sk* holds iff either of the following holds for every module Mj in *Ilm*(M_i).

(i) A is a member of *MAFgrounded extension* of Mj.

(ii) A is not a member of *Arj* .

(2) *ST*(A) = *def*

ST(A) = *def* holds iff the both of the following holds for at least one module Mj in *Ilm*(M_i).

(iii) A isn't a member of *MAFcomplete extension* of Mj.

(iv) There is at least one *MAF complete labelling* of Mj in which A is labelled *out*.

(3) *ST*(A) = *unc*

ST(A) = *def* holds iff the both of the following holds for at least one module Mj in *Ilm*(M_i).

(v) A isn't a member of *MAFcomplete extension of* Mj.

(vi) A is labelled *undec* in every *MAF complete labelling* of Mj.

(4) *ST*(A) = *cr*

ST(A) = *cr* holds iff the all of the following holds for every module Mj in *Ilm*(M_i).

(vii)A is a member of *MAFcomplete extension* of Mj.

(viii)A isn't a member of *MAFgrounded extension* of Mj.

.

5. Property of MBAF

If an AF is divided into two modules, there is a close relation between semantics of AF and those of MAF of module as follows.

Let $AF = (Ar, attacks)$ is divided into two modules Mi and Mj which satisfy {Mj}=Ilb(Mi). And let $MAF_i = (Ar_i, attacks_i, Status_{, STi})$ and $MAF_j = (Ar_j, attacks_j, Status_{, STj})$ which satisfy following conditions.

(i) $(Ar_i \cup Ar_j) = Ar$

(ii) $(attacks_i \cup attacks_j) = attacks \ \wedge \ (attacks_i \cap attacks_j) = \emptyset$

(iii) For any $(A, B) \in attacks$, {A,B} isn't a subset of $(Ar_i \cap Ar_j)$.

(iv) For any (A, B)$\in$*attacks*, (A$\in Ar_i \wedge$ B$\in Ar_i$)$\vee$(A$\in Ar_j \wedge$ B$\in Ar_j$) holds.

Theorem

If argument A$\in$*Ari* is a member of an *AF complete extension* S in the *AF*, and if (S$\cap$Arj) is an *AF complete extension* in *AFj=(Arj, attacksj)*, then A is a member of *MAF$_i$ complete extension*.

Proof

Let $Ar_{ij} = (Ar_i \cap Ar_j)$. As A is a member of an AF complete extension, for two arguments B and C which satisfy (B,A)$\in$attacks and (C,A)$\in$ attacks, there is a labelling such that *L*(B)=*out* and *L*(C)=*in*.

For A, B and C, we consider 8 cases.

(1) A$\in$*Ari\Arij*, B$\in$*Ari\Arij*, C$\in$*Ari\Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk*, *ST*(B)=*sk*, *ST*(C)=*sk*

(2) A$\in$*Ari\Arij*, B$\in$*Ari\Arij*, C$\in$*Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk*, *ST*(B)=*sk*, *ST*(C)=*sk/cr*

(3) A$\in$*Ari\Arij*, B$\in$*Arij*, C$\in$*Ari\Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk*, *ST*(B)=*cr/def*, *ST*(C)=*sk*

(4) A$\in$*Ari\Arij*, B$\in$*Arij*, C$\in$*Arj\Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk*, *ST*(B)=*cr/def*

(5) A$\in$*Arij*, B$\in$*Ari\Arij*, C$\in$*Ari\Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk/cr*, *ST*(B)=*sk*, *ST*(C)=*sk*

(6) A$\in$*Arij*, B$\in$*Ari\Arij*, C$\in$*Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk/cr*, *ST*(B)=*sk*, *ST*(C)=*sk/cr*

(7) A$\in$*Arij*, B$\in$*Arj*, C$\in$*Arj*

In this case, the function *ST* takes following values.

ST(A)=*sk*

(8) A$\in$*Arij*, B$\in$*Arj*, C$\in$*Arij*

In this case, the function *ST* takes following values.

ST(A)=*sk/cr*, *ST*(C)=*sk/cr*

In any case, there is a MAF labelling L(A)=*in*, L(B)=*out* and L(C)=*in.* Therefore, A is a member of MAF_i *complete extension.*

6 Conclusion

To analyze actual discussion records, we introduced a module structure to the theory of argumentation framework, and proposed a Module Based Argumentation Framework (MBAF). This framework is useful when the discussion contains several hard issues and a lot of arguments are contained in it. MBAF divides the huge arguments structure into several smaller ones, and integrates semantics of arguments of modules. We showed the definition of MBFA semantics using Caminada's labelling method, and showed the relation between AF semantics and MBAF semantics.

Acknowledgement
We acknowledge the financial support of AOARD (Asian Office of Aerospace Research and Development).

References

1. Phan Minh Dung.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77:321-357, (1995)
2. Phan Minh Dung.: Modular argumentation for modelling legal doctrines of performance relief, Argument and Computation, Taylor & Francis, (2010).
3. Sanjay Modgil.: Reasoning about preferences in argumentation frameworks. Artificial Intelligence 173, ELSEVIER, (2009).
4. M.W.A. Caminada.: On the issue of reinstatement in argumentation. Technical Report UU-CS-2006-023, Institute of Information and Computing Sciences, Utrecht University, (2006).
5. Y. Wu, M.W.A. Caminada and Dov Gabbay.: Complete Extensions in Argumentation Coincide with 3-Valued Stable Models in Logic Programming. Studia Logica 93(2-3):383-403, (2009).
6. Martin Caminada and Dov Gabbay.: A Logical Account of Formal Argumentation, Paper 347, May (2009).
7. T. J. M. Bench-Capon.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation, 13(3):429–448, (2003).
8. T. J. M. Bench-Capon.: Value-based argumentation frameworks. In: Proceedings of NonMonotonic Reasoning, pp. 444–453, (2002).

Quantum Artificial Intelligence for Modelling Legal Knowledge of Legal Definitions: Cases from EU Water Energy and Food Legislations and Nexus *

Md Mizanur Rahman

Erasmus Mundus Doctoral Fellow, LAST-JD program, CIRSFID, University of Bologna, mdmizanur.rahman2@unibo.it and mizanur3@gmail.com

Abstract. This paper proposes an initial idea of how to model legal knowledge that appears from legal definitions and their multiple interpretations by quantum artificial intelligence using an example of EU water, energy and food legislations and nexus. At first, it shows the critical legal reasoning problem of Water-Energy-Food nexus and demonstrates of why and how legal definitions play the most important role in the legal reasoning process. Then it presents the fundamental structural components of a legal definition and how applicability effects may take in place by making any change in any component of a legal definition. It used Article 2(1) of the Water Directive 98/83/EC as an example to show different structural components of a legal definition. Then it proposed an extended version of State-Context-Property Systems, known as SCOPs for quantum formalism for a concept, for quantum formalism of a legal definition with its basic mathematical function.

Keywords: Quantum Artificial Intelligence. Quantum Formalism of a Legal Definition. SCOPs Quantum Formalism of a Concept. Structural Component of a Legal Definition, Water Energy Food Nexus. AI & Law, EU Legislation.

1 Introduction

Since the beginning of twenty first century, on one hand, policy makers of all countries have been facing comprehensive challenges coming from, one of the most dominating public policy and legal reasoning discourse, water-energy-food (WEF) nexus such as detecting WEF nexus in real time, space and legal setting [1][2][3][4][5]. Quantum artificial intelligence, on the other hand, a newly emerging branch of artificial intelligence, provides a new array of computational potentialities such as formalism of many-world interpretations within polynomial time performance and the capacity of measuring contextual typicality and applicability of different properties coming from combined-concepts etc [6][7]. Therefore, applying quantum artificial intelligence in order to solve those problems that exist in WEF nexus might bring new opportunities for

* This paper is a product of a doctoral project in Erasmus Mundus Joint International Doctoral Degree in Law, Science and Technology, coordinated by CIRSFID, University of Bologna, and submitted to both conferences – JURIX 2014 and JURISIN 2014.

policy makers for formulating various products of public policies in more efficient and effective way. In order to achieve this goal, one of the fundamental requirements is to represent a set of legal definitions and their corresponding constitutive and regulative rules in a complete and sound form that includes to represent all structural components of a legal definition such as concepts, properties, contexts, constitutive and regulative rules and their associated legal purposes and agents etc., in the Hilbert Space. Hence, the paper first briefly presents the critical issues in the WEF nexus from legal reasoning perspective and importance of a legal definitions in the legal reasoning process. Then it proposes an extended from of state-context-property formalism (SCOPs) [8], as a generalization of quantum formalism, for representing various legal definitions including their inherent constitutive and regulative rules in a holistic way.

The following definitional clarity might help readers to understand the paper easily – (a) WEFC domains represents water, energy, food and climate domain separately, (b) WEFC nexus means the entangled[1] point of WEFC domains, (c) constitutive rule means the rule of meaning-making depended on other corresponding rule or definition, (d) Composite constitutive rule, in WEFC nexus context, means the entangled point of all available constitutive rules coming from WEFC domains.

1. Critical Issues in WEF Nexus: From Legal Definitions to Legal Reasoning

Fundamentally WEF domains are, on one hand, deterministic due to their sectorial, that is based on legal and structural governances, biasedness and, on the other hand, non-deterministic due to the unknown consequences originated from each of their legal biasedness over the others [9][10][11]. These deterministic and non-deterministic natures of WEF nexus, in combined, generate complexities in the legal reasoning process of WEF nexus. Because legal reasoning works like a complete circuit composed by legal rules as the point of departure from the source to the proof and supported by evidences throughout the proofing process back to the source. In the case of WEF domains, this legal reasoning [12][13][14] circuit does not function completely. Because the most, if not all, of institutional rules of WEFC domains are driven by the legal definitions and constitutive and regulative rules that appear from such sets of legal definitions those are by character limitedly expressive and operated by the respective authorities of WEFC domains. This limited expressiveness of legal definitions coming from one domain is further compressed and challenged by the deterministic and non-deterministic natures of the WEFC domains. Consequently legal reasoning process of WEFC nexus becomes paralyzed at some point of the legal reasoning circuits by their own legal biasedness.

The legal definitions of water, food and biofuel have considered for following explanation from – (a) EU Water Directive 98/83/EC[2] on the quality of water intended for human consumption, (b) Regulation no 178/2002[3] of the European Parliament and

[1] Entangled state is known as not decomposable sate. See Timpson C.G. 2004. Quantum information theory and the foundation of quantum mechanics. Oxford: University of Oxford. 126 p. and Wichert A. 2014. Principles of quantum artificial intelligence. Singapore: World Scientific Publishing Co. 126 p.

[2] http://eur-lex.europa.eu/legalcontent/EN/ALL/?uri=CELEX:31998L0083

[3] http://eur-lex.europa.eu/legalcontent/EN/ALL/?uri=CELEX:32002R0178

of the Council on food safety, (c) EU Directives 2003/30/EC[4] on Promotion of the use of biofuel or other renewable fuels for transport. On the basis of above legal definitions, legal reasoning mechanism of WEFC nexus is shown in Figure 1 and briefly explained below -

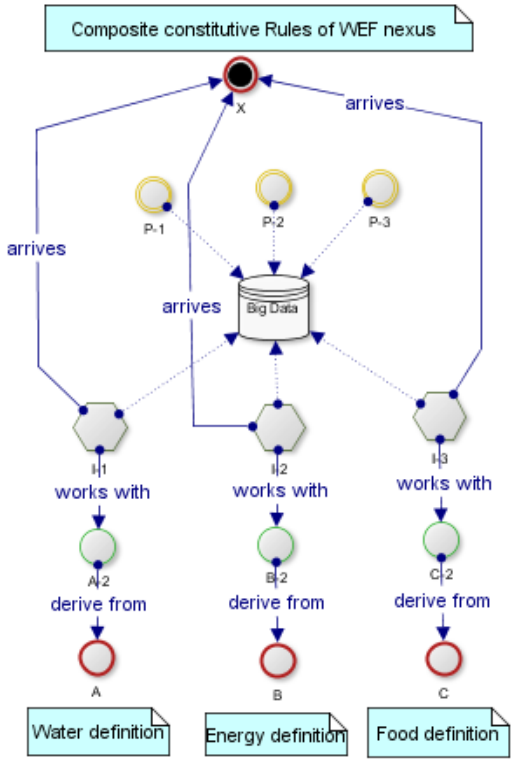


Figure 1. Legal reasoning framework of WEF nexus [from legal definition to legal reasoning]

- A, B and C represents legal definitions of water, energy and food respectively and each of them represents non-interacting system or interacting system in a very limited way.
- A2, B2 and C2 are corresponding constitutive rules of WEFC domains coming from respective legal definitions of A, B and C. These constitutive rules also represent non-interacting system.
- I-1, I-2 and I-3 are legally accountable institutions responsible to implement respective constitutive rules A2, B2 and C2. By nature, there institutions are non-interacting systems too.
- A, B and C; A2, B2 and C2; and I-1, I-2 and I-3 are not a composite system. Consequently, they do not themselves produce composite constitutive rules.

[4] ec.**europa.eu**/energy/res/legislation/doc/**biofuels**/en_final.pdf

- Institutions I-1, I-2 and I-3 are legally responsible to produce their own scientific evaluative report corresponding their own domains.
- All available knowledge coming from satellite, environmental technologies and social web related with WEF domains are known as Big Data of WEF domains. These Big Data contains the evidence that is legally admissible for proofing legal reasoning of WEF nexus.
- In order to realize composite constitutive rules, P1, P2 and P3 need to be applied over the Big Data. P1, P2 and P3 represents precautionary principles, coherence of law and utility of law respectively.
- As the content of Big Data is unknown to anyone, the answer of any queries asked by applying P1, P2 and P3 is unknown too to everyone.
- Appling P1, P2 and P3 over the Big Data is intended to produce composite constitutive rules X.
- Once X as composite constitutive rules will appear, legal definition A, B and C may lose the reasons for their own legitimacy even though they might have their own legal validity. This state can also be addressed as entangled state of WEFC nexus.

Like above scenario of WEF nexus, one of the departure points of all legal reasoning is the legal definition, which is static in its legal nature. Therefore, it is an indispensable step to investigate the content of a legal definition. Mainly following questions – what are the structural components of a legal definition and how does the change in any structural component affects the other structural components of the same legal definition? The answers of these questions have discussed below in brief.

2. Content and Structure of a Legal Definition

In the present state-of-art of legal philosophy, the structure of a legal definition is still unrecognized and undisclosed research area. In this section, however, a general structure of a legal definition has been drawn with the example of 'water intended for human consumption' of Water Directive 98/83/EC, shown in table 1.

2.1. The Major Structural Components of a Legal Definition

Generally a legal definition may have – *concepts* that is a legal entity and can be found in the different states of the definition, *state* of a concept is made out of properties considered as the body of concept, *context* is legally declared setting, *properties* provides the normative body of a concept, *constitutive rules* that helps new normal state-of-affairs derived from the multiple interpretations of the legal definition, and the *purposes* of a legal definition provides a wide ranges of scopes for legal validation of the same legal definition.

2.2. An Example of the Structure of a Legal Definition

1.2.1. Direction

The definition of 'water intended for human consumption' begins with the direction pointing that 'For the purposes of this directives'. It generally provides following inherent features –

- It expresses a wide range of applicability of different components of the definitions. Because this legislation contains total 34 purposes that expresses wide horizon of contextual typicality and applicability effect. At the same time, it also limits definition's expressivity and applicability. For example, none of these purposes indicates about the energy consumption for per drop of water.
- This direction itself express a constitutive rules in order to legally validate the construction of the legal definition. One of the examples of such constitutive rules in this definition is if any context mentioned in the definition violated any part, if not entirely, of the purposes mentioned in the legislation is legally not valid. More precisely, the context 'water includes all water used in any food-production intended for human consumption' in the definition legally rationalizes the purpose no 7 of the legislation, which says 'whereas it is necessary to include water used in the food industry unless it can be established that the use of such water does not affect the wholesome of the finished product'.

Table 1. Example of the strucutre of a legal definition

Article 2(1) of the Water Directive 98/83/EC says –

"For the purposes of this Directive: 'water intended for human consumption' shall mean: (a) all water either in its original state or after treatment, intended for drinking, cooking, food preparation or other domestic purposes, regardless of its origin and whether it is supplied from a distribution network, from a tanker, or in bottles or containers; (b) all water used in any food-production undertaking for the manufacture, processing, preservation or marketing of products or substances intended for human consumption unless the competent national authorities are satisfied that the quality of the water cannot affect the wholesomeness of the foodstuff in its finished form."

Meaning of color legends

For the purposes of this Directive = ***Direction*** through the constitutive rules for the definitions [for the purposes of legal reasoning] directed to the purpose of the legislation

Water = ***Concept***

intended for human consumption = ***The state of concept***

intended for = ***Property***

Either in its original state or after treatment = ***Context***

Unless the competent national authorities are satisfied that the quality of the water cannot affect the wholesomeness of the foodstuff in its finished form. = ***constitutive rules***

1.2.2. Concepts

The concept 'water' is as a legal entity based on a set of properties 'intended for human consumption'. This set of properties also expresses the state of the concept 'water'. Now, for example, if the state of the concept 'intended for human consumption' changes with the state of 'intended for bio-fuel production', then the applicability effect

of the set of the properties ‘intended for human consumption’ will also change over the legal entity ‘water’ and eventually it will also change the contextual typicality of the concept ‘water’. Hence it may lose its applicability under the purposes of this legislation.

1.2.3. State of the Concept

The constitutive rule implicitly embedded in the state of the concept may plays an important role in order to experiment the changes that might happen in the concept under the influence of changing the state of the concept. For example, if any change happens in the state of the concept ‘water used in food production is intended for human consumption’ by a new state of the concept derived from the reality such that ‘water used in the medicinal products’ will change the applicability effect of the concept ‘water’ as it is also mentioned in the purposes no 10 of the legislation, which says ‘whereas it is necessary to exclude from the scope of this Directive natural mineral waters and waters which are medicinal products, since special rules for those types of water have been established’.

1.2.4. Context

If any change happens in the ‘context’ of a concept, it may changes the applicability effect of the properties or state of the concepts. For example, if the context ‘all water either in its original state or after treatment’ is changed by new context ‘all water during the treatment’, it may changes the applicability effect of the properties or the state of concepts, e.g. ‘intended for’ and concept itself, e.g. ‘drinking’.

1.2.5. Property

The property ‘used in’ makes relationship between the concepts ‘water’ and ‘food production’. However, if changes take in place in either any of these concepts or properties itself, it may changes again the applicability of the properties between these concepts or vice versa. For example, if the property ‘used in’ remains same, but the concept ‘food production’ is replaced by ‘energy production’ then the applicability effect of the property ‘used in’ between the concepts ‘water’ and ‘energy production’ will changes. Then eventually the relationship established by the property ‘used in’ between the concepts ‘water’ and ‘energy production’ will be legally valid relationship under the constitutive rules coming the direction of this legal definition. This change any also bring a set of changes among other concepts, state of concepts, contexts, properties of the definition.

1.2.6. Constitutive Rule

The legal definition contains a number of implicit or explicit constitutive rules. In the above example, the direction of definition ‘for the purposes of this directive’ itself is an implicit constitutive rules. It implies that if the applicability effect of any part, if not all, of the definition violates any part, if not all, of the purposes mentioned in the beginning of the legislation, it may reasons legally invalid definition partly or entirely, see the example in the ‘Direction’ section. Another example of the constitutive rules

that exist explicitly in the definition is 'unless the competent national authorities are satisfied that the quality of the water cannot affect the wholesomeness of the foodstuff in its finished form'. The applicability effect of this constitutive rules is very wide and cross-sectorial related with multiple sets of external contexts and concepts such as the concept 'quality of water' that requires legal applicability of a number of parameters given in the appendix of the legislation. On the other hand, the examination of wholesomeness of the foodstuff may also requires to have legal institutional relation with the administrative authorities of food domains.

2.3. The Applicability Effect of Changes in One Component of the Legal Definition to Others

All components and their internal structures are interweaved in any legal definition. An interference, caused by any perspective or interpretation, in one of the components of the legal definition, as it is discussed in section 2.1 and 2.2, may changes the applicability effect of the other components of the legal definition as a 'Doppler effect[5]' in a radar system, as shown in figure 1[6].

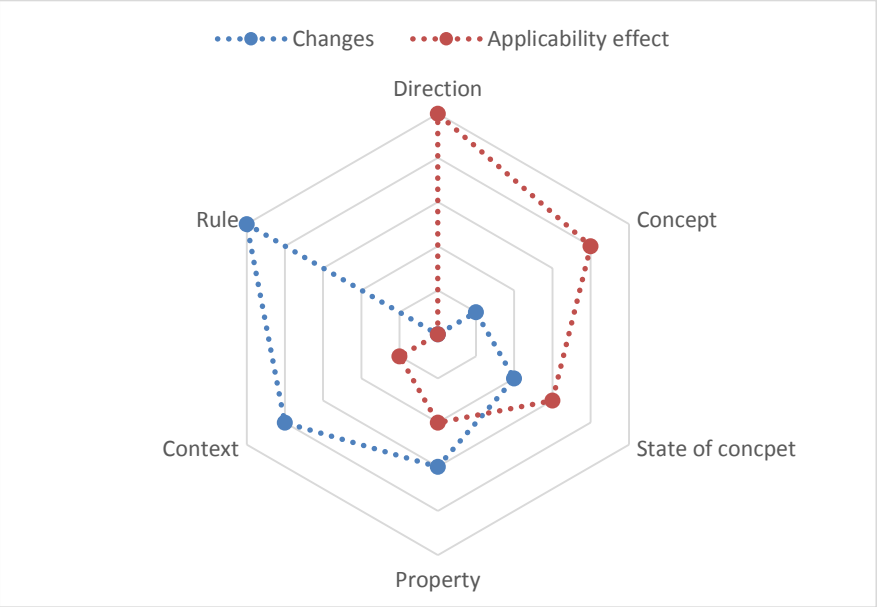


Fig. 2**.** Applicability effect of any changes in any component of the legal definition

Among the various components of a legal definition, the contextual typicality or rule typicality may effect applicability of other components at greater extend. For example,

[5] Doppler Effect happens when any frequency shift is caused by any level of motion that brings changes the numbers of wavelengths between the reflector and the radar. That may either enhance or degrade radar performance depending upon how that affects the detection process. See more at https://ccrma.stanford.edu/~jos/pasp/Doppler_Effect.html

[6] The following diagram is developed based on theoretical assumptions, not based on empirical study.

in the above mentioned water definition, see in Table 1, if a new contextual typicality, such as 'energy production' is replaced in the given contextual typicality 'food production', then the applicability effect of the entire legal definition of water may enforce in energy domain instead of food domain.

2.4. How Applicability Effect of Any Component of a Legal Definition Affects the Legal Reasoning

Applicability of the expressivity of any component of a legal definition, generally, limits capacities of the premises needed to construct any well-formulated argument for the legal reasoning process. For example, if no change happens in the properties of the concept 'water intended for human consumption', see the water definition in Table 1, the concept itself limits the capacities of the premises in order to construct a well-formulated argument for reasoning legally about water intended for energy consumption. However, if one single property 'human' is replaced by a new property 'energy' of the same concept 'water intended for human consumption', then the state of the concept will be changed and then the new state of concept will be considered as a new concept 'water intended for energy consumption'. Now this very new concept 'water intended for energy consumption' express new meaning with new understanding and applicability in the legal reasoning process of what it is meant by water.

In the case of legal reasoning of WEF nexus, complexities are much higher as the applicability effect of each of the legal definitions of water, energy and food is limited within their own legal deterministic radius, which is mainly directed by the direction or purposes of a legal definition. Therefore, the constitutive rules coming from each of these legal definitions are also applicable only within its own legal deterministic radius. As a result, the composite constitutive rules, the combined form of all constitutive rules coming from each of these definitions, are required in order to complete the legal reasoning circuit of WEF nexus.

2.5. Why It Is Necessary To Compute Applicability Effect of Any Change Happens in A Legal Definition

As it is discussed in section 2.4, the composite constitutive rules is required in order to understand the nexus. That requires to compute the applicability effects of any interaction happens by any interference caused by any property changes in any component of a legal definition or a set of legal definitions. In addition, it is also important to compute how a new set of contextual typicality or rule typicality, may appear from the outside of a legal definitions, may affects the applicability effect of any part, if not entire, of a legal definition or a set of legal definitions. That may open new possibilities to analyze legal definitions and constitutive rules in real space, time and legal setting.

3. Quantum Formalism of a Legal Definition in the Hilbert Space

The initial idea of how legal knowledge base of a legal definition or a set of legal definitions, that is capable to represent all structural components of a legal definition in a holistic way, can be designed is discussed below in brief.

3.1. State Context Property Systems (SCOPs) – A Concept-representation Model for Hilbert Space

The Brussels research group has been working on the axiomatic and operational approaches to quantum physics for last several decades [8]. The goal is to model specific macroscopic situations, besides modeling of quantum particle's situation in the micro-world, by using quantum-like formalism than by the classical formalism [15]. Because concept's combination coming from human cognition, on one hand, cannot be modelled adequately by using fuzzy set theory and, on the other hand, rules of classical logic technically is not capable to measure the membership weights of combined concepts and their interactions [16]. Considering of these weakness is particularly relevant in the case of modelling formally of a legal definition as a legal definition generally, if not always, is based on the combination of many concepts and their interactions between and among other components of a legal definition such as context, rule, purposes etc [17][18]. As a result, SCOPs theory is introduced as a generalization of quantum formalism, for representing human concepts and their dynamics in the Hilbert Space [8].

2.1.1. The Conceptual Framework of SCOPs System

The theory is basically based on an ontology of concept, consisting three structural components – state of concept, context and property, as an entity in a state that gets changes under the influence of a context. The applicability effect of the context over the state of a concept is measured by analyzing the role of complex amplitudes, which is basically coming from the different human's interpretations of a concept, context and their properties. This also helps to explain the interferences happens throughout the interactions within various components of a concept by analyzing the statistics of measurement outcomes from the role of complex amplitudes. The theory itself introduces a new way of concept formalism that opposes traditional concept theories, such as prototype theory, exemplar theory and theory theory, and classical probability weights.

2.1.2. The Mathematical Structure of SCOPs System

In order to build SCOP system mathematically, an arbitrary concept S is introduced with three sets – the set $\sum$ of states, denoting states by *p, q, ...,* the set M of contexts, denoting contexts by *e , f,....,* and the set L of properties, denoting the properties by *a , b, ……*. The ground state $\acute{P}$ of the concept S represents the state where the concept S is not under any influence of any context. However, when S is under the influence of a specific context *e,* change occurs in the state of the concept S. In this case, the ground state $\acute{P}$ of the concept S changes to the state *p*. The difference between the ground sate $\acute{P}$ and the changed state *p* of the concept S is demonstrated by the typicality values of the different exemplars of the concept and the applicability values coming from different properties being different in two states $\acute{P}$ and *p*.

In addition, two mathematical functions μ and ν has been introduced. The function

$$\mu: \sum * M * \sum \rightarrow [0, 1]$$

is defined such that $\mu(q,e,p)$ is the probability when the state p of the concept S under the influence of context e changes to the state q of the concept S. The another function

$$v: \sum * L \rightarrow [0, 1]$$

is defined such that $v(p , a)$ is the weight or the normalization of applicability of the property a in the state p of the concept S. These mathematical tools and structure of SCOPs formalism of a concept manage both contextual typicality and contextual applicability.

3.2. Why SCOP Formalism Is Not Capable to Represent a Legal Definition in a Complete and Sound Form

In the architectural design of a knowledge base, whereas OWL Full[7] considers concept and their properties, the SCOP formalism considers state, properties and context of a concept. Even though the formalism of a context of a concept in a concept representation of a knowledge base is well performed by SCOP system, some important components of a legal definition, such as rule and purposes that exist within the body of a legal definition, are not covered by SCOP formalism. More importantly a legal definition not only concerns about the contextual typicality and applicability, rather the rule's typicality and applicability within the legal definition in conjunction with the purposes of the legal definition arise the formalism complexity of a legal definition. That requires to develop an extended version of SCOP formalism in order to manage all components of a legal definition in a complete and sound form.

3.3. The Extend Version of SCOP Formalism for Representing a Legal Definition in the Hilbert Space

As it is discussed in section 2, a legal definition may have six-dimensional structure consisting with purpose or direction, concept, state of concept, property, context and rule. However, it is not fundamentally a necessary condition, on one hand, to have all of these components in the formation of a legal definition. On the other hand, the structure of a legal definition may not limit within this six-dimensions. In the case quantum formalism, as Hilbert Space is an infinite dimensional mathematical machine, any number of components and their multi-dimensional characteristic of a legal definition or any number of legal definitions can be easily represented in the Hilbert Space.

In the case of WEF nexus, the legal definitions of water, energy and food contain six-dimensional structure, shown in the Figure 3. The role of complexity amplitude appears from the values of many-world interpretations of any component of a legal definition. This leads to measure the applicability effect and its normalization of any changes happens in any component of a legal definition.

3.3.1. Necessary Conditions for Quantum Formalism of a Legal Definition

[7] See http://www.w3.org/TR/owl-ref/

The following conditions must be well-care off in the process of quantum formalism of a legal definitions or any number of legal definitions –

- A legal definition generally, if not in all cases, may combines many concepts, contexts, purposes and rules through their state and properties within its body all together in order to give birth of a meaning that is legally valid and executable or extend of any legally established meaning of any point of a legal discourse.
- Therefore, unlike SCOP system, any change happens by any interference in any component of a legal definition may influence the applicability effect on the entire structure of a legal definition.

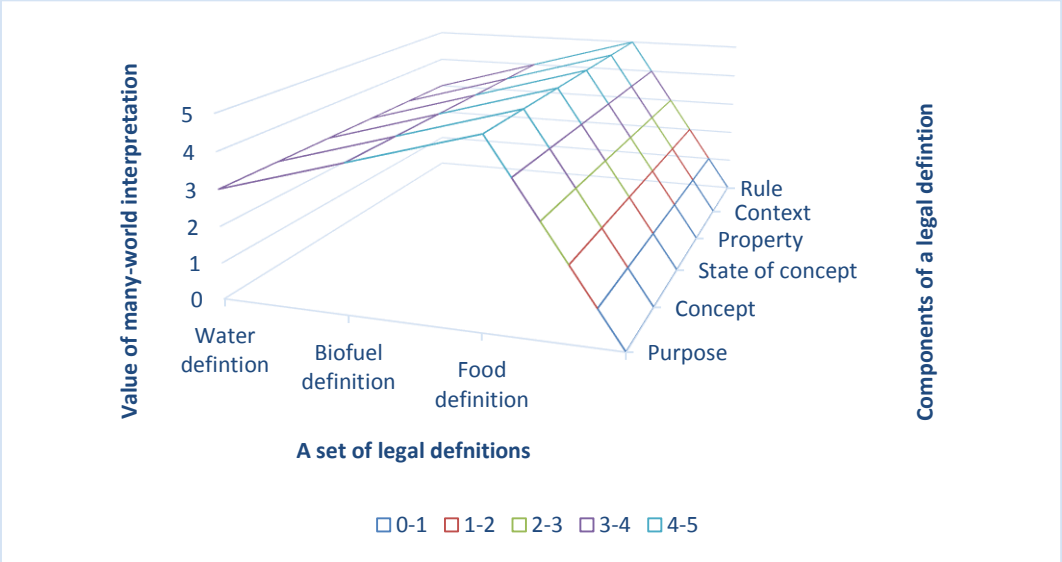


Fig. 3. A set of legal definitions in the Hilbert Space

- The value of many-world interpretations of any legal definition may represents the rate and frequency of interpretation of contextual typicality as well as of rule's typicality. It also may represent weight of the properties of concept, context or rules that exist within the body of legal definition. In addition, it may also represent rate and frequency of the purposes of a legal definition.
- The value of many-world interpretation leads the role of complexity amplitude, which helps measuring applicability effect of concept, context, rule and purpose over the definition and the normalization of their associated properties and state.

3.3.2. Mathematical Structure for Quantum Formalism of a Legal Definition under the Context of WEF Nexus

Let us denote the legal definition of water by W, bio-fuel definition by $\mathcal{B}$ and food definition by $\mathcal{F}$ and each of these legal definitions has six components – concept, context, rule, purpose, state and property. Let the set of states of water definition be $\sum_W$={

$W_1, W_2, W_3, W_4, W_5, W_6\}$, let the set of sates of biofuel definition be $\sum_{\mathcal{B}} = \{ \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6\}$, let the set of food definition be $\sum_{\mathcal{F}} = \{ \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6\}$, where the elements indexed by 1 represent the concept component , the elements indexed by 2 represent context component, the elements indexed by 3 represent rule component, the element indexed by 4 represent purpose component, the element indexed by 5 represent state component and lastly the elements indexed by 6 represent property component for each state.

The function for applicability effect of changes happen in a legal definition is defined as follows:

$$f_A: \left(\sum_A\right)^6 \mapsto \left(\sum_A\right)^6, \ A \in \{W, F, B\}$$
$$(A_1, A_2, A_3, A_4, A_5, A_6) \mapsto (A_1', A_2', A_3', A_4', A_5', A_6')$$

$f_A(\underline{a}) := f_A(a_1, a_2, a_3, a_4, a_5, a_6) \mapsto (f_{A1}(a_1), f_{A2}(a_2), f_{A3}(a_3), f_{A4}(a_4), f_{A5}(a_5), f_{A6}(a_6))$

such that,

$$f_{An}: \sum_A \mapsto \sum_A, \ A \in \{W, F, B\}$$
$$a_n \mapsto f_{An}(a_n) = a_n', \ \forall n \in \{1, \ldots, 6\}.$$

4. Final Remarks

The paper presents briefly the critical issues present in the legal reasoning process of WEF nexus. These mainly two-folds – (a) the deterministic nature of legal definitions coming from water, energy and food domains has non-deterministic effect on each other's, and (b) the constitutive rules coming from each definitions are legally confined within a deterministic but legal structure. Therefore the implications of the definitions of each domain does not cover WEF nexus to be legally reasoned. That makes a basis of why the structural component of a legal definition must be systematized in order to measure the applicability effect of each component of a legal definition over others in order to generate not only well-formulated constitutive rules for each legal definition, rather to normalize the applicability effect of one legal definition over others by generating composite constitutive rules coming from a set of legal definitions and their associated contexts and constitutive rules.

Therefore, the paper then presents six structural components of a legal definitions with an example of Article 2(1) of the Water Directive 98/83/EC. These are – purpose, concept, context, rule, state and property. Then it shows two fundamental limitations, representational and reasoning, of OWL Full in order to design a knowledge base of a legal definition in a complete and sound form. Then it traduced SCOP systems known as a generalization of quantum formalism of human concepts, which can represent also context along with concept and their properties in a quantum-like knowledge base designed in Hilbert Space. But both OWL Full and SCOP quantum formalism has limited capacity to represent all components of a legal definition in a complete and sound form. Because OWL Full, on one hand, covers concepts, properties and their relations, and

on the other hand, SCOP quantum formalism covers state, context and property. What would be left if a knowledge base of a legal definition is designed by either OWL Full or SCOP quantum formalism is two other important components – rules and purposes, of a legal definition. Therefore, the paper at the end briefly proposed an extended form of SCOP quantum formalism that is capable to design a legal knowledge base of a legal definition or a set of legal definitions in a complete and sound form in the Hilbert Space. However, the future papers of this series will bring more quantum mathematical structures in order to manifest the composite constitutive rules coming from a set of constitutive rules inherent in a legal definition.

Acknowledgement. Thanks to all professors and fellows of Erasmus Mundus LAST-JD program for their inspirational support and discussion on this paper and I especial thank Fatmanur Yildirim for her intensive help for developing functions.

References

[1] UNWater. Water security and the global water agenda: a UN-Water analytical brief. pp. 26. Ontario: United Nations University Press (2013).
[2] Bizikova L, Roy D, Swanson D, Venema HD, McCandless M. 2013. The water energy food security nexus: towards a practical planning and decision support framework for landscape investment and risk management, pp. 14. Manitoba: International Institute for Sustainable Development (2013)
[3] Campbell A. Food, energy, water: conflicting in securities (and the rare win-wins offered by soil stewardship. Journal of soil and water conservation 63(5): pp 149-151, (2008)
[4] [NIC, USA] National Intelligence Council, US. Global trends 2030: alternative worlds. Washin ton: Office of the Director of National Intelligence, pp 5, (2012)
[5] Bazillian M, Rogner H, Howells M, Hermann S, Arent D, Geilen D, Steduto P, Mueller A, Komor P, Tol RSJ. Yumkella KY. Considering the energy, water and food nexus: towards an integrated modelling approach. Energy Policy 39(12):pp 7896-7906, (2011)
[6] Baltag, A., and Smets, S. LQP – the dynamic logic of quantum information. *Mathematical Structures in Computer Science* 16(3): pp 491-525, (2006)
[7] Wichert, A. Principles of quantum artificial intelligence. Singapore: World Scientific Publishing Co.Pte.Ltd, (2014)
[8] Aerts, D., and Gabora, L. A theory of concepts and their combinations I: the structure of the sets of contexts and properties. *Quantum Physics* 34(2): pp 151-175, (2005).
[9] Andrews-Speed P, Bleischwitz R, Boersma T, Johnson C, Kemp G, VanDeveer SD. The global resource nexus: the struggles for land, energy, food, water and minerals. Washington: Transatlantic Academy. pp 55, (2012)
[10] UN-ESCAP. The status of the water, food, energy nexus in Asia and the pacific. Bangkok: UN publication. pp 38, (2013)
[11] Fry A. Facts and trends water. Geneva: World Business Council for Sustainable Development. pp 2, (2006)

[12] Levy, E.H. An introduction to legal reasoning. Chicago: University of Chicago Press, (1948)
[13] Hannu, T.K., Johanna, S. Minna, H. Evidence and legal reasoning: on the interwinement of the probable and the reasonable. Law and Philosophy, 10(1), pp 73-107, (1991)
[14] Jaap, H. A theory of legal reasoning and a logic to match. Artificial Intelligence and Law, 4(3-4), pp 199-273, (1996)
[15]Jauch, M. Foundations of quantum mechanics, Addison Wesley, New York, NY, (1968).
[16] Rips, L.J. The current status of research on concept combination. Mind and Language, Vol. 10, pp. 72-104, (1995)
[17] Randall, C. and Foulis, D. "Properties and operational propositions in quantum mechanics", Foundations of Physics, Vol. 13, pp. 835, (1983)
[18] Nosofsky, R.M., Kruschke, J.K. and McKinley, S.C. "Combining exemplar-based category representations and connectionist learning rules", Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 18, pp. 211-33, (1992)

The Applicability of Nationality Extraction in Crime Domain to Construct Legal Roles Extraction in Law Domain

Masnizah Mohd [1], Mohammad Darwich [2], Kiyoaki Shirai [1]

[1] Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan
{masnizah, kshirai}@jaist.ac.jp
[2] Universiti Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia
modarwish@hotmail.com

Abstract. Nationality is one of the significant information used by crime analyst to analyze whether a person extracted from the crime news could be referred to as the suspect, victim, witnesses or the offender. However crime analysts must manually search for it. To solve this issue, several information extraction models have been implemented, all of which are capable of being enhanced. This has created the motivation to propose an enhanced information extraction model that uses named entity recognition to extract the nationality from crime news and co-reference resolution to associate the nationality to either the suspect or the victim. We constructed a pipeline consisting of a dictionary-based nationality extractor (Nationality Indicator List, Nationality List, Country List, Victim Keyword List, Suspect Keyword List) and a distance-based reference linker. We also discussed the applicability to construct legal roles extraction in law domain; concept (Citizen, Resident, Domicile and Immigrant) and instance attributes (Citizen Keyword List, Resident Keyword List, Domicile Keyword List and Immigrant Keyword List). Two experiments were conducted to evaluate the model. The first experiment (Nationality Extractor Component) achieved 90%, 94%, and 91% and the second experiment (Reference Identification Component) achieved 65%, 68%, and 66% for precision, recall, and f-measure respectively. The model has achieved promising results after evaluation.

***Keywords*:** Nationality, Crime, Citizenship, Law

1 Introduction

Crime analysts and investigators access criminal justice data and intelligence on crime cases efficiently (in terms of speed) and effectively (in terms of accuracy) to perform investigations and prevent crime. There is valuable information in online crime news, which usually contain text that is unstructured. Valuable information are entities within the text, which may be person names, nationalities, crime locations, crime dates, crime types, criminal properties, weapons used, narcotic drugs, car brand, among others [1,2]. Information Extraction (IE) systems are used to extract significant information.

2 Information Extraction Models in Crime Domain

Several IE models (or systems) have been implemented in order to keep analysts and investigators updated with the information they need accurately and efficiently [3]. These systems [1,2,4,5,6,7,8] have successfully guided analysts with crime cases [2].

In the domain of crime and crime analysis, information on crime is needed as quickly as possible and as accurately as possible. It is difficult to manually access data that is needed for investigations [2].

[1] created a neural network based entity extractor, which implements named entity extraction techniques to extract address, person, drugs and personal property from crime documents and police reports. The model had a precision value and recall value for person name of 74.1% and 73.4% respectively. The precision value and recall value for narcotic drugs was 85.4% and 77.9% respectively. The model performed with greater accuracy for narcotic drugs.

[2] created an IE model customized for the crime domain that uses Natural Language Processing (NLP) to identify crime related information from police reports, witness narratives, and news. The model extracts people, vehicles, weapons, time, locations, and clothes, and was tested on two different types of documents, which were police narrative reports and witness narrative reports, that were all obtained from online forums, blogs, and news. The model uses both the lexical lookup and rule-based approaches. The model had achieved high precision and recall values of 96% and 83%, respectively, when tested on police narrative reports. However, the model achieved lower precision and recall values of 93% and 77% when tested on witness narrative reports.

[5] has created the Crime Type Recognition System (CTRS) that recognizes different crime types and uses two combined techniques. The first one is direct recognition using gazetteers of crime verbs and crime names. The second one is completely rule based, and relies on several rules and a crime indicator list to identify crime types. The work of [5] was developed based on the previous research of [6], who had used a rule based approach to create a named entity recognition system, and also based on the research of [9], who had developed a multilingual named entity recognition framework using the rule based approach. The model was tested against human based entity extraction, and the precision, recall and f-measure were recorded to be 60%, 97%, and 74% respectively. By increasing the value of the recall, the precision had decreased accordingly.

[8] has created an IE model that extracts the nationality from online crime news, and uses co-reference identification to associate the nationality to either a Suspect (S) or Victim (V) or none (N, if nothing is matched). Regarding the evaluation of the direct and indirect extraction approaches, the precision, recall, and f-measure values were 55%, 96% and 70% respectively. Regarding the evaluation of the victim or suspect reference identification, the precision, recall, and f-measure values were 62%, 53%, and 57% respectively. Although the model was effective, the approach used is not dynamic because it references nationality to suspect or victim based on the nearest keyword from the position of the nationality. Hence, it is capable of being enhanced.

Therefore in our work, we constructed a probability score algorithm to decide whether the nationality is more likely to belong to either the victim or the suspect based on their total distances.

3 Nationality Extraction Model

During the first stage, data related to the crime domain was gathered to create the corpus used for this work. The data source used for the corpus was collected and gathered from Bernama, the Malaysian national news agency. The test corpus includes approximately 248 crime news, 48 of the documents are from the same dataset used by [8]. An example of a crime news that was used for this research is shown in Figure 1.

04/02/2011

FOUR FOREIGNERS CAUGHT CUTTING DOWN KARAS TREE

JOHOR BAHARU, 4 Feb (Bernama) -- Four foreigners have been caught cutting down a Karas tree in Pulau Pisang, Pontian district police chief Supt Zahaliman Jamin said today. They were arrested on Wednesday with the help of members of Rela, the people's volunteer corps, he said. The four men, aged between 29 and 47, did not have travel papers with them. Karas, which can be used as incense and for medicinal purposes and has a very high commercial value, has been called the black gold of the forest. Zahaliman said police were trying to determine whether the four suspects were part of a syndicate or operated on their own.

Fig. 1. Sample crime news from corpus

The proposed model uses a lexical lookup approach and a rule based approach. The proposed model contains the gazetteers lists (Section 3.1), preprocessing component, the nationality extractor component (Section 3.2), and the victim or suspect reference component (Section 3.3).

The pre-processing component includes five main parts, which are title removal, HTML tag removal, punctuation removal, lowercase conversion, and tokenization.

3.1 Gazetteers and Internal Lists

The gazetteers and internal lists were gathered. The Nationality List (NL), Nationality Indicator List (NIL), and Country List (CL) were collected from Wikipedia, and also from the GATE[1] gazetteer lists. In addition, some manual additions were made to the NIL by searching the documents for any words that indicate a nationality, such as "from" and "national". These lists are used to check if there is a nationality match in

[1] https://gate.ac.uk/

the nationality extractor component. The Victim Keyword List (VKL) and Suspect Keyword List (SKL) consist of both verbs and nouns. Both lists were constructed manually, by searching the test data for any words that were either victim related, for the VKL, or suspect related, for the SKL. Table 1 show samples of the gazetteer lists respectively.

Table 1. Gazetteer lists

No.	Gazetteer lists	Values
1.	Nationality Indicator List (NIL)	from, national, nationals, nationality, nationalities, originally, origin ……
2.	Nationality List (NL)	Koreans, Kosovo Albanians, Japanese, Kuwaitis, Lao, Latvians, Lebanese, Liberians, Libyans, Liechtensteiners, Lithuanians, Luxembourgers, Macedonia, Malaysian …..
3.	Country List (CL)	French Guiana, French Polynesia, French Southern, and Antarctic Lands, Gabon, Gambia, Gambie, Gaza Strip, German Democratic Republic, Germany, Ghana, Gibraltar, Great Britain, Greece, Greenland, Grenada, Grenade, Greenland, Greece, Guadeloupe, Guam, Guatemala, Guernsey, …….
4.	Victim Keyword List (VKL)	victim, victims, dead, died, hospitalized, wounded, stabbed, suffer, suffered …
5.	Suspect Keyword List (SKL)	suspect, suspects, caught, nabbed, committed, detained, detainees, arrested …

3.2 Nationality Extractor Component

The output of this component is shown to the user, and is also used as input for the next component, which is the reference identification component. This component works based on two algorithms, the direct match algorithm and the indirect match algorithm. The model first uses the direct algorithm to check for matches using the Nationality List (NL). An example of a direct match is "Indonesian". The direct match algorithm is shown in Figure 2.

```
Require: file ≠ 0
token <= file
nl <= NationalityList
while   file ≠ 0
          for nlCounter <= 0 to length[nl]-1     do
                    if     token == nl[nlCounter]    then
NationalityExtracted <= token
                    end if
          end for
end while
```

Fig. 2. Direct match algorithm.

If there are no matches found using the direct match algorithm, the component moves on to attempt to use the indirect match algorithm, along with the Nationality Indicator List (NIL) and the Country List (CL) to find any matches. An example of an indirect match is "from Indonesia". The word "from" is matched using the NIL, and the nationality "Indonesia" is matched using the CL. Figure 3 shows the indirect match algorithm.

```
Require: file ≠ 0
token <= file
previousToken <= token[-1]
nextToken <= token[+1]
cl<=CountryList
nil<=NationalityIndicatorList
while   file ≠ 0
          fornilCounter <= 0 to length[NIL]-1     do
                    if     token == nil[nilCounter]    then
                       NationalityExtracted <= token
                  end if
end for
          for nilCounter <= 0 to length[nil]-1     do
                    if     token == nl[nilCounter]    then
for clCounter <=0  to length[cl]-1    do
                          if    previousToken == cl[clCounter]  or
                          nextToken ==  cl[clCounter]     then
                                    NationalityExtracted <= previousToken +token or
                                    NationalityExtracted <= token +nextToken
                  end if
              end for
          end if
      end for
end while
```

Fig. 3. Indirect match algorithm.

After this component has extracted all of the nationalities in the text, either by using the direct match algorithm or indirect match algorithm, it stores all of the nationalities in an array. This array of nationalities is output to the user, and also used as input to the reference identification component.

3.3 Victim Suspect Reference Identification

Finally, the last component is the victim or suspect reference identification component. During model execution, for each nationality, this component attempts to associate the nationality extracted to being a suspect or victim. It does this using the Victim Keyword List (VKL) and Suspect Keyword List (SKL). First, it uses the VKL to find all of the words in the text that are victim related. Next, it calculates the distance of each victim related word in relation to the position of the nationality extracted. For example, if a victim related word is four words away from the nationality within the text, it has a distance of four (distance = 4.0).

After that, the distances of all the victim related words are added to give the total distance, which is divided by their count in order to get the average distance. The average distance is the victim probability score. Next, the suspect probability score is calculated using the same technique used to calculate the victim probability score. These two probability scores are stored by this component to be used for comparison.

Both the victim probability score and the suspect probability score are compared. If the victim probability score is less than the suspect probability score, then the nationality extracted is referenced as that which belongs to the victim feature. This is because the victim probability score is less, which means that the average distance between the nationality position and the victim related keywords positions is less than the average distance between the nationality position and the suspect related keywords positions. Hence, the probability of the nationality being related to the victim is higher due to the shorter distance of the victim related keywords from the nationality position. Figure 4 shows the reference identification algorithm used by the model.

```
Require: file ≠ 0
while   file ≠ 0
find positions of all victim related keywords
find positions of all suspect related keywords
        for n <= 0 to length[nationalityArray]-1     do
             for victimKeyword <= 0 to length[victimKeywordsArray]-1     do
              calculate distance between keyword and nationality
              totalDistance <= sum of distance of all keywords from nationality
            end for
              avgDistance <= totalDistance / countOfVictimKeywords
              victimProbabilityScore <=   avgDistance
             for suspectKeyword <= 0 to length[suspectKeywordsArray]-1     do
              calculate distance between keyword and nationality
              totalDistance <= sum of distance of all keywords from nationality
           end for
              avgDistance <= totalDistance / countOfSuspectKeywords
              victimProbabilityScore <=   avgDistance
            IF (victimProbabilityScore < suspectProbabilityScore)
                Nationality referenced to victim
            ELSE
                Nationality referenced to suspect
            end for
end while
```

Fig. 4. Reference identification algorithm.

Figure 5 shows the victim suspect reference identification component after processing a snippet of text from the corpus. In this case, the average distance between the victim related keywords and the nationality is 8. This is the victim probability score. The average distance between the suspect related keywords and the nationality is 7.33. This is the suspect probability score. The suspect probability score is less than the victim probability score, which means that the suspect related keywords are nearer to the position of the nationality, and therefore the nationality "Iranian" is referenced to "Suspect".

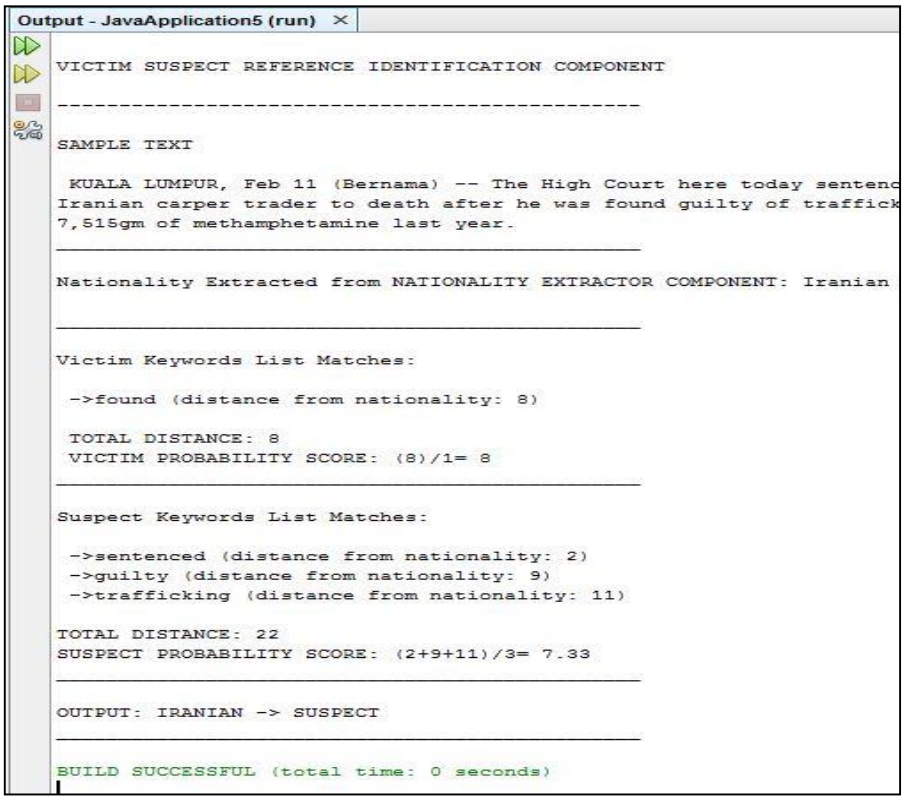


Fig. 5. Output of victim suspect reference identification component

4 Evaluation

This section presents the evaluation of the effectiveness and efficiency of the proposed IE model during the process of nationality extraction and reference identification to suspect or victim features. The model was used to process a total 248 crime news from the test data used in this work. 48 random documents were selected to be included in the experiments that were performed on the model.

Each document was input into the system, and went through the three system components (the preprocessing component, the nationality extractor component and the reference identification component) mentioned previously, in order to have the nationalities extracted, and to have each nationality referenced. In addition, all of the same 48 documents were processed manually, by the researcher. Both manual processing and system processing were compared in order to measure how well the model did in terms of the efficiency and accuracy. Two experiments were conducted.

a. Experiment 1: Evaluation of Nationality Extractor Component
   The first experiment was conducted to evaluate the nationality extraction component of the model. For every document, the researcher had extracted all of the nationalities in the documents, after reading and comprehending the text.
b. Experiment 2: Evaluation of Reference Identification Component
   The second experiment was conducted to evaluate the reference identification component of the model. In this experiment, the 104 nationalities that were manually extracted from the test data in the first experiment were used. For every document, the researcher had manually referenced every nationality, and recorded whether the nationality was associated with either the victim feature or the suspect feature, after reading and comprehending the text. After the reference identification process was done manually, it was performed by the proposed model. The model uses co-reference identification to associate the nationality to either a suspect or victim or none (if nothing is matched).

The effectiveness, or accuracy, of the IE model was measured in terms of precision, recall and f-measure. These evaluation metrics are the standard metrics in use for measuring how well information extraction systems perform [10]. To do an evaluation on a model, it should contain a document collection, specific information to be extracted (such as person name) and a binary value to state whether the extracted piece of information is relevant or not relevant [10,11].

The results from the first and second experiment were promising and performed well according to the evaluation metrics. The proposed model achieved relatively higher results. Overall, these results are relatively high and this component did its job successfully with high evaluation metrics as shown in Table 2.

Table 2. Comparison of Experiment 1 and Experiment 2

	EXP. 1			EXP. 2		
Setup	P	R	F	P	R	F
Baseline [8]	55	96	70	62	53	57
Experimental	90	94	91	65	68	66

P=Precision (%); R=Recall (%); F=F-measure (%)

Based on the analysis of the results from both experiments, several observations were obtained. The first one is that the proposed model have better results by including more keywords in both; the Victim Keyword List (VKL), and Suspect Keyword List (SKL), so that the model may input more keywords into the algorithm in order to reference the nationality to the correct features, and achieve more accuracy.

The second observation is that there are nationalities that are stated throughout the text that are not logically related or associated with the Suspect (S) or Victim (V), but are rather associated to other nouns (N), for instance, “Chinese New Year”, or “Malaysian Anti-Corruption Commission”. The nationalities *Chinese* and *Malaysian* are not associated with the Suspect or Victim features, but associated with other nouns (N).

5 Constructing Legal Roles Extraction in Law Domain

Nationality extraction in crime domain is also applicable to construct legal roles in law domain as shown in Figure 6.

CITIZENSHIP	
Instance attributes	Concept
Date Country List (CL) Nationality List (NL) Citizen Keyword List (CKL)	Citizen
Date Country List (CL) Nationality List (NL) Resident Keyword List (RKL)	Resident
Date Country List (CL) Nationality List (NL) Domicile Keyword List (DKL)	Domicile
Date Country List (CL) Nationality List (NL) Immigrant Keyword List (IKL)	Immigrant

NATIONALITY	
Concept	Instance attributes
Suspect	Date *Location (crime; non-crime) *Weapon Person Country List (CL) Nationality List (NL) Suspect Keyword List (SKL)
Victim	Date *Location (crime; non-crime) *Weapon Person Country List (CL) Nationality List (NL) Victim Keyword List (VKL)
*Witnesses	Date *Location (crime; non-crime) *Weapon Person Country List (CL) Nationality List (NL) *Witnesses Keyword List (WKL)
*Offender	Date *Location (crime; non-crime) *Weapon Person Country List (CL) Nationality List (NL) *Offender Keyword List (OKL)

*future work

Fig. 6. Citizenship in Law domain and Nationality in Crime domain

We can differentiate the Date in Citizenship; Date in Citizen concept is lifetime whereas Date in Resident concept is restricted based on e.g. visa period. Table 3 show samples

of the gazetteer lists under Citizenship concept. Gazetteer for Citizen Keyword List (CKL) should consist term that support instances e.g. birth, marriage, family relation and naturalization process.

Table 3. Gazetteer lists for Citizenship

No.	Gazetteer lists	Values
1.	Citizen Keyword List (CKL)	born, stayed, lived, resided, voted, served, …….
2.	Resident Keyword List (RKL)	worked, studied, business, travelled, owned, resided, voted, certified, …..
3.	Domicile Keyword List (DKL)	taxed, resided, purchased, titled, registered, obtained, …..
4.	Immigrant Keyword List (IKL)	worked, studied, business, travelled, ….

The same approach (direct and indirect algorithm) can be applied for reference identification. Indirect algorithm calculates the distance of each Citizen/Resident/Domicile/Immigrant related word in relation to the position of the nationality extracted. Term score using *tf.idf* can be used to identify values in the gazetteer lists.

6 Conclusion

In this paper, we used named entity recognition to extract the nationality from crime news and co-reference resolution to associate the nationality to either the suspect or the victim. We constructed a pipeline consisting of a dictionary-based nationality extractor (Nationality Indicator List, Nationality List, Country List, Victim Keyword List, Suspect Keyword List) and a distance-based reference linker. Furthermore, we also have discussed the applicability of nationality extraction in crime domain to construct legal roles extraction in law domain by focusing on Citizenship.

This is one of early work that extract nationality and perform the prediction/referencing task to suspect or victim. It look simple but the idea of constructing Nationality and concept (Suspect, Victim) with instance attributes (NIL, NL, CL, VKL, SKL) is relevant and shall contribute to crime ontology design. This approach is also feasible and applicable in law domain. We believe this preliminary work will open and invite more extensive research on information extraction focusing on crime domain.

Finally we have created a crime corpus and crime related dictionaries. We aim to perform several improvements, e.g., embedding Witnesses and Offender, to extract Location into Crime and Non-crime and to extract Weapon. Identifying patterns of verbs with weapon to differentiate between Suspect, Victim, Witnesses and Offender. For example:

a. hold/found/arrested (verb) + knife (Weapon List): Suspect, Offender
b. death/lying/sticking/stabbed (verb) + knife (Weapon List): Victim
c. saw/recall/witness/describe (verb) + knife (Weapon List): Witnesses

We also aim for a combination with other approaches e.g., advanced machine learning techniques or text mining methods in order to support rule formalization.

References

[1] Chau, M., Xu, J.J and Chen, H. Extracting Meaningful Entities from Police Narrative Reports, Digital Government Society of North America. Los Angeles, California, 12(2): 1-5 (2002)

[2] Hao, H.C., Iriberri, A. and Leroy, G. Crime Information Extraction from Police and Witness Narrative Reports, Conference on Technologies for Homeland Security, IEEE, pp.193-198 (2008)

[3] Jurafsky, D. and Martin, J.H. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics, 2nd edition, Prentice-Hall (2009)

[4] Bache, R., Crestani, F., Canter, D. and Youngs, D. A Language Modelling Approach to Linking Criminal Styles with Offender Characteristics, Natural Language and Information Systems 5039, Springer, pp. 257-268 (2008)

[5] Alruily, M., Aladdin, A. and Abdulsamad, M. Using Self Organizing Map to cluster Arabic Crime Documents, Proceedings of the International Multiconference on Computer Science and Information Technology, 357-363 (2010)

[6] Shaalan, K. and Raza, H. NERA: Named Entity Recognition for Arabic. J. Am. Soc. Inf. Sci., 60: 1652–1663 (2009)

[7] Riloff, E. Information Extraction as a Stepping Stone Toward Story Understanding. Understanding Language Understanding, MIT Press, pp. 435-460 (1999)

[8] Alkaff, A and Mohd, M. Extraction of Nationality from Crime News. Journal of Theoretical and Applied Information Technology, 54(2):304-312, (2013)

[9] Poibeau, T. The Multilingual Named Entity Recognition Framework. Proceedings of the European Association for Computational Linguistics Conference (EACL 2003). pp. 155-158 (2003)

[10] Manning, D., Raghavan, P. and Schütze, H. Introduction to Information Retrieval, Vol. 1. Cambridge: Cambridge University Press (2008)

[11] Cunningham, H. GATE, a General Architecture for Text Engineering. Computers and the Humanities 36(2): 223-254 (2002)

Recognizing logical parts in Vietnamese Legal Texts using Discriminative Sequential Learning

Nguyen Truong Son[1], Nguyen Thi Phuong Duyen[1], Ho Bao Quoc[1], Nguyen Le Minh[2], Akira Shimazu [2]

[1] Faculty of Information Technology University of Science - VNU, HCMC, Viet Nam
[2] Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, JAPAN

ntson@fit.hcmus.edu.vn,phuongduyen022@gmail.com,
hbquoc@fit.hcmus.edu.vn, nguyenml@jaist.ac.jp,
shimazu@jaist.ac.jp

Abstract. Analyzing the structure of legal sentences in legal document is an important phase to build a knowledge management system in Legal Engineering. This paper proposes a new approach to recognize logical parts in Vietnamese legal documents based on discriminative sequential learning methods. Beside linguist features such as word shape features, part of speech features, we use semantic features of logical parts such as trigger features and ontology features to improve the result of the annotation system. Experiments conducted using two popular discriminative sequential learning algorithms including Conditional Random Fields (CRFs) and Margin Infused Relaxed Algorithm (MIRA) on a Vietnamese Business Law data set. The our best model obtained 78.1% at precision and 68.7% at recall measure. Compare to state-of-the-art systems, it improves the result for recognizing some logical parts.

Keywords: legal text mining; semantic annotation; machine learning; conditional random field; named entities recognition, discriminative sequential learning; margin infused relaxed algorithm

1 Introduction

In every country, the law system is always one of the most important parts that ensure the safety and the development of our society. Law articles in legal documents must be consistent with other articles. If this requirement is not satisfied, our society will have suffered from the political and social unrest. In addition to, the number of law documents in a legal system is very big so that law experts cannot check the consistency in these documents manually or make mistakes easily. Therefore, we need to build a legal based knowledge management system which be able to automatically exam and verify whether a law contains contradictions, whether the law is consistent with related laws, and whether the law has been modified, added, and deleted consist-

ently. That is the mission of Legal Engineering - a new research field proposed in the 21st Century COE Program, Verifiable and Evolvable e-Society [1].

Legal Engineering is a new research field in the Vietnam research community. To build a fully worked-out legal based knowledge management system, we must solve many problems such as analyzing logical parts of law sentences in legal texts, recognizing legal terms, identifying the relationship between logical parts, recognizing the reference and so on. Among them, recognizing logical parts in legal texts is one of the fundamental tasks need to be solved first.

The legal document is a special kind of documents because it has special characteristics differed from those of regular documents. A challenge for understanding legal texts by using computer algorithms is that law sentences are often long and complex. However, they have well structure and minor errors in spelling and grammar. In addition, legal documents use typical structures and syntaxes of legal areas so we can use these characteristics to help our computer systems can understand legal documents better.

Each country has its own legal system and types of legal documents used differing from other countries. In Vietnam, legal documents are enacted by the government and the competent authorities. Moreover, formats, writing styles, grammar styles, and vocabularies used in legal documents must follow specified rules issued by the government. In legal documents, each law sentence includes a certain number of logic parts. *Assumption*, *provision*, and *sanction* parts are three main types of logical parts [17]. An assumption part indicates subjects (including individuals or organizations) of law sentences and conditions, situations in which those subjects encountered. A provision part describes rules, rights and responsibilities that subjects have or should do if they encountered conditions and situations described in the assumption part. A logical part in sanction type describes penalties or other means of enforcement if subjects do not follow the law provision. Besides, we also have 2 more special types of logical parts called *terms* and *definitions* that describe legal terms and their definitions in legal documents.

Three classical approaches used to solve this problem are dictionary-based, rule-based and machine learning based approaches. In rule-based and dictionary-based approaches, the result still depends much on the quality of the dictionary and rule sets. Moreover, the cost to find the rule set and dictionary that can cover all logical parts in legal documents is expensive. Machine learning approaches have many advantages especially if we have an annotated corpus. Among many learning algorithms, Conditional Random Fields (CRFs), a discriminative method, offers more advantages than others like Hidden Markov model (HMM) and Maximum Entropy. Therefore, this article utilize CRFs model to recognize logic parts automatically based on the model learned from a given corpus with many different feature sets. Beside common linguist features such as word shape features and part of speech features, we also use some other features that are typical of each type of logical parts in legal documents to improve the performance of our system. Moreover, we also do experiments using another discriminative method, Margin Infused Relaxed Algorithm, to choose the best model for recoginizing logical parts in Vietnamese legal documents.

Our experiments conducted on the Vietnamese Business Law corpus show that our method can achieve an accuracy measure of 78.1% and a coverage measure of 68.7% (F-measure of 73.1%). Besides word surface feature, adding part-of-speech features and special features such as trigger features and ontology features can improve the results of our system. Our system achieves the higher performance than the previous state-of-the-art rule-based system, especially if we have a large training corpus.

The remainder of this paper is structured as follows. Section 2 is related works. The system architecture and problem solving methods will be described in Section 3. Section 4 shows experimental results and comments. A brief conclusion and future work are showed in Section 5.

2 Related works

Recognizing logical parts in legal documents is a kind of semantic annotation problems that try to annotate the assumption part, the provision part, and the sanction part in text sequences of legal documents. In the same problem and the same data set with this paper, authors in [2] and [3] have used a rule-based approach by using GATE framework with a manual rule set written in JAPE language to recognizing logical parts in documents of Vietnamese Business Law data set. The system achieved an F-measure of 64.37% on assumption type, 64.15% on the provision type and 75% on the sanction type.

In another point of view, logical parts in law sentences include antecedent parts (describe conditions and contexts), consequence parts (describe law provisions and sanctions), and topic parts (describe subjects related to the law provision) [5]. Authors in [4], [5], [6], and [7] try to identify antecedent parts, consequence parts and topic parts in Japanese legal documents. In these papers, authors used CRFs method with language features such as word and neighbors, and their POSs features to build an annotation model to recognize these logic parts. The experience is conducted on the National Pension Law corpus get 74.37% at the F1 score.

Recognizing logical parts in legal texts can be treat like segmenting and labeling sequence data so it can be solved by using approaches which are popular these areas. In natural language processing and text mining, many problems can be casted as segmenting and labeling sequence data such as word segmentation, part of speech tagging, chunking, named entities recognizing (NER). NER is a semantic annotation problem which is popular in both general documents and specific domain documents. In general documents, the NER engine try to identify general entities such as person name, place or location, phone number… In the biomedical and health care field, people try to recognize special entities such as proteins, genres, DNAs, drugs, diseases in biomedical texts or electronic health care reports. In legal domain, people try to identify legal terms in legal case reports or legal texts.

Using rules, dictionaries and machine learning, or combining these methods together are popular approaches to segment and label sequence data. Whereas rule based approaches will recognize entities or concepts by applying a pre-built rule set on the input texts, dictionary approaches will check whether a piece of text belongs to

the dictionary used. An entity or concept will be recognized if a piece of text matches with a certain rule in the rule set or a record in the dictionary. However, the limitation of rule based and a dictionary-based approach is that we must have a rule set and a given dictionary in advance.

The work in [8] is one of the first researches used the rule-based approach with hand-written rule to identify company names in text. The result is over 95% precision. The authors in [9] proposed a rule-based approach to recognize entities in documents written by Urdu language. The result obtained a precision measure of 91.5%. Rule-based approaches will get the high precision for simple problems however it only covers entities and concepts matching with that rules. Therefore, it is costly to build a rule set which can cover all situations of the real life. In case of solving problems which we do not have an annotated corpus or other language resources, rule based approaches are usually used.

On the other hand, machine-learning approaches will learn the knowledge automatically from an annotated corpus to build a model by using learning algorithm with features extracted from this corpus. Then, that model will be used to recognize entities or concepts in new documents. Besides, rule sets and dictionaries can be used as additional features in machine learning approaches but they are not required. Among many machine-learning algorithms, CRFs approaches are the most popular learning algorithms which used to segment and label sequence data. CRFs got the lower error rate than other machine learning models such as Hidden Markov Model (HMM) or Maximum Entropy Markov Model (MEMM) approach [14]. Furthermore, CRFs approaches are used successfully in many researches in biomedical and chemical fields [10] [11], [12], [13], [14], [18] [19]. In the task of recognizing chemical compounds, [10] presented a tool called ChemSpot which based on a hybrid approach combining CRFs with a dictionary to recognize chemical compounds in natural language texts, including trivial names, drugs, abbreviations, molecular formulas and International Union of Pure and Applied Chemistry entities. ChemSpot get an F1 measure of 68.1% on the SCAI corpus and it is higher by 10.8% than OSCAR4, a MEMM based system.

Authors in [11] used CRFs to recognize genes and proteins such as PROTEIN, DNA, RNE, CELL-LINE and CELL-TYPE in biomedical abstracts. The features set was used to label a piece of text in abstracts including the word shape of text, neighbor words and their POS tags, 17 orthographic features (ALPHA, HASHDASH) and typical semantic features of each class. The result gets an F1 score of 70%. Later, authors in [19] improve the result of [11] by using CRFs with a richer feature set including shallow syntactic features, orthographical features, word shape features, prefix and suffix features, context features and POS features. The system achieved an F1 score of 71%.

Beside that, CRFs are also applied successfully on the Vietnamese NLP tasks. Authors in [18] used CRFs and the MEMM to solve the POS tagging task in Vietnamese. In spite of few of simple features, the result of CRFs method is very promising with 91.98% at the F1 score. In the average, it is higher 0.7% than the MEMM method.

Due to those effectiveness and successes of CRFs, in our research, we use CRFs combining with different kinds of feature sets to recognize the logical parts in legal documents. Moreover, we also applied MIRA, a large-margin online algorithm which

has been employed successfully for a number of classification tasks in NLP. CRFs and MIRA are discriminative learning methods which are suitable for tasks such as classifications by building a model which can separate new instances into given classes based on the conditional probability distribution P(y|x) where y are classes and x are observations. The detail of CRFs algorithm is described in [14] and the details of MIRA is described in [20].

3 Methods

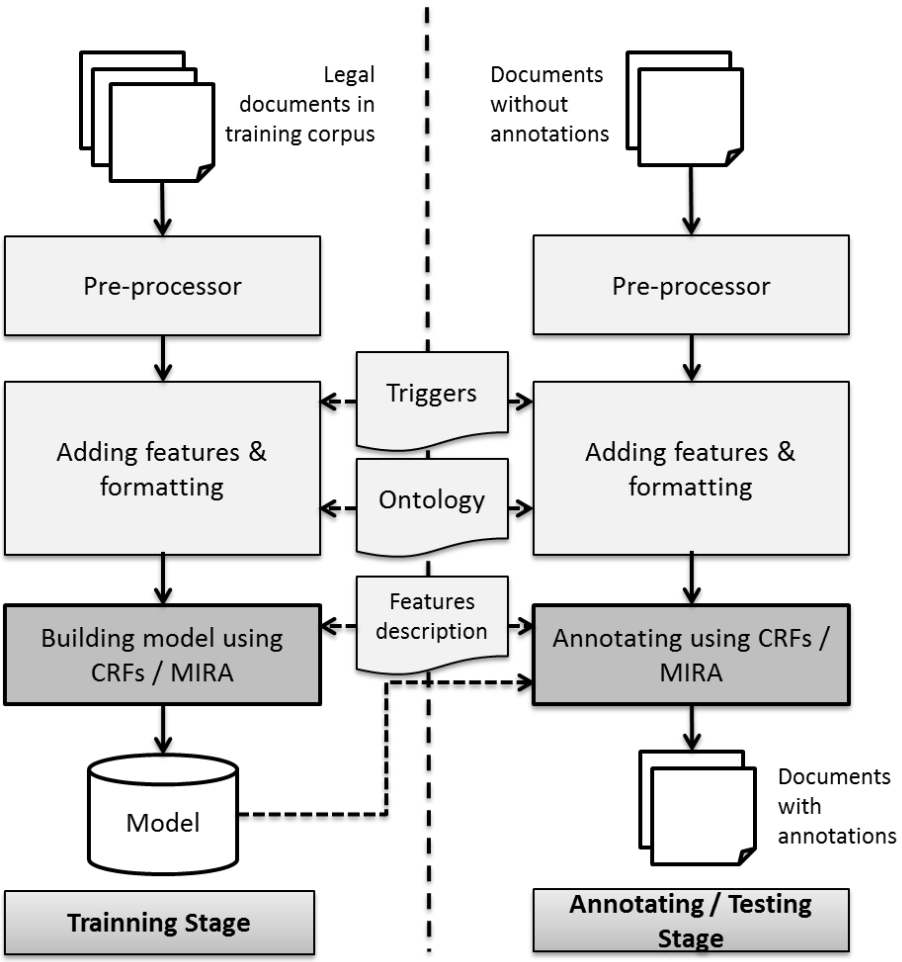


Fig. 1 The system architecture for recognizing logical parts in legal documents

The idea of the our proposed approach is using popular discriminative learning methods include Conditional Random Fields and Margin Infused Relaxed Algorithm with different feature sets to build a system of recognition logical parts in legal texts.

The architecture of this system includes 2 following stages: (1) Training stage: In this stage, we will build a knowledge model by learning from annotated legal documents (called the training corpus) by using CRFs and MIRA learning algorithms and feature sets. (2) Annotation / Testing Stage: This stage will identify logical parts in new legal documents by passing these documents through the model built in the training stage. In this stage, if we want to evaluate the performance of the system, we can compare the output of our system with the judges of law experts.

Main components of this system will be showed in Fig. 1 Section 3.1 describes the input data format of each stage. Section 3.2 presents feature sets used for the machine-learning algorithm. The details of components will be described in Section 3.3.

3.1 The input data

The input data of the training stage is a collection of annotated legal documents written in XML format called the training corpus. The **<rule> </rule>** tags are used to separate law articles in these documents. Logical parts in these documents are marked by suitable tags. The corpus use tags in {**<a>**, ** **, **<s></s>**, **<t></t>**, **<d></d**} to distinguish assumption, provision, sanction, term and definition parts in law articles. Several examples of the training corpus are shown in Table 1.

The input data of the testing/annotation stage is new legal documents. They are also written in XML format as documents in the training corpus. However, there are no logical parts identified.

Table 1. Examples of the training corpus

#1	<rule id="2567-1"> <a>Nhà nước Cộng hoà xã hội chủ nghĩa Việt Nam khuyến khích các nhà đầu tư nước ngoài đầu tư vào Việt Nam trên cở sở tôn trọng độc lập, chủ quyền và tuân thủ pháp luật của Việt Nam, bình đẳng và các bên cùng có lợi . <a>Nhà nước Cộng hoà xã hội chủ nghĩa Việt Nam bảo hộ quyền sở hữu đối với vốn đầu tư và các quyền lợi hợp pháp khác của nhà đầu tư nước ngoài ; tạo điều kiện thuận lợi và quy định thủ tục đơn giản, nhanh chóng cho các nhà đầu tư nước ngoài đầu tư vào Việt Nam . </rule>
#2	<rule> <a>Thực hiện góp vốn và cấp giấy chứng nhận phần vốn góp 1. <a>Thành viên hợp danh và <a>thành viên góp vốn phải góp đủ và đúng hạn số vốn như đã cam kết . 2. <a>Thành viên hợp danh không góp đủ và đúng hạn số vốn đã cam kết gây thiệt hại cho công ty** <s>**phải chịu trách nhiệm bồi thường thiệt hại cho công ty**</s>**. </rule>
#3	<rule> <a>Giải thích từ ngữ 1. **<t>**Đơn vị kinh doanh</t> <d>là doanh nghiệp, hợp tác xã, hộ kinh doanh tham gia kinh doanh vận tải bằng xe ô tô</d>. 2. <t>Kinh doanh vận tải bằng xe ô tô</t> <d>là việc sử dụng xe ô tô vận tải hành khách, hàng hóa có thu tiền</d>. </rule>

3.2 Feature sets

This problem is a kind of Natural Language Understanding tasks, so we also exploit languges feature set like many other NLP tasks. Four feature sets used to build the annotation model as following:

- **WORD features**: using the current word at the observation point and his neighbors in a 5-word-window in size (including the current word, 2 previous words and 2 next words).
- **POS tag features**: the part of speech feature of a word can reveal which kind of logical parts it belongs to. For examples, in many cases, the beginning of assumption part is an adverb or the beginning of the provision part is a verb. In our system, the POS tag feature set includes part of speech of the current word and neighbor words.
- **TRIGGER features**: to add semantic features into the training corpus, we use a gazetteer containing list of typical words of each logical part. For examples, some words or phrases such as "Trong trường hợp" (in case of), "Nếu" (If) are signals to identify the beginning of assumption parts. The other words such as "phải" (must), "được phép" (is allowed), "có quyền" (have the rights doing something) are signals indicated the beginning of provision parts. Some words such as "bị", "sẽ bị" (be + past participle Verb) is signal to recognize the beginning of sanction parts while "là" (is) is the special verb to separate legal terms and their definition. Our gazetteer contains **140** trigger words for this feature set.
- **ONTOLOGY features**: we use a small ontology [3], which contains Vietnamese legal terms in order to add semantic features into the input data. This ontology contains 299 entities which belong to 6 categories. Ontology features are signals to recognize logical parts in legal texts, especially terms and definitions. Ontology features are used to add semantic information to words, we hope that will improve the result of recognizing logical parts.

3.3 Components

- **Pre-processing of the input data:**

The input data will be normalized to increase the performance of the word segmentation tool. The word segmentation tool [15] does not recognize single words with hyphen (-) or splash (/) characters such as 1/3, 22/2006/NĐ-CP, phút/lượt, công-ten-nơ, etc. Therefore, we use some regular expressions to transform these words into new ones as follows: 1/3 → 1-3, 22/2006/NĐ-CP → 222006NĐCP, phút/lượt → phút-lượt, công-ten-nơ → congtenno, etc. Beside that, we normalize punctuations and redundant white spaces.

- **Adding features and formatting the input data:**

We use vnTagger [15] as a tool for the word segmentation and part of speech tagging step. Then, for each word in the input data, we add the trigger type and the on-

tology category information if it matches with a word in the trigger words list or the ontology.

For the convenience of CRFs and MIRA algorithms, we represent the corpus use the IOB2 notation in which **B-** and **I-** tags are used to annotate the begin word and inside words of each logical part. The detail of IOB2 tags used in our system is showed in Table 2.

Table 2. IOB2 Tags

IOB2 tag	Describe
B-A, I-A	The beginning word and inside words of each assumption part.
B-P, I-P	The beginning word and inside words of each provision part.
B-S, I-S	The beginning word and inside words of each sanction part.
B-T, I-T	The beginning word and inside words of each legal term.
B-D, I-D	The beginning word and inside words of each defintion of legal terms.
O	Do not belong to any logical parts.

Finally, the input data file will be formatted according to the requirement of *CRF++*, an implementation of both CRFs and MIRA used in our experiments. In this file, each word will be presented in each line as the following format:

<Word> <POS > <Ontology> <Trigger> <IOBTag>

An example of the input file is show in Fig. 2.

```
Hai             M    0    U    B-A
bên             N    0    U    I-A
hoặc            CC   0    U    I-A
nhiều           A    0    U    I-A
bên             N    0    U    I-A
được            R    0    Pp   B-P
hợp_tác         V    0    U    I-P
với             E    0    U    I-P
nhau            N    0    U    I-P
để              E    0    Adv  I-P
thành_lập       V    0    Pv   I-P
doanh_nghiệp    N    2    Ag   I-P
liên_doanh      V    0    U    I-P
tại             E    0    U    I-P
Việt_Nam        Np   5    U    I-P
trên            E    0    Adv  B-A
cơ_sở           N    6    U    I-A
hợp             V    0    U    I-A
đồng            N    0    U    I-A
liên_doanh      V    0    U    I-A
.               .    0    Os   O
```

Fig. 2 The input data format for CRF++

- **Building the model**

In this step, we use an implementation of CRFs and MIRA [16] as a tool to build a model from the training corpus using feature sets described in the previous section.

- **Testing and annotating new legal documents**

The model built in the previous step will help recognize logical parts in new legal documents. The learning algorithm will retrieve the knowledge represented in the model to annotate logical parts in new legal documents. Before passing these documents through the annotating step, they are also applied steps as the same as the training stage. After logical parts are identified, we will compare the result of the system with the annotation of law experts to evaluate the model. If the performance of this model is acceptable, it can be used in real applications.

4 Experiments

4.1 Data sets

The corpus used in our experiments is a part of Vietnam Business Law documents. It belongs to the project supported by Vietnam National University HoChiMinh City (VNU-HCM) and was constructed manually by 3 annotators. It consists of 132 law articles, 840 sentences, and 21205 words. It contains 1720 logical parts. The statistics of this corpus is described Table 3. Several examples of this corpus are shown in Section 3.1.

Table 3. The statistic of the corpus

Logical part	Count	Percentage	BIO tag	Count
Assumption	966	56.1%	B-A	966
			I-A	7006
Provision	562	32.7%	B-P	562
			I-P	7545
Sanction	56	3.3%	B-S	56
			I-S	541
Term	68	3.95%	B-T	68
			I-T	269
Definition	68	3.95%	B-D	68
			I-D	1765
Total	**1720**			**18846**

4.2 Experiments and comments

- **Feature sets configuration and evaluation methods**

We have conducted many experiments on different feature set configuration including both using a single kind of feature set and combination of different kinds of feature sets. In four first experiments, we just used single kind of feature set to evalu-

ate the effectiveness of each feature type. From experiment 5 to 8, we use some combinations of different feature set kinds to improve the performance of the system.

The settings of our experiments are shown in Table 4.

Table 4. Feature sets configuration

	Experiment	WORD	POS	TRIGGER	ONTOLOGY
SINGLE	1	X			
	2		X		
	3			X	
	4				X
COMBINATION	5	X	X		
	6	X		X	
	7	X	X	X	
	8	X	X	X	X

For each experiment, we apply the cross validation k-fold (k=10) method to evaluate each model. The result of these experiments including Precision, Recall and F1 measure is described in Table 5. The overall result is shown in Table 6.

Table 5. Experimental result in logical parts annotation on different feature sets

Logical part	Exp.	FEATURE SET	CRF			MIRA		
			P	P	F1	P	R	F1
A	1	WORD	0.790	0.683	0.733	0.676	0.646	0.661
	2	POS	0.738	0.570	0.643	0.537	0.403	0.460
	3	TRIGGER	0.779	0.598	0.677	0.643	0.540	0.587
	4	ONTOLOGY	0.684	0.394	0.500	0.372	0.147	0.211
	5	WORD-POS	0.816	0.706	0.757	0.707	0.689	0.698
	6	WORD-TRIGGER	0.812	0.706	0.755	0.697	0.695	0.696
	7	WORD-POS-TRIGGER	0.809	0.715	0.759	0.695	0.686	0.691
	8	WORD-POS-TRIGGER-ONTOLOGY	0.803	0.707	0.752	0.681	0.695	0.688
P	1	WORD	0.690	0.633	0.660	0.543	0.569	0.556
	2	POS	0.481	0.402	0.438	0.295	0.244	0.267
	3	TRIGGER	0.615	0.511	0.558	0.544	0.498	0.520
	4	ONTOLOGY	0.258	0.141	0.182	0.084	0.046	0.060
	5	WORD-POS	0.707	0.648	0.676	0.580	0.601	0.590
	6	WORD-TRIGGER	0.696	0.639	0.666	0.560	0.591	0.575
	7	WORD-POS-TRIGGER	0.708	0.657	0.681	0.582	0.616	0.599
	8	WORD-POS-TRIGGER-ONTOLOGY	0.709	0.664	0.686	0.560	0.603	0.581
S	1	WORD	0.789	0.536	0.638	0.531	0.464	0.495
	2	POS	0.667	0.357	0.465	0.276	0.143	0.188
	3	TRIGGER	0.737	0.500	0.596	0.500	0.500	0.500
	4	ONTOLOGY	-	-	-	-	-	-
	5	WORD-POS	0.795	0.554	0.653	0.500	0.571	0.533
	6	WORD-TRIGGER	0.780	0.571	0.660	0.500	0.536	0.517
	7	WORD-POS-TRIGGER	0.795	0.625	0.700	0.607	0.607	0.607
	8	WORD-POS-TRIGGER-ONTOLOGY	0.814	0.625	0.707	0.500	0.536	0.517
T	1	WORD	1.000	0.676	0.807	0.907	0.721	0.803
	2	POS	0.865	0.662	0.750	0.613	0.721	0.662
	3	TRIGGER	0.904	0.691	0.783	0.781	0.735	0.758
	4	ONTOLOGY	0.118	0.118	0.118	-	-	-
	5	WORD-POS	1.000	0.662	0.796	0.942	0.721	0.817
	6	WORD-TRIGGER	0.977	0.632	0.768	0.857	0.706	0.774

Logi-cal part	Exp.	FEATURE SET	CRF			MIRA		
			P	P	F1	P	R	F1
	7	WORD-POS-TRIGGER	0.957	0.662	0.783	0.920	0.676	0.780
	8	WORD-POS-TRIGGER-ONTOLOGY	0.957	0.647	0.772	0.839	0.691	0.758
D	1	WORD	0.935	0.632	0.754	0.574	0.515	0.543
	2	POS	0.811	0.632	0.711	0.361	0.574	0.443
	3	TRIGGER	0.846	0.647	0.733	0.569	0.603	0.586
	4	ONTOLOGY	0.186	0.191	0.188	-	-	-
	5	WORD-POS	0.933	0.618	0.743	0.536	0.441	0.484
	6	WORD-TRIGGER	0.909	0.588	0.714	0.403	0.397	0.400
	7	WORD-POS-TRIGGER	0.894	0.618	0.730	0.463	0.368	0.410
	8	WORD-POS-TRIGGER-ONTOLOGY	0.913	0.618	0.737	0.456	0.382	0.416

Table 6. The average result of all logical parts

	FEATURE SET	CRF			MIRA			$F1_{CRF} - F1_{MIRA}$
		P	R	F1	P	R	F1	
1	WORD	0.766	0.660	0.709	0.629	0.613	0.621	0.088
2	POS	0.655	0.515	0.576	0.443	0.362	0.398	0.178
3	TRIGGER	0.728	0.572	0.641	0.606	0.535	0.569	0.072
4	ONTOLOGY	0.478	0.280	0.353	0.241	0.098	0.139	0.214
5	WORD-POS	0.786	0.677	0.728	0.657	0.648	0.653	0.075
6	WORD-TRIGGER	0.778	0.672	0.721	0.637	0.644	0.641	0.080
7	WORD-POS-TRIGGER	0.781	0.687	0.731	0.652	0.648	0.650	0.081
8	WORD-POS-TRIGGER-ONTOLOGY	0.778	0.684	0.728	0.631	0.647	0.639	0.089

- **Comments:**

The result is very encouraged. The best model learned by CRFs (in experiment 7) gets a precision measure of 78.1% and a recall measure of 68.7%, F1 measure of 73.1%.

In four types of feature sets, word surface is a very important feature for recogizing logical parts in legal documents. Word surface features give the best result in both CRFs and MIRA algorithms, compared with other kinds of feature sets independently. In the other hand, the result of experiments used ontology features is very low due to the small size of the ontology used.

In almost every experiment, the combination of word face features with other features improves the performance of the system. For example, the average F1 score in experiment #7 is nearly 3% higher than the average F1 score in experiment #1 (see Table 6).

Although it has only a few terms and definition parts in the training corpus, the result of recognition of terms and definitions is very impressive. The precision score of recognition term and definition parts are nearly 100% (See Table 6). That is understandable, because terms and definitions in legal texts are written in special and consistent styles. For example, terms and definitions are always separated by the special verb "là" (is). However, due to the small size of training corpus, the recall score falls short of our expectation.

Compared to CRFs, MIRA is much faster to train however the performance of MIRA is lower. In all experiments, the precision and recall of MIRA are lower than CRFs.

Table 7. Comparations with the previous system based on F1-score

Logical part	**Our result in Experiment #8 (F1-score) (%)**	**Bui Thinh et al [2] (F1-score) (%)**	
Assumption	75.2	64.37	+10.83%
Provision	68.6	64.15	+4.45%
Sanction	70.7	75.76	-5.06%
Term	77.2	N/A	
Defintion	73.7	N/A	

Comparing with the pervious system (see Table 7), our best model gets the better result on recognition of assumption parts (+10.8%) and provision part (+4.5%). The number of sanction parts in the training corpus is small led to the result of recognizing sanction parts is lower (-5%). If we extent the training corpus, we might get a better result on the recognition of these kinds of logical parts.

5 Conclusions and future works

In this paper, we presented a new approach to solve the task of recognizing logical parts in Vietnamese Legal Text. We used the CRFs and MIRA with different feature sets such as word features, POS tags features, trigger features and ontology features. Experiments showed that the combination of word-features with other kinds of features improve the result significantly. However, because the size of corpus is small, the result of our system falls short of our expectation for some logical parts. In the future, we can extend this corpus, use a bigger ontology and explore some feature sets such as shallow syntax features and orthographic features. Besides, we can use the hybrid method of dictionaries, rule-based and machine learning approaches to improve the performance of our system.

Acknowledgments

This work is a part of the project supported by Vietnam National University, Ho Chi Minh City.

References

[1] Katayama, T. 2007. Legal engineering - An engineering approach to laws in e-society age. In Proceedings of the 1st International Workshop on Juris-Informatics (JURISIN'07)

[2] Bui Thinh, Ho Quoc, "An Approach for Automatically Structuring Vietnamese Legal Text". In IALP 2014, International Conference on Asia Language Processing, 2014 (accepted)

[3] Bui Thinh, Nguyen Son, Ho Quoc, “Towards a Conceptual Search for Vietnamese Legal Text”. In CISIM 2014, 13th International Conference on Computer Information Systems and Industrial Management Applications, Vietnam, 2014 (accepted).

[4] N. X. Bach, N. Le Minh, and A. Shimazu, “Recognition of Requisite Part and Effectuation Part in Law Sentences,” in Proceedings of (ICCPOL), 2010, pp. 29–34.

[5] N. X. Bach, N. L. Minh, T. T. Oanh, and A. Shimazu, “A Two-Phase Framework for Learning Logical Structures of Paragraphs in Legal Articles,” ACM Transactions on Asian Language Information Processing (TALIP), vol. 12, no. 1, p. 3, 2013.

[6] N. X. Bach, L. M. NGUYEN, and A. Shimazu, “RRE task: The task of recognition of requisite part and effectuation part in law sentences,” International Journal of Computer Processing Of Languages, vol. 23, no. 02, pp. 109–130, 2011.

[7] L.-M. Nguyen, N. X. Bach, and A. Shimazu, “Supervised and semi-supervised sequence learning for recognition of requisite part and effectuation part in law sentences,” in Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, 2011, pp. 21–29.

[8] L. F. Rau, “Extracting company names from text,” in Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on, 1991, vol. 1, pp. 29–32.

[9] K. Riaz, “Rule-based named entity recognition in Urdu,” in Proceedings of the 2010 Named Entities Workshop, 2010, pp. 126–135.

[10] T. Rocktaschel, M. Weidlich, and U. Leser, “ChemSpot: a hybrid system for chemical named entity recognition,” Bioinformatics, vol. 28, no. 12, pp. 1633–1640, Jun. 2012

[11] B. Settles, “Biomedical named entity recognition using conditional random fields and rich feature sets,” in Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, 2004, pp. 104–107.

[12] L. Yao, C. Sun, Y. Wu, X. Wang, and X. Wang, “Biomedical named entity recognition using generalized expectation criteria,” International Journal of Machine Learning and Cybernetics, vol. 2, no. 4, pp. 235–243, Dec. 2011.

[13] N. Ponomareva, P. Rosso, F. Pla, and A. Molina, “Conditional random fields vs. hidden markov models in a biomedical named entity recognition task,” in Proc. of Int. Conf. Recent Advances in Natural Language Processing, RANLP, 2007, pp. 479–483.

[14] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” 2001.

[15] L Hong Phuong, vnTagger -- Vietnamese part-of-speech tagging, http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTagger

[16] Kudo, Taku. "CRF++: Yet another CRF toolkit." Software available at http://crfpp. sourceforge. net (2005).

[17] Giáo trình Lý Luận Nhà nước và pháp luật, Đại học Luật Hà Nội.

[18] Nguyen, Cam-Tu, et al. "Vietnamese word segmentation with CRFs and SVMs: An investigation." Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC 2006). 2006.

[19] Sun, Chengjie, et al. "Rich features based Conditional Random Fields for biological named entities recognition." Computers in Biology and Medicine 37.9 (2007): 1327-1333.

[20] Crammer, K., Singer, Y. (2003): Ultraconservative Online Algorithms for Multiclass Problems. In: Journal of Machine Learning Research 3, 951-991.

Translating Simple Legal Text to Formal Representations

Shruti Gaur, Nguyen H. Vo, Kazuaki Kashihara, and Chitta Baral

Arizona State University, Tempe, Arizona, USA
{shruti.gaur,nguyen.h.vo,kkashiha,chitta}@asu.edu

Abstract. Various logical representations and frameworks have been proposed for reasoning with legal information. These approaches assume that the legal text has already been translated to the desired formal representation. However, the approaches for translating legal text into formal representations have been mostly focused on inferring facts from text or translating it to a single representation. In this work, we use the NL2KR system to translate legal text into a wide variety of formal representations. This will enable the use of existing logical reasoning approaches on legal text(English), thus allowing reasoning with text.

Keywords: natural language processing ·natural language understanding ·natural language translation

1 Introduction and Motivation

One of the tasks of the Competition on Legal Information Extraction and Entailment [1] consists of finding whether a given statement is entailed by the given legal articles or not. This is similar to the Recognition of Textual Entailment(RTE) challenge [3]. It has been observed that the systems that rely on statistical and machine learning methods perform much better in the RTE challenge than systems which rely on logical reasoning. However, we believe that statistical and machine learning methods do not offer much explanation about why a certain sentence is entailed or not, hence providing little insight into the cause of entailment. In this respect, we consider the approaches based on logical reasoning such as the work by Bos and Markert[7] to be more promising.

There have been several works that propose logical representations and logics for representing and reasoning with legal information[15,13,21,11,12]. Reasoning rules and frameworks assume that the information given in the form of natural language can somehow be understood and represented in the required form. However, current methods to convert legal text to formal representations[8,17,4,18] either focus on extracting important facts or are not generalizable to a wide variety of representations. Currently, there is no consensus on a single representation to express legal information. Therefore, a system that can translate natural language to a wide variety of formal languages, depending on the application, is desired. In this paper we show how our NL2KR system can be used for translation of simple legal sentences in English to various formal representations. This will facilitate reasoning with various frameworks.

2 Related Work

Some approaches to translate text into formal representations focus on extraction of specific facts from the text. For example, Lagos et al.[17] present a semi-automatic method to extract specific information such as events, characters, role, etc. from legal text by using the Xerox Incremental Parser(XIP)[2]. The XIP performs preprocessing, named entity extraction, chunking and dependency extraction, and combination of dependencies to create new ones. Bajwa et al.[4] propose an approach to automatically translate specification of business rules in English to Semantic Business Vocabulary and Rules(SBVR). Their method is essentially a rule-based information-extraction approach, which identifies SBVR elements from text. The goal of other approaches like the work by McCarty[18] is to obtain a semantic interpretation of the complete sentence. This approach uses the output from the state-of-the-art statistical parser to obtain a semantic representation called Quasi-Logical Form(QLF). QLF is a rich knowledge representation structure which is considered an intermediate step towards a fully logical form.

There have been similar efforts in other languages. Nakamura et al.[20] present a rule-based approach to convert Japanese legal text into a logical representation conforming to Davidsonian style. They ascertain the structure of legal sentences and identify cue phrases that indicate this structure, by manually analyzing around 500 sentences. They also define transformation rules for some special occurrences of nouns and verbs. In their subsequent work[16], they propose a method to resolve references that point to other articles or itemized lists, by replacing them with the relevant content.

As mentioned in the previous section, different legal reasoning frameworks expect input in different logical representations. There is currently no single representation that has been unanimously considered the de-facto standard for legal text. Therefore, we need a system that can translate natural language to a particular representation depending on the application.

3 The NL2KR Framework

NL2KR is a framework to develop translation systems that translate natural language to a wide variety of formal language representations. It is easily adaptable to new domains based on the training data supplied. It is based on the algorithms presented in Baral et al.[5].

The workflow using the NL2KR systems consists of two phases: 1.)learning and 2.)translation, as shown in Fig.1. In the learning phase, the system takes training data, an initial dictionary and any optional syntax overrides, as inputs. The training data consists of a number of natural language sentences along with their formal representations in the desired target language. The initial dictionary (or lexicon) contains meanings of some words. The dictionary is manually supplied to the system. Using these inputs, NL2KR tries to learn the meanings of as many words as possible. Thus, the output of the learning phase is an updated dictionary which contains the meanings of all newly

learned words. The translation phase uses the dictionary created by the learning phase to translate previously unseen sentences. At the core of NL2KR are two very elegant algorithms, Inverse Lambda and Generalization, which are used to find meanings of unknown words in terms of lambda(λ) expressions.

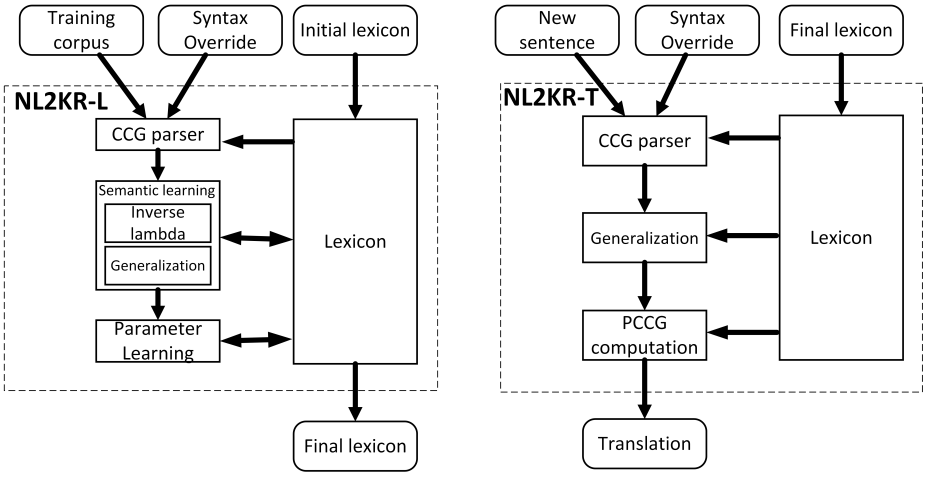


Fig. 1. The NL2KR system showing learning(left) and translation(right)

NL2KR is inspired by Montague's approach [19]. Every word has a λ expression meaning and the meaning of a sentence is successively built from the combination of the λ expressions of words according to the rules of combination in Lambda(λ) calculus [9]. The order in which the words should be combined is given by the parse tree of the sentence according to a given Combinatory Categorial Grammar(CCG)[22]. As an example illustrating this approach, consider the sentence "John loves Mary" shown in Table 1. The CCG category of "loves" is (S\NP)/NP. This means that this word takes arguments of type NP(noun-phrase) from the left and the right, to form a complete sentence. From the CCG parse, we observe that "loves" and "Mary" combine first and then their combination combines with "John" to form a complete parse. The λ expression corresponding to "loves" is $\#y.\#x.loves(x,y)$[1], which means that this word takes two inputs, $\#x$ and $\#y$ as arguments and the application of this

Table 1. Example

John	loves	Mary
NP	$(S\backslash NP)/NP$	NP
$john$	$\#y.\#x.loves(x,y)$	$mary$
	$S\backslash NP$	
	$\#x.loves(x,mary)$	
	S	
	$loves(john,mary)$	

[1] # is used in place of λ to enable typing into a terminal

word to the arguments results in a λ expression of the form $loves(x, y)$.

The close correspondence between CCG syntax and λ calculus semantics is very helpful in applying this method. In the first step, the λ expression for "loves" is applied to "Mary", with the former as the function and the latter as the argument, in accordance with CCG categories. This application, denoted as $\#y.\#x.loves(x, y)@mary$ results in $\#x.loves(x, mary)$. Proceeding this way, the meaning of the sentence is generated in terms of λ expressions. This is a very elegant way to model semantics and has been widely used [6,23,10,5]. The problem, however, is that for longer sentences, λ expressions become too complex for even humans to figure out. This problem is addressed by NL2KR by employing the Inverse Lambda and Generalization algorithms to automatically formulate λ expressions from words whose semantics are known.

Learning Algorithms: The two algorithms used to learn λ semantics of new words are the Inverse Lambda and Generalization algorithms. When the λ expressions of a phrase and that of one of its sub-parts (children in the CCG parse tree) are known, we can use this knowledge to find the λ expression of the unknown sub-part. The Inverse Lambda operation computes a λ expression F such that $H = F@G$ or $H = G@F$ given H and G. These are called Inverse-L and Inverse-R algorithms, respectively. For example, if we know the meaning of the sentence "John loves Mary"(Table1) as $loves(john, mary)$ and the meaning of John as $john$, we can find the meaning of "loves Mary" using Inverse Lambda, as $\#x.loves(x, mary)$. Going further, if we know the meaning of "Mary" as $mary$, we can find the meaning of "loves" using Inverse Lambda.

The Generalization algorithm is used to learn meanings of unknown words from syntactically similar words with known meanings. It is used when Inverse Lambda algorithms alone are not enough to learn new meanings of words or when we need to learn meanings of words that are not even present in the training data set. For example, we can generalize the meaning of the word "likes" with CCG category $(S\backslash NP)/NP)$, from the meaning of "loves", which we already know from the previous example. The meaning of "likes" thus generated will be $\#y.\#x.likes(x, y)$. We will illustrate learning in later sections with the help of examples.

For every sentence in the training set, we first use the CCG Parser to obtain all possible parse trees. Using the initial dictionary supplied by the user, the system assigns all known meanings to the words (at the leaf) in each parse tree. Moving bottom up, it combines as many words with each other as possible (in the order dictated by the parse tree) by performing λ applications. The meaning of each complete sentence is known from the training corpus. We need to traverse top-down from this known translation, while simultaneously traversing bottom up, by filling in missing word or phrase meanings. Meanings of unknown words and phrases are obtained using Inverse Lambda and Generalization, as applicable, until nothing new can be learned.

Dealing with Ambiguity: To deal with ambiguity of words, a parameter learning method [23] is used to estimate a weight for each word-meaning pair such that the joint probability of the training sentences getting translated to their given formal representation is maximized. However, this method might not work in all cases and more complex approaches, possibly involving word sense identification from context, might have to be used. This is a part of future work.

Translation Approach: Given a sentence, we consider all the possible parse trees, consisting of meanings of every word learned by the system or obtained from Generalization algorithm. Then we use Probabilistic CCG(PCCG)[23] to find the most probable tree, according to weights assigned to each word.

Availability: NL2KR is freely available for Windows, Linux and MacOSX systems from http://nl2kr.engineering.asu.edu. It is configurable for different domains and can be adapted to work with a large number of formal representations. A tutorial has also been provided.

4 Translating to Formal Legal Representations

NL2KR can be used to translate sentences into various logical representations, either directly or by using an intermediate language. It can be customized to different domains based on the initial dictionary and training data provided. Several logics have been proposed in the literature for representing legal information[15,13,21,11,12], from which we have selected a few. In this section, we will illustrate the method of creating the initial lexicon and demonstrate how to use the system to learn new word meanings, with respect to these examples. We will start with simple examples and progress to more complicated ones.

4.1 Translating to First Order Logic Type Representations

In this section, we demonstrate translating a sentence from the Competition on Legal Information Extraction and Entailment[1] corpus to a first order logic type representation.
Sentence: Possessory rights may be acquired by an agent.
Translation: $rights(X) \wedge type(X, possessory) \wedge agent(Y) > acquirable(X, Y, may)$
Here >is used to denote implication. The form of an action, for e.g., "acquirable" is $action(X, Y, Z)$. It denotes X(possessory rights) is being acquired by Y(agent) and the type of this action is Z. In the given example, the *acquiring* is a possibility, not an obligation, which is why we use *may* as its type.

Once we provide this training data and other required inputs to the NL2KR learning interface (Fig.2), we can start the Learning process. We will describe how to create inputs for learning in the next sub-section. Apart from the training

data, an initial dictionary is required. It contains a list of words and their meanings in terms of λ expressions. Even if we do not know meanings of some words, we can use the system to figure them out on its own using Inverse Lambda or Generalization algorithms. Fig.2 shows that the system learns the meaning of "rights" automatically using Inverse Lambda.

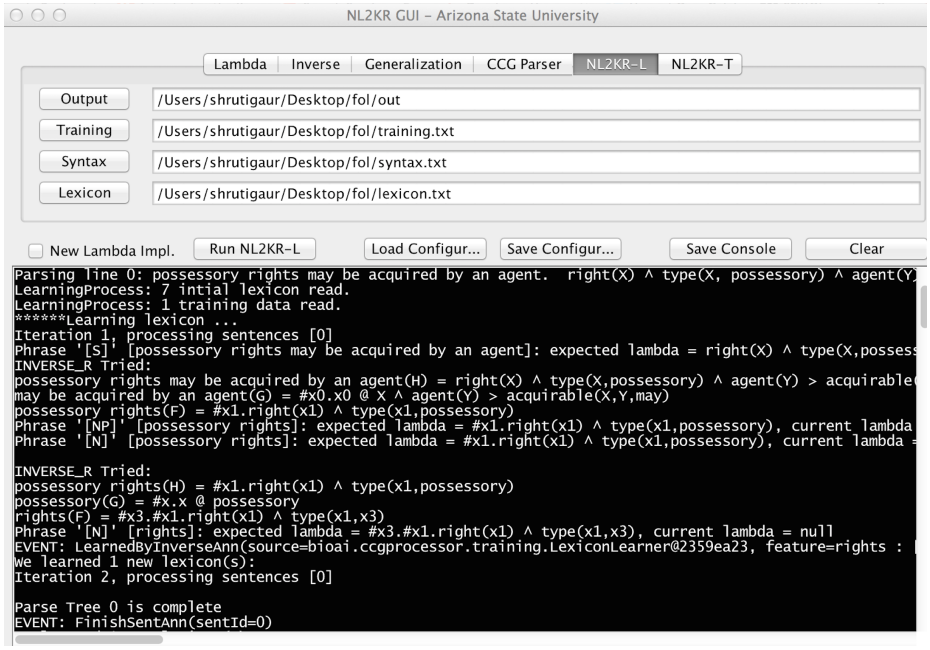


Fig. 2. NL2KR automatically learning the meaning of "rights" using Inverse Lambda Algorithm : feature=rights : [N] : #x3.#x1.right(x1) $\wedge$ type(x1,x3)

4.2 Translating Sentences with Temporal Information

Consider the following sentence and translation, which shows an example of temporal ordering.
Sentence: After the invoice is received the customer is obliged to pay.
Translation: $implies(receipt(invoice, T1) \wedge (T2 > T1), obl(pay(customer, T2)))$

Here $implies(x, y)$ denotes $x \rightarrow y$.The predicate *obl* denotes that the action is an *obligation*(usually marked by words such as obliged to,shall,must, etc.) in contrast to a *possibility*(usually marked by words such as may). T1 and T2 are the instances of time at which the two events occurred.

Words that do not contribute significantly to the meaning of the sentence can be assigned the trivial meaning $\#x.x$ in the dictionary. It is a λ expression that does not affect the meaning of other λ expressions. We can assign it to words such

as: "is","to" and "the" since these do not carry much meaning in this example. Next, we can start entering the meanings that are evident from looking at the target representation. Since "invoice" occurs as itself, we can give it the simple meaning *invoice*(similarly for "customer"). From the representation, we observe that "received" is a function called *receipt* with two arguments, hence we can give it the meaning $\#x.\#t.receipt(x,t)$(similarly for "pay"). *Obliged* is a more complicated function because it takes another function(pay) as its argument and therefore uses @y@t to carry forward the variables in *pay* to the next higher level of the tree, when we obtain the real arguments(customer and T2). Once all these meanings (Table 2) have been supplied, the system can automatically find the meaning of the word "after" using Inverse Lambda algorithm(Fig.3). This is remarkable from the perspective that the meaning of "after" looks complicated and it might be tedious for users to supply such meanings manually in the initial lexicon. However, this demonstrates one of the advantages of using NL2KR. If we look at the meaning of "after" it makes intuitive sense. The λ expression #x14.#x13.implies(x14 @ T1 $\wedge$ T2 >T1,x13 @ T2) means that "after" is a λ function which takes two inputs:x14 and x13, where the first input event(x14) occurs at time T1, T2>T1, the second input event(x13) occurs at time T2 and x14 implies(or leads to) x13. Hence, we were able to learn a significantly complicated meaning automatically by providing relatively simple λ expressions in the initial dictionary.

Table 2. λ expressions and CCG categories in the initial dictionary for the sentence "After the invoice is received the customer is obliged to pay."

Word	Syntax	Meaning
invoice	N	invoice
is	(S\NP)/NP	#x.x
received	NP &#x.#t.receipt(x,t)	
customer	N	customer
obliged	NP/NP	#x.#y.#t.obl(x@y@t)
to	NP/(S\NP)	#x.x
pay	S\NP	#x.#t.pay(x,t)
the	NP/N	#x.x

4.3 Translating to Temporal Deontic Action Laws

Giordano et al. [15] have defined a Temporal Deontic Action Language for defining temporal deontic action theories, by introducing a temporal deontic extension of Answer Set Programming(ASP) combined with Deontic Dynamic Linear Time Temporal Logic(DDLTL). This language is used for expressing domain description laws, for e.g., action laws, precondition laws, causal laws, etc., which describe the preconditions and effects of actions. It is also used for expressing obligations, for e.g., achievement obligations, maintenance obligations, contrary

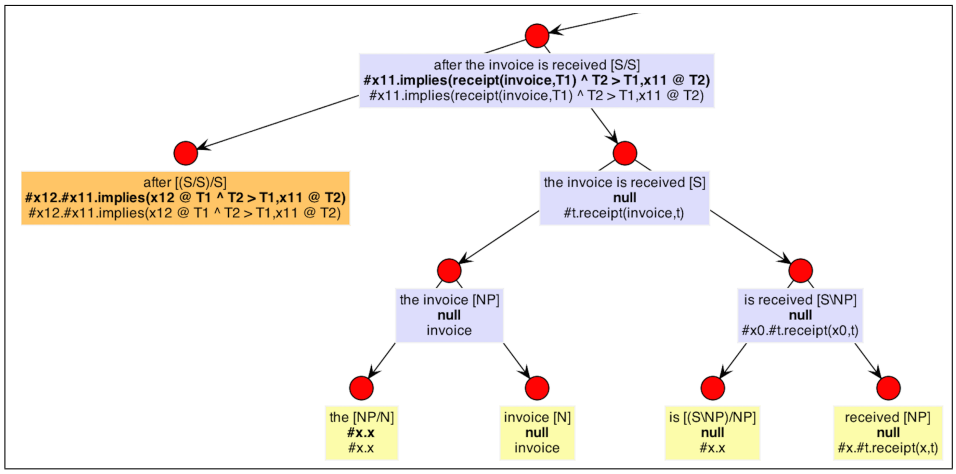


Fig. 3. NL2KR auomatically learning the meaning of "after" using Inverse Lambda Algorithm : feature=after : [(S/S)/S] : #x14.#x13.implies(x14 @ T1 ∧ T2 >T1,x13 @ T2)

to duty obligations, etc. In this paper we take examples of several domain description laws from the paper and demonstrate how to translate them automatically from natural language to the Deontic action language, using NL2KR.

Since NL2KR does not support some special symbols used in the Temporal Deontic Action Language, we first use NL2KR to convert the natural language sentences to an intermediate representation which is directly convertible to the Temporal Deontic Action Language. Then using the one-to-one correspondence between the intermediate language and the action language, we can obtain the desired representation. In the Intermediate representation shown below, we have defined the predicate $creates(x, y)$, which means x creates y. We have also changed the representation of *until* to have two parameters a and b denoting "a until b" as defined by the paper.

Action Law:
Sentence: The action *accept_price* creates an obligation to pay.
Translation: $[accept\_price]\mathbf{O}(\top U < pay > \top)$
Intermediate: creates(action(accept_price),O(until(a(T),b(pay,T))))

The NL2KR learning component can be used to make the system learn words and meanings from this sentence. The iterative learning process is depicted in the screenshot in Figure 4. We start by giving meanings of simple words first. We give trivial meanings $\#x.x$ to "the","an" and "to", because they are not very meaningful. Next, we guess the meanings of words from the target representation. Since "action" is a function that accepts a single argument, we give it the meaning $\#x.action(x)$. Similarly, "accept_price" which occurs as itself is just given the meaning *accept_price*. Similarly, Obligation, O is a function also, but it contains more structure, which can be obtained from the target repre-

sentation. We interpret an obligation to also have an implicit notion of time by having "until" embedded in its meaning. However, the verb "pay" should be replaceable, because there can be other sentences like "The action *accept_price* creates an obligation to ship". Therefore, we leave it as a variable input. The word "pay" could have been simply *pay* but we assign it the meaning $\#x.x@pay$. This is because the node, "an obligation", expects "to pay" to be an argument to it, but their CCG categories dictate otherwise. In cases where there is such inconsistency, we use meanings prefixed with $\#x.x@$ for the function (according to CCG categories), so that their role is *flipped* to that of arguments[2]. The λ expressions and CCG categories of the constituent words are shown in Table 3. After giving these meanings, we find that the meaning of "creates" is obtained automatically by the system using the Inverse Lambda algorithm (Fig.4).

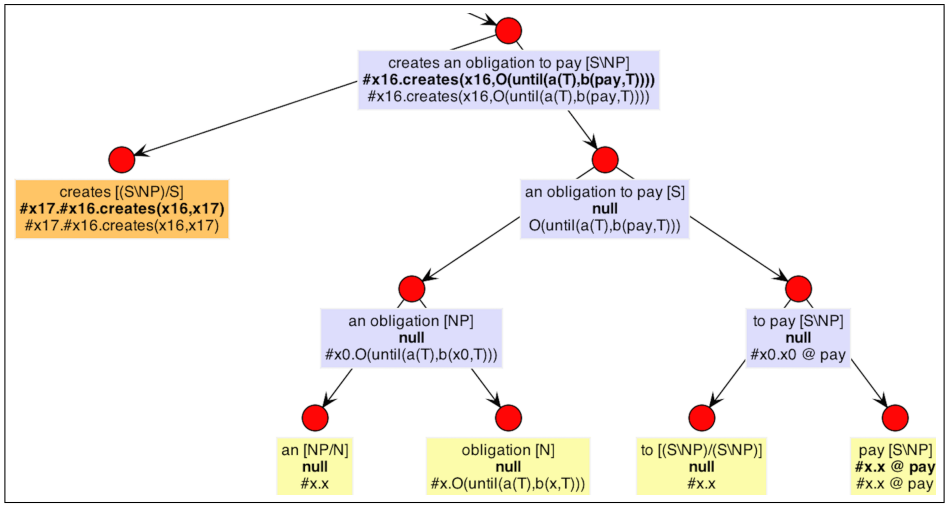


Fig. 4. Screenshot of the Learning Process in NL2KR for the sentence "The action *accept_price* creates an obligation to pay.".The meaning of "creates" is obtained automatically by the system using the Inverse Lambda algorithm

Once the learning process is complete, we can use the Translation component of NL2KR to translate a new sentence. In this case, we use NL2KR to translate the following action law.

Action Law:

Sentence: The action *cancel_payment* cancels the obligation to pay.

[2] Let the required function be A and the required argument be B. Let the CCG-determined function be B and the CCG-determined argument be A. Recall that @ denotes λ application. By giving a meaning of the form $\#x.(x@b)$ to B, and performing application as determined by CCG, we obtain the result as $(\#x.(x@b))@a$ or $a@b$ which is what we wanted.

Table 3. λ expressions and CCG categories for the words in the action law "The action *accept_price* creates an obligation to pay."

Word	Syntax	Meaning
the	NP/N	#x.x
action	N/N	#x.action(x)
accept_price	N	accept_price
an	NP/N	#x.x
obligation	N	#x.O(until(a(T),b(x,T)))
to	(S\NP)/(S\NP)	#x.x
pay	S\NP	#x.x@pay

Translation: $[cancel\_payment]\neg\mathbf{O}(\top U < pay > \top)$
Intermediate: cancels(action(cancel_payment),O(until(a(T),b(pay,T))))

The screenshot of the translation process is shown in Fig.5. We observe that the sentence was automatically translated by the system successfully. In this case this was done by generating the meanings of unknown words ("cancel_payment" and "cancels") using Generalization(Fig.6) from the words learned from the first action law.

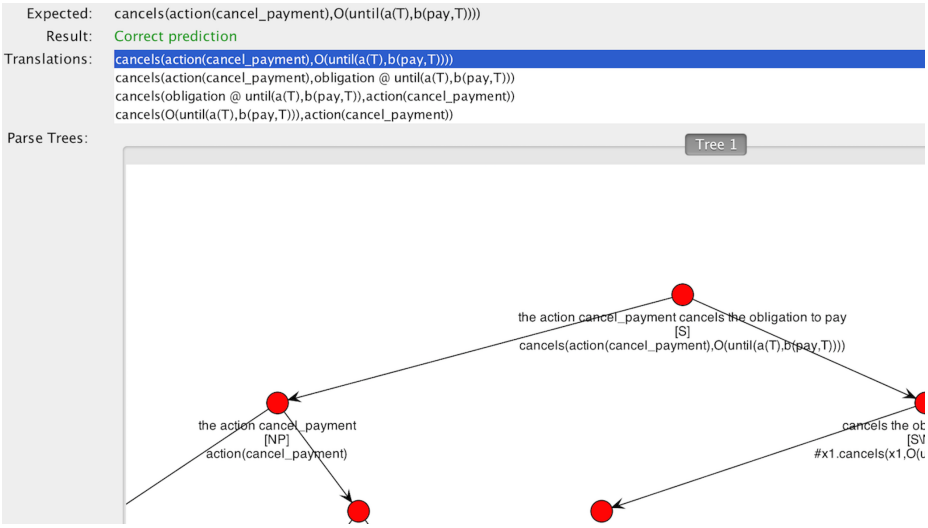


Fig. 5. Screenshot of the Translation Process in NL2KR for the sentence "The action *cancel_payment* cancels the obligation to pay."

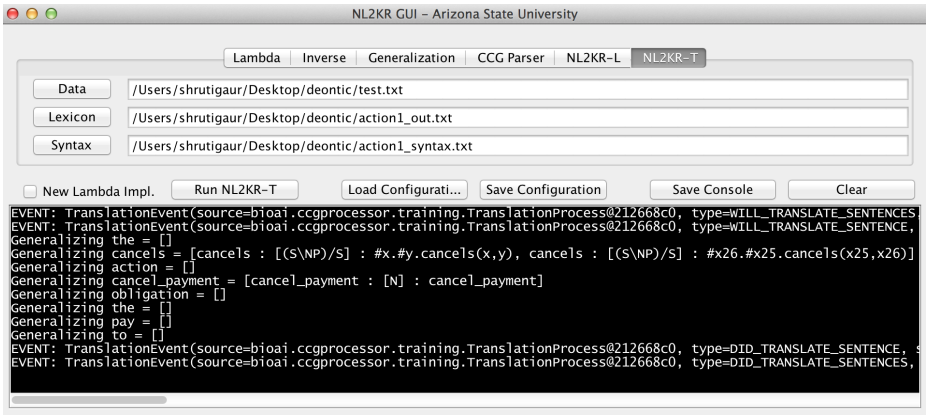


Fig. 6. Generating the meanings of unknown words(cancel_payment and cancels) using Generalization during the Translation Process in NL2KR for the sentence "The action *cancel_payment* cancels the obligation to pay."

4.4 Translating to Temporal Object Logic in REALM

Regulations Expressed as Logical Models(REALM)[14] is a system that models regulatory rules in temporal object logic. The concepts and relationships occurring in this rule are mapped to predefined types and relationships in a Unified Modeling Language(UML) model. Using some examples from this paper, we will show how NL2KR can be used to translate rules specified in natural language to this temporal object logic representation.

As in the previous section, we have created an intermediate representation which can directly be converted to the desired temporal object logic representation. This is needed due to unavailability of certain symbols in NL2KR's vocabulary. We also assume that coreference in sentences has been resolved. For example, in the following sentence, the second occurrence of "bank" has replaced the pronoun "it".
Sentence: Whenever a bank opens an account *bank* must verify customers identity within two_days
Translation: $\Box t_{open}(DoOn_F(bank, open, a) \rightarrow$
$\Diamond t_{verify}(DoInput_F(bank, verify, a.customer.record) \wedge t_{verify} - t_{open} \leq 2_{[day]})$
Intermediate: $implies(g(do(bank, open, a, T1)),$
$f(do(bank, verify, a\_customer\_record, T2)$
$\wedge equals(difference(T2, T1), two\_days)))$

Similar to the previous examples, we use NL2KR to learn unknown words from these sentences. We do not give the meaning of "verify" for the second sentence(which is different from its meaning in the first sentence) but the system is able to figure it out on its own. Moreover, it also generalizes the correct meaning of three_days using the meaning of two_days from the previous sentence.
Sentence: whenever a bank can not verify an identity bank has to close the account within three_days

Translation: $\Box t_{open}(DoOn_F(bank, open, a) \rightarrow$
$\Diamond t_{verify}(DoInput_F(bank, verify, a.customer.record) \wedge t_{verify} - t_{open} \leq 2_{[day]})$
Intermediate:$implies(g(do(bank, verify, a\_customer\_record, T1) \wedge isfalse)$,
$f(do(bank, close, a, T2) \wedge equals(difference(T2, T1), three\_days)))$

We observe that the initial dictionary for this case (Table 4) looks more complicated than the one in Section 4.3. This is because the target language in this case is such that the functions which would have been intuitive according to their natural language meanings, for e.g., "opens", "verify", etc. are not functions but arguments of an artificially created function, "do". It is obvious that the language of REALM was designed for different purposes than that of translation, which is why such a situation exists. Our motivation here was to give the reader an explanation of why some languages are easy for NL2KR to translate, while others are more difficult.

Table 4. Initial Lexicon containing λ expressions and CCG categories for both REALM examples

Word[Syntax]	Meaning
whenever [(S/S)/S]	#y.#x.implies(g(y@T1),f((x@T1)@T2))
a [NP/N]	#x.x
bank [N]	bank
opens [(S\NP)/NP]	#y.#x.#t1.do(x,open,y,t1)
an [NP/N]	#x.x
account [N]	a
must [(S\NP)/(S\NP)]	#x.x
verify [(S\NP)/NP]	#x.x@#x1.#x2.#x3.#x4.#x5.(do(x3,verify,x1,x5) ∧ x2@x4@x5)
customers [NP/N]	#x.x
identity [N]	a_customer_record
within [(NP\NP)/NP]	#z.#y.#x.x@y@#t1.#t2.equals(difference(t2,t1),z)
has [(S\NP)/(S\NP)]	#x.x
to [(S\NP)/(S\NP)]	#x.x
the [NP/N]	#x.x
can [(S\NP)/(S\NP)]	#x.x
close [(S\NP)/NP]	#x.x@#x1.#x2.#x3.#x4.#x5.(do(x3,close,x1,x5) ∧ x2@x4@x5)
not [(S\NP)/(S\NP)]	#y.#x.#t1.(y @ x @ t1 ∧ isfalse)
two_days [N]	two_days

5 Conclusion and Future Work

Although legal text is written in natural language, one needs to do some sort of formal reasoning with it to draw conclusions. The first step to do that is to translate legal text to an appropriate logical language. At present there is

no consensus on a single logical language to represent legal text. Therefore, one cannot develop a translation system targeted to a single language. Thus, a platform that can translate legal text to the desired logical language depending on the application, is needed. We have developed such a system called NL2KR. In this paper, we showed how NL2KR can be used to translate sentences from legal texts in English to various formal representations defined in various works, thereby bridging the gap from language to logical representation and enabling the use of various logical frameworks over the information contained in such texts.

So far we have experimented with a few small sentences picked from the literature on logical representation of legal texts. However, we need to expand this approach to capture nuances of legal texts used in real laws and statutes. Further enhancements are needed in NL2KR to equip it to deal with longer and more complicated sentences. One approach that can be used would involve breaking the sentence into smaller parts and subsequently dealing with each part separately. Such a parser, called L-Parser is available at
http://bioai8core.fulton.asu.edu/lparser.

Acknowledgements: We thank Arindam Mitra and Somak Aditya for their work in developing the L-Parser. We thanks NSF-Datanet and ONR for their support of the development of NL2KR.

References

1. Jurisin legal information extraction and entailment competition. `$http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html$` (2014)
2. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond shallowness: Incremental deep parsing. Nat. Lang. Eng. 8(3), 121–144 (Jun 2002)
3. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. J. Artif. Int. Res. 38(1), 135–187 (May 2010)
4. Bajwa, I., B., L.M., Behzad: Sbvr business rules generation from natural language specification. In: AAAI 2011 Spring Symposium AI for Business Agility. pp. 2–8. San Francisco, USA (2011)
5. Baral, C., Dzifcak, J., Gonzalez, M.A., Zhou, J.: Using Inverse lambda and Generalization to Translate English to Formal Languages. CoRR abs/1108.3843 (2011)
6. Blackburn, P., Bos, J.: Representation and Inference for Natural Language: A First Course in Computational Semantics. Center for the Study of Language and Information, Stanford, CA (2005)
7. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 628–635. HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
8. Brüninghaus, S., Ashley, K.D.: Improving the representation of legal case texts with information extraction methods. In: Proceedings of the 8th International Conference on Artificial Intelligence and Law. pp. 42–51. ICAIL '01, ACM, New York, NY, USA (2001)

9. Church, A.: An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics 58(2), 345–363 (April 1936)
10. Costantini, S., Paolucci, A.: Towards Translating Natural Language Sentences into ASP. In: Faber, W., Leone, N. (eds.) CILC. CEUR Workshop Proceedings, vol. 598. CEUR-WS.org (2010)
11. De Vos, M., Padget, J., Satoh, K.: Legal modelling and reasoning using institutions. In: Proceedings of the 2010 International Conference on New Frontiers in Artificial Intelligence. pp. 129–140. JSAI-isAI'10, Springer-Verlag, Berlin, Heidelberg (2011)
12. Distinto, I., Guarino, N., Masolo, C.: A well-founded ontological framework for modeling personal income tax. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 33–42. ICAIL '13, ACM, New York, NY, USA (2013)
13. Giblin, C., Liu, A.Y., Müller, S., Pfitzmann, B., Zhou, X.: Regulations expressed as logical models (realm). In: Proceedings of the 2005 Conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference. pp. 37–48. IOS Press, Amsterdam, The Netherlands, The Netherlands (2005)
14. Giblin, C., Liu, A.Y., Müller, S., Pfitzmann, B., Zhou, X.: Regulations expressed as logical models (realm). In: Proceedings of the 2005 Conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference. pp. 37–48. IOS Press, Amsterdam, The Netherlands, The Netherlands (2005)
15. Giordano, L., Martelli, A., Dupré, D.T.: Temporal deontic action logic for the verification of compliance to norms in asp. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 53–62. ICAIL '13, ACM, New York, NY, USA (2013)
16. Kimura, Y., Nakamura, M., Shimazu, A.: New frontiers in artificial intelligence. chap. Treatment of Legal Sentences Including Itemized and Referential Expressions — Towards Translation into Logical Forms, pp. 242–253. Springer-Verlag, Berlin, Heidelberg (2009)
17. Lagos, N., Segond, F., Castellani, S., ONeill, J.: Event extraction for legal case building and reasoning. In: Shi, Z., Vadera, S., Aamodt, A., Leake, D. (eds.) Intelligent Information Processing V, IFIP Advances in Information and Communication Technology, vol. 340, pp. 92–101. Springer Berlin Heidelberg (2010)
18. McCarty, L.T.: Deep semantic interpretations of legal texts. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law. pp. 217–224. ICAIL '07, ACM, New York, NY, USA (2007)
19. Montague, R.: English as a Formal Language. In: Thomason, R.H. (ed.) Formal Philosophy: Selected Papers of Richard Montague, pp. 188–222. Yale University Press, New Haven, London (1974)
20. Nakamura, M., Nobuoka, S., Shimazu, A.: Towards translation of legal sentences into logical forms. In: Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence. pp. 349–362. JSAI'07, Springer-Verlag, Berlin, Heidelberg (2008)
21. Riveret, R., Rotolo, A., Contissa, G., Sartor, G., Vasconcelos, W.: Temporal accommodation of legal argumentation. In: Proceedings of the 13th International Conference on Artificial Intelligence and Law. pp. 71–80. ICAIL '11, ACM, New York, NY, USA (2011)
22. Steedman, M.: The Syntactic Process. MIT Press, Cambridge, MA (2000)
23. Zettlemoyer, L.S., Collins, M.: Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In: UAI. pp. 658–666. AUAI Press (2005)

A logic-based system for recognizing textual entailment applied to the bar exam competition

Yusuke Miyao and Ken Satoh

National Institute of Informatics, Japan
{yusuke,ksatoh}@nii.ac.jp

Abstract. This paper describes the result of applying a logic-based system for recognizing textual entailment (RTE) to the bar exam competition at JURISIN 2014. The system has been applied previously to NTCIR RITE, which is the first shared task on RTE in Japanese, achieving near state-of-the-art accuracy. The present paper applies this system mostly as is, to the data set created from bar exams. In the experiments, the accuracy on this data set is almost comparable to the random baseline, which indicates the difficulty of this task compared to the RITE task. The analysis of the result reveals that the difficulty lies in the semantics and linguistic expressions peculiar in the legal domain, such as pervasive usages of negations, modality expressions, and domain-specific terms.

1 Introduction

Recognizing textual entailment (RTE) is a task in natural language processing (NLP) that aims at identifying semantic relations between two texts. Given a text pair, t and h, the goal is to judge whether t *entails* h. The *entailment* relation is defined as "a human reading t would infer that h is most likely true" [12]. It should be noted, however, that this task aims to replicate human judgment of semantic relations among texts, rather than to precisely compute logical entailment/inference relations.

Recognition of semantic inclusion/equivalence is an essential issue in semantic processing in NLP, and RTE has been proposed as a generic solution to the problem without referring to any concrete formalisms of semantics. In fact, the task has been designed so that it supports a variety of NLP tasks/applications, such as text summarization, information extraction, and question answering. For example, in question answering, RTE systems are supposed to be used in the validation phase; a RTE system is expected to validate the correctness of answer candidates, by judging an entailment relation between an evidence text and a query text filled with a candidate answer.

Several shared tasks have been organized to date, mainly in English known as RTE shared tasks, [12, 2, 15, 14, 5, 3, 4], while recently, in Japanese and Chinese known as NTCIR RITE [22, 29]. The top systems use an integration of a variety of NLP techniques, such as logic-based inference, similarity of syntactic/semantic structures, large-scale thesauri and paraphrase dictionaries automatically obtained from the Web, etc. However, the achievements so far are far

from satisfactory. The accuracy of the top-level systems is only slightly above the random baseline, which is still far from human performance.

In this paper, we apply *the Tifmo system*, which was previously applied to the Japanese subtasks of NTCIR RITE2 [29], to the Japanese data set of the bar exam competition at JURISIN 2014. The Tifmo system is an integration of a recently proposed logic framework called dependency-based compositional semantics (DCS) [18, 26], and a method for measuring similarity of words/phrases called word embeddings [20]. This RTE system integrates logical and statistical directions of semantic processing nicely in the framework called *on-the-fly knowledge injection*, which is described in Section 3. This system achieved the top performance in the NTCIR RITE2 exam BC (binary classification) subtask in Japanese, and state-of-the-art accuracy on the RTE5 data set in English.

The main contribution of this work is to report a first result of applying the Tifmo system on the legal domain, and to explore possibilities for future improvements. In this work we applied the system mostly as is; we do not apply any improvement or tuning of the system for this particular domain/task. In other words, this paper does not propose any novel technologies for RTE, but is aimed to empirically evaluate the system in a domain different from typical RTE datasets. Experiments on the development set demonstrated that the system achieves accuracy comparative to the majority label baseline, which tells little contribution of the current system to the bar exam task. The further investigation of the result shows that the current system lacks the proper processing of semantics and linguistic expressions that appear frequently in legal documents, such as negations, modality expressions, and domain-specific terms. Additionally, the result shows that the system achieves higher precision than recall, which suggests the effectiveness in applications where high precision is demanded.

2 Related Work

Recognizing textual entailment (RTE) has been studied extensively, although the main focus was mostly in English. A series of shared tasks and data sets have been organized [12, 2, 15, 14, 5, 3, 4], and they contributed significantly to the research on RTE. State-of-the-art systems integrate a variety of NLP techniques and resources, such as methods for computing similarity of surface strings or linguistic structures [30, 28] and transformation of dependency trees [23, 24]. Logic-based systems, most of which are based on first-order logic, have also been applied in the early age of RTE research [21, 10, 25]. However, logic-based systems were fragile on real-world data where necessary knowledge is often missing in currently available resources. Therefore, approaches for integrating logic-based methods with shallow statistical methods have been pursued [11, 9]. For more details, refer to [1, 13].

RTE research in Japanese has been limited, while the shared tasks NTCIR RITE1/RITE2 boosted RTE research on Japanese. Systems proposed in RITE1 and RITE2 mostly follow the current state-of-the-art technologies in the English systems. For details, refer to the overview papers of RITE1 and RITE2 [22, 29].

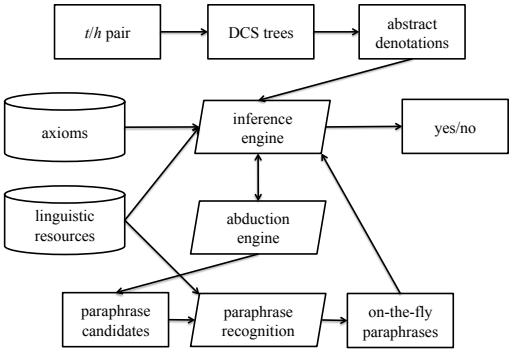


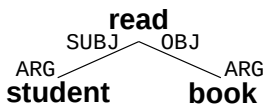
Fig. 1. Tifmo system

We apply the Tifmo system [26], which is a RTE system based on a recently proposed logic framework called dependency-based compositional semantics (DCS) [18]. DCS was proposed as a query language for databases (in this sense it is similar to SQL or SPARQL), while its structure resembles dependency trees of natural language sentences. This allows for a relatively easy mapping from natural language texts into DCS trees. [18] applied this framework for the learning of text-to-DCS mappings from question-answer pairs in a distant supervision manner, where DCS representations are not explicitly given as training data while modeled as latent structures. This work was further extended to question answering on Freebase [6], which is becoming an active research area in NLP recently [7].

[26] extended the DCS framework so that it can be applied to RTE without any databases by introducing the notion of *abstract denotations*, which we describe next.

3 System Overview

This section introduces the Tifmo system, which was applied to NTCIR RITE1 and RITE2 [22, 29]. The system achieved the highest score on the RITE2 exam BC (binary classification) subtask, which is a task very similar to the bar exam competition, in the sense that the data set was taken from university entrance exams and high school textbooks. We employed a system mostly identical to that used in RITE2 [27], while we added a method for scoring of on-the-fly paraphrase knowledge using word embeddings (see below for details), which was successfully applied in the Tifmo system for English [26]. In this work we stick to use the logic-based system only, while the results submitted to NTCIR RITE2 used combination of the logic-based system with a robust statistical method using shallow features like overlap ratios of named entities. This is mainly because the purpose of the present work is to evaluate the potential and the future improvements of the logic-based approach.



$$F_1 = \mathbf{read} \cap (\mathbf{student}_{\texttt{SUBJ}} \times \mathbf{book}_{\texttt{OBJ}}),$$

Fig. 2. An example DCS tree and its abstract denotation (cited from [26])

Figure 1 shows an overview of the system. The system accepts a pair of natural language texts, t and h, as input, and parse them into *DCS trees*. DCS trees are converted into *abstract denotations* and statements on them, using the algorithm proposed in [26]. The inference engine tries to prove h from t by processing abstract denotations and statements, while the abduction engine finds missing paraphrase candidates on-the-fly, and inject the validated paraphrases to proceed the inference.

In the following, we introduce DCS trees and abstract denotations briefly, and describe how the inference proceeds. Finally, NLP tools and resources used in the present system are presented.

Figure 2 shows an example DCS tree and its abstract denotation. They show the semantics of the sentence "*Students read books*". DCS trees are tree structures where nodes denote words/phrases while edges between words represent syntactic/semantic relations between words/phrases. Both sides of edges are labeled with semantic roles such as `SUBJ` (subject) and `OBJ` (object). The semantic role `ARG` is introduced to denote the semantic role of nouns. For example, the edge "$\mathbf{read}_{\texttt{OBJ}-\texttt{ARG}}\mathbf{book}$" means that the object of the "read" event is the things denoted by the noun "book". As you see in this example, DCS trees resemble dependency trees we often use in NLP. Since high-accuracy dependency parsers are already available in languages like English and Japanese [19, 17], we can easily obtain DCS trees from natural language texts by applying an off-the-shelf dependency parser, and converting its output into DCS trees.

DCS trees were originally proposed for representing database queries. For example, the DCS tree in Figure 2 describes the query to take database entries in the "read" table, whose "SUBJ" field has an entry of the "student" table and the "OBJ" field has an entry of the "book" table; that is, it obtains *denotations* of the DCS tree by computing the intersection of the three tables. The original DCS framework [18] presupposed a database is given, and denotations of the query is directly computed over the given database. This is reasonable for database question answering, while it is a significant drawback when we intend to apply this framework to RTE of unrestricted natural texts, because we cannot prepare a database of unrestricted concepts in the world beforehand.

The heart of the problem is that we cannot directly compute denotations of natural language texts. [26] solved this problem by introducing *abstract denotations*, which represent the result of computing denotations as algebraic forms of relational algebra, independently of any concrete databases. Figure 2 shows

the abstract denotation F_1 of the example DCS tree. This represents that the denotation is obtained as taking the intersection of the set "read" with the Cartesian product of the sets "student" and "book". It should be noted that the abstract denotation represents the result of computing denotations of DCS trees as mentioned above. Abstract denotations are obtained from DCS trees by the algorithm proposed in [26]. The algorithm traverses a DCS trees and map it into set representations like intersection and Cartesian product. See [26] for the definition of abstract denotations and the detail of the translation algorithm.

DCS trees and abstract denotations not only describe simple set intersection/product, but also can represent quantifiers and negation such as "all" and "not", as well as generalized quantifiers like "at most X" etc. However, we omit the detail here.

The core part of inference consists of two components. One is the *inference engine*, which computes all abstract denotations and their *statements* from t. Statements are relations over abstract denotations, including *non-emptiness* $X \neq \emptyset$, *subsumption* $X \subset Y$, and *disjointness* $X||Y$. Intuitively, they correspond to satisfiability, entailment, and contradiction of natural language expressions. Entailment relations of texts can be computed as relations over statements; if statements computed from t cover all of the statements of h, h is proven from t. For example, we can infer:

$$\mathbf{read} \cap (\mathbf{student}_{\texttt{SUBJ}} \times \mathbf{something}_{\texttt{OBJ}}) \neq \emptyset,$$

from $F_1 \neq \emptyset$ and a knowledge $\mathbf{book} \subset \mathbf{something}$, which means "*Students read books*" entails "*Students read something*".

However, this sort of strict logical inference has not been successful in RTE of real-world sentences, because real sentence pairs involve a variety of linguistic phenomena and paraphrases, many of which are missing in available resources and/or context dependent. It is infeasible to prepare all the necessary knowledge to prove h from t beforehand. Statistical approaches like machine learning classifiers are robust in this respect because they output scores as a "degree of proof" even when full knowledge is unavailable, which is missing in traditional logic-based systems.

The other core part of the Tifmo system, the *abduction engine*, solves this problem by producing paraphrase candidates *on-the-fly* during inference. This component identifies parts of h which could not be proven in the inference, and find its possibly corresponding fragments in t. An alignment of fragments of t and h constitute a paraphrase candidate. If a candidate is a true paraphrase, this on-the-fly knowledge is injected into the inference engine to proceed the inference.

We can apply NLP techniques of paraphrase recognition for judging whether or not a paraphrase candidate is a true paraphrase. This is possible because abstract denotations and DCS trees have clear correspondences to natural language expressions. In the current system, we use a measure of distributional similarity computed by word embeddings [20]. In other words, the abduction engine reduces the entailment relation of entire sentences into smaller pieces of para-

Table 1. Results on the training set

	H18	H19	H20	H21
accuracy	0.47 (17/36)	0.45 (19/42)	0.49 (22/45)	0.61 (34/56)
precision	0.33 (2/6)	0.63 (5/8)	0.62 (8/13)	0.78 (14/18)
recall	0.12 (2/17)	0.20 (5/25)	0.31 (8/26)	0.44 (14/32)
F-score	0.17	0.30	0.41	0.56
baseline accuracy	0.53 (19/36)	0.60 (25/42)	0.58 (26/45)	0.57 (32/56)

phrase relations where a variety of statistical NLP techniques can be exploited, by using logical relations computed by the inference engine.

The external tools and resources we use in this system for Japanese are listed below.

Dependency parsing and named entity recognition We apply Cabocha [17], which is the most widely used parser for Japanese. Accuracy of dependency parsing is reported as around 90% on news texts, while the accuracy on legal texts is unknown. Cabocha performs named entity recognition as well, which is also used in the Tifmo system. Nihongo Goi Taikei[1] and Wikipedia are additionally used for named entity recognition.

Coreference resolution We apply Syncha [16], which uses ILP-based inference for coreference and anaphora resolution for Japanese.

Linguistic knowledge Synonyms, hypernyms, hyponyms, and antonyms are extracted from Nihongo Goi Taikei, Wikipedia, Japanese WordNet [8], and Kojien[2]. For any pair of words in input texts, these linguistic knowledge is input to the inference engine beforehand.

Paraphrase recognition Paraphrase candidates generated by the abduction engine are evaluated with two similarity scores. One is the similarity computed using Bunrui Goi Hyo[3]. The other is the distributional similarity computed from word embeddings [20]. We applied `word2vec`[4] to Japanese Wikipedia to obtain word embeddings of 300 dimensions.

Note that our system is unsupervised; we don't have any component which requires supervised training on RTE data. The only parameter is the threshold of the similarity scores for accepting on-the-fly knowledge, which is fixed to 0.6 in the experiments below.

4 Experiments

Table 1 shows accuracy, precision, recall, and F-score for each section of the training set. Baseline results are shown additionally, where the majority label is

[1] http://www.kecl.ntt.co.jp/icl/lirg/resources/GoiTaikei/
[2] http://www.iwanami.co.jp/kojien/
[3] http://www.ninjal.ac.jp/archives/goihyo/
[4] https://code.google.com/p/word2vec/

given to all instances ("N" for H18 and "Y" for H19, H20, and H21). The result shows that the system achieves no significant improvement in terms of accuracy over the baseline, indicating that the current system has little contribution to the task of the bar exam competition. This result also implies the difficulty of the bar exam competition in comparison with NTCIR RITE2, while the results are not directly comparable.

Additionally, the result reveals that the system attained higher precision than recall (except for H18), in particular for H21. In other words, the system does not simply output labels randomly. The system predicts entailment pairs rather correctly, while the low recall degrades the overall score. This is probably because the current system produces paraphrase candidates under logical constraints (details omitted in this paper; see [26] for details), and this resulted in injecting accurate paraphrases even with the rough similarity scoring in the current system. This observation suggests the effectiveness of this logic-based system in applications where high precision is demanded.

Further investigating the result we identified several issues that have to be addressed to improve accuracy.

Domain-specific terms and expressions Legal documents include several domain-specific expressions like technical terms. Semantic relations among such terms (e.g. "shiyosha (employer)" and "hiyosha (employee)") are not fully covered by the resources we currently have, and this confuses the inference and abduction engines. In particular, antonym relations are crucial, because they cannot be processed in the on-the-fly knowledge injection. In addition, the documents include domain-specific coreference expressions such as "zenko (previous item)" frequently, which are not processed in the current coreference resolution system. Such coreferences are often crucial in recognizing entailment relations in this domain.

Negation and modality The data set includes negations and modality expressions, such as "kagiri-de-nai (not restricted to)", "deki-nai (cannot)", and "nakereba-nara-nai (must)" very frequently, while such expressions rarely appear in previous data sets like NTCIR RITE. The Tifmo system currently recognizes simple negation "nai (not)", while not the other modality expressions. Apparently semantics of modality expressions are often crucial in correctly recognizing entailment relations in this domain, and we suspect that this hurt the accuracy significantly.

Relating instances to abstract concepts Several pairs in the data set consists of an abstract statement in t (e.g. about the contract of trading) and its concrete instance in the real life as h (e.g. selling jewelries). This type of entailment was not popular in previous data sets, and cannot be recognized by the current RTE systems. This type of pairs require common-sense knowledge, and we don't have a promising solution at the moment.

Further more, inferences across multiple sentences/clauses are often required, and this not only makes the inference difficult but also increases the possibility of having errors of NLP tools including parsing and anaphora/coreference resolution.

5 Conclusion

This paper reported the first result on applying the Tifmo system, which achieved near state-of-the-art performance on NTCIR RITE2, to the bar exam competition. The off-the-shelf system was applied to the data sets provided, resulting in accuracy mostly comparable to the majority label baseline. While this result implies the current RTE system has little contribution to the bar exam task, the investigation of the result suggested several possibilities for improvements, like precise processing of negation/modality and domain-specific terms/expressions. The result also showed that the logic-based system attains higher precision than recall, suggesting possibility of usability in precision-oriented applications.

References

1. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. J. Artif. Int. Res. 38(1) (2010)
2. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B.: The second PASCAL recognising textual entailment challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (2006)
3. Bentivogli, L., Clark, P., Dagan, I., Giampiccolo, D.: The sixth PASCAL recognizing textual entailment challenge. In: Proceedings of TAC 2010 (2010)
4. Bentivogli, L., Clark, P., Dagan, I., Giampiccolo, D.: The seventh PASCAL recognizing textual entailment challenge. In: Proceedings of TAC 2011 (2011)
5. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The fifth PASCAL recognizing textual entailment challenge. In: Proceedings of TAC 2009 (2009)
6. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on Freebase from question-answer pairs. In: Proceedings of EMNLP 2013 (2013)
7. Berant, J., Liang, P.: Semantic parsing via paraphrasing. In: Proceedings of ACL 2014 (2014)
8. Bond, F., Baldwin, T., Fothergill, R., Uchimoto, K.: Japanese SemCor: A sense-tagged corpus of Japanese. In: Proceedings of GWC-2012 (2012)
9. Bos, J.: Is there a place for logic in recognizing textual entailment? Linguistic Issues in Language Technology 9 (2013)
10. Bos, J., Markert, K.: When logical inference helps determining textual entailment (and when it doesnt). In: Proceedings of the 2nd PASCAL RTE Challenge Workshop (2006)
11. Clark, P., Harrison, P.: Recognizing textual entailment with logical inference. In: Proceedings of 2008 Text Analysis Conference (TAC'08) (2008)
12. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: et al., J.Q.C. (ed.) Machine Learning Challenges, Lecture Notes in Computer Science, vol. 3944. Springer-Verlag (2006)

13. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.M.: Recognizing Textual Entailment: Models and Applications. Morgan & Claypool Publishers (2013)
14. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Dolan, B.: The fourth PASCAL recognizing textual entailment challenge. In: Proceedings of TAC 2008 (2009)
15. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third PASCAL recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (2007)
16. Iida, R., Poesio, M.: A cross-lingual ILP solution to zero anaphora resolution. In: Proceedings of ACL-HLT 2011 (2011)
17. Kudo, T., Matsumoto, Y.: Japanese dependency analyisis using cascaded chunking. In: Proceedings of CoNLL 2002 (2002)
18. Liang, P., Jordan, M., Klein, D.: Learning dependency-based compositional semantics. In: Proceedings of ACL 2011 (2011)
19. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC 2006 (2006)
20. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of NAACL 2013 (2013)
21. Raina, R., Ng, A.Y., Manning, C.D.: Robust textual inference via learning and abductive reasoning. In: Proceedings of AAAI 2005 (2005)
22. Shima, H., Kanayama, H., Lee, C.W., Lin, C.J., Mitamura, T., Miyao, Y., Shi, S., Takeda, K.: Overview of NTCIR-9 RITE: Recognizing inference in text. In: Proceedings of NTCIR-9 (2011)
23. Stern, A., Dagan, I.: A confidence model for syntactically-motivated entailment proofs. In: Proceedings of RANLP 2011 (2011)
24. Stern, A., Stern, R., Dagan, I., Felner, A.: Efficient search for transformation-based inference. In: Proceedings of ACL 2012 (2012)
25. Tatu, M., Moldovan, D.: A logic-based semantic approach to recognizing textual entailment. In: Proceedings of the COLING/ACL 2006 (2006)
26. Tian, R., Miyao, Y., Matsuzaki, T.: Logical inference on dependency-based compositional semantics. In: Proceedings of ACL 2014 (2014)
27. Tian, R., Miyao, Y., Matsuzaki, T., Komatsu, H.: BnO at NTCIR-10 RITE: A strong shallow approach and an inference-based textual entailment recognition system. In: Proceedings of NTCIR-10 (2013)
28. Wang, M., Manning, C.: Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In: Proceedings of Coling 2010 (2010)
29. Watanabe, Y., Miyao, Y., Mizuno, J., Shibata, T., Kanayama, H., Lee, C.W., Lin, C.J., Shi, S., Mitamura, T., Kando, N., Shima, H., Takeda, K.: Overview of the Recognizing Inference in Text (RITE-2) at the NTCIR-10 Workshop. In: Proceedings of NTCIR-10 (2013)
30. Zanzotto, F.m., Pennacchiotti, M., Moschitti, A.: A machine learning approach to textual entailment recognition. Nat. Lang. Eng. 15(4) (2009)

JAVN system for Legal Extraction and Entailment Tasks

Tu Thanh Vu, Binh Thanh Kieu, Quan Hung Tran, Cuong Xuan Chu, Son Bao Pham[1] and Minh Le Nguyen[2]

[1]VNU-UET, Hanoi, Vietnam
[2]JAIST, Japan
{tuvt@vnu.edu.vn,
binhkt.vnu@gmail.com,
quanthdhcn@gmail.com,
cuongcx@vnu.edu.vn,
sonpb@vnu.edu.vn,
nguyenml@jaist.ac.jp}

Abstract. This paper describes the textual entailment system (JAVN) for the competition on Legal Information Extraction/Entailment task (COLIEE-14). Our system is based on machine learning method; particularly, Support Vector Machine that uses features from lexical similarity, lexical distance with radial basis function kernel and string kernel. We also consider the use of alignment methods (machine translation evaluation) as well as text rewriting kernel for textual entailment recognition. Experimental results show that the alignment methods achieved a promising result.

Team Name

JAVN: The team includes members of VNU-UET and JAIST with the lead of Associate Professor Minh Le Nguyen and Associate Professor Pham Bao Son.

Keywords: entailment, system validation, binary class, question answering, legal text processing

1 Introduction

Textual entailment recognition is the task of deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text [3]. This task captures generically a broad range of inferences that are relevant for multiple applications such as Question Answering (QA), Summarization, Paraphrasing, Information Extraction.
This paper describes our textual entailment for COLIEE-14 task. To recognize relation between two sentences, we developed a system based on machine learning method for textual entailment in the legal domain. We did an experiment to

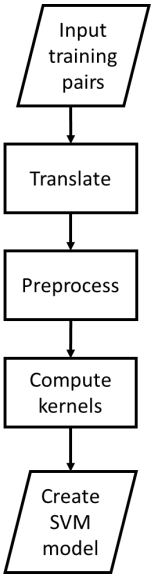


Fig. 1. Training model workflow

explore how the system is tested on the legal data set.
Section 2 describes some methods to build systems. Section 3 describes the submitted systems. In section 4, we show the experiment of these systems.

2 Methods

We employed three systems corresponding three main methods: SVM using string kernel, SVM with feature extraction, and alignment method. In order to help readers easily understand the proposed system, we briefly describe them as follows.

2.1 String kernel

Modeling the sentence re-writing is an important task in natural language processing. There are a number of re-writing representations: lexicon based [8], syntactic trees based [5], re-writing rules based [1]. In the research of [1], the authors proposes a new class of kernel function called string rewriting kernel (SRK) that "defines the similarity between two re-writings (pairs) of strings as the inner product between them in the feature space induced by all the rewriting rules". Experiments of SRK method showed strong performance in textual recognition and paraphrase identification in English. In this paper, we employ the machine learning method with SRK kernel and test their effectiveness on RTE task in Japanese.

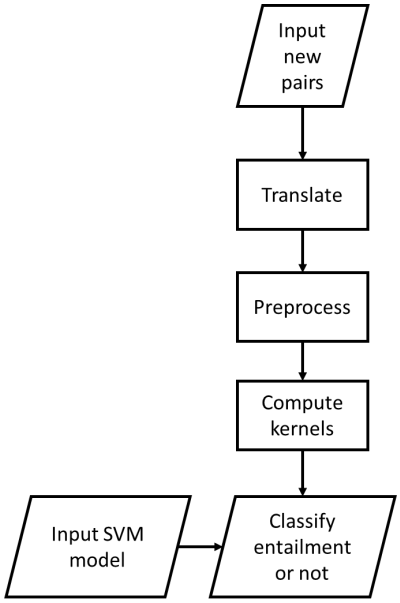


Fig. 2. Training model workflow

Computing the SRK kernel directly is expensive because of large number of re-writing rules. In this paper, we use the K-gram Bijective String Re-writing Kernel (kb-SRK) [1] to reduce the computational complexity. We experimented the kb-SRK in both original Japanese text and translated text (English). The translation step were done using Google API. As our experiment using last year's data showed that the kb-SRK method performs better on English text, we decide to translate all the sentences into English before computing the kernel. As the overall time complexity of computing kb-SRK is highly dependent on the string length, we remove stop words from the translated English text to reduce the length of the text.
Using the precomputed model, we classify a pair of text as "entailment" or "not entailment".

The accuracy of our system is 58% when we conduct a 10 folds cross validation test. It is clearly to show that the performance of the system is not high. A possible reason is that the legal text are very long and we currently do not have an appropriate method for dealing with this matter. In addition, in our implementation of the kb-SRK, we do not take into account the semantic relations of words. A pair of very similar words and a pair of totally different words are treated similarly. Incorporate semantic relation into SRK should be considered in the future. Another problem with the String Re-writing kernel is time complexity, even the kb-SRK is very computational expensive (each kernel takes 1-3 seconds to compute), it makes building the SVM model from training data very time-consuming, especially when applying for long text span.

2.2 Feature Extraction

We extract some features to analysis the relation between two sentences and using these features to run SVM method.

Lexical features We extracted four lexical features. These features present the similarity of two sentences. The higher value of similarity has higher rate for entailment between two sentences.

- Cosine similarity of Words: Let CW_1 be the set of content words in t_1 and CW_2 be the set of content words in t_2. The value of this measurement is defined as $|CW_1 \bigcap CW_2| = |CW_1||CW_2|$, with $|CW_1|, |CW_2|$, and $|CW_1 \bigcap CW_2|$ are the number of content words in the sets. We used Mecab[1] to recognize content words for each sentence.
- Cosine similarity of Characters: In addition to word based one, we also calculate character-based cosine similarity.
- Jaccard coefficient: Let $|CW_1 \bigcap CW_2|$ - the number of content words that occur both in t_1 and t_2, $|CW_1 \bigcup CW_2|$ - the number of content words that occur in t_1 or t_2. This measurement is defined as $|CW_1 \bigcap CW_2|/|CW_1 \bigcup CW_2|$
- Longest common subsequence: This measurement is defined as a common subsequence of string t_1 and t_2 of maximum length. We used the length of the longest common subsequence that is normalized by the length of t_2.

Dependency features Due to the lack of tools for Japanese dependency, the sentence is translated into English. Its dependency features are extracted by using Treex[2]. The list of features taken into consideration are as follows:

- Word Overlapping: The number of tokens shared between two sentences.
- Dependency edge overlapping: The number of edges shared between two dependency trees.
- Dependency relation to HEAD (deprel): The relation of words with regards to the head of the sentence. This feature measures how many words in two sentences have the same role (relation).
- Dependency edge and *deprel* combined
- Co-reference: If two sentences is parsed together, this feature measures how many co-reference nodes in the dependency tree.

Other features Moreover, we use other features for our implementation. Firstly, we use the quantity of negative words in each sentence. Indeed, the inconsistency of these indexes can be a strong proof for the disagreement of entailment between them. Secondly, our method is based on word correspondence between sentences. To specify, we find the rate of words in each sentence containing in another sentence. This feature is processed on surface level. Finally, after

[1] http://mecab.googlecode.com/svn/trunk/mecab/
[2] http://ufal.mff.cuni.cz/treex

translating Japanese into English, we compute the length values of all word list excluding stop words. If these values of two sentences are far different, we may predict that two sentences disagree. However, the experiment for these features had no high accuracy so these features were not selected to submit formal runs.

2.3 Alignment Value Method

JURISIN 2014 Dataset Because of our ignorance of Japanese, we just investigate the provided English translation of the data sets. We have a paired dataset containing triples each of which consists of a query t_2, relevant articles t_1 and a correct answer "Y" or "N". We note that, the relevant articles field t_1 usually contain several Japanese civil code articles, each article starts with a formatted header (for example, "(Special Agreement Disclaiming Warranty)Article 572") and may contain several clauses, cited as "(1)", "(2)", "(3)", etc., each clause may contain sentences or sub-clauses, cited as sub-clauses "(i)", "(ii)", "(iii)", etc. The query t_2 may also contain more than one sentence. We processed the dataset as follows:

- Plain Text Extraction: We extracted plain texts as query sentences and corresponding civil codes by removing specific XML tags.
- Pre-processing: This step includes sentence boundary detection and tokenization with the Stanford Parser package [9]. Our system treats each relevant articles field t_1 as a set of sentences and operates on individual sentences. We also omitted article headers as we observe that they do not make much sense and removed stopwords by using a pre-compiled stopwords list. The complexity of long sentences can therefore be reduced.
- Sentence Alignment: As mention above, we have a set of sentences $S = (S_1, S_2, ..., S_n)$ from each articles field t_1. Each query field t_2 consists of a legal bar exam question Q considered as a hypothesis. Our task involves the identification of an entailment relationship such that $Entails(S_1, S_2, ..., S_n, Q)$ or $Entails(S_1, S_2, ..., S_n, notQ)$. We adopt the following two-step classification classification strategy given a set of texts $(S_1, S_2, ..., S_n)$ and a hypothesis Q.

Strategy

- Step 1: Considering each pair $< S_i, Q >$ $(i = 1, 2, ..., n)$ and identifying whether S_i and Q are related to each other or not based on the similarity in appearance between S_i and Q. When two sentences are related semantically, they tend to be similar in appearance. By applying some heuristic rules, we determine if there is a "rough" entailment relationship between S_i and Q which mean S_i roughly entails Q.
- Step 2: If there exists at least one aligned pair $< S_i, Q >$ $(i = 1, 2, ..., n)$ such that S_i entails Q, we conclude that there is an entailment relationship between the relevant articles S and Q. There may be case where there is no pair $< S_i, Q >$ such that S_i entails Q, however, there still exists an entailment

relationship that S entails Q. This can be explained as S might contain some sentences $S_{i_1}, S_{i_2}, ..., S_{i_k}$ of which each sentence partly entails a certain part of Q and the inference that S entails Q can be obtained from the combination of clues $< S_{i_j}, Q >$ $(j = 1, 2, ..., k)$. However, in our experiments, we ignored this case since it requires complex computations.

Paraphrase Identification The question posed here is how to identify whether S_i and Q are related to each other or not based on the similarity in appearance between them. Usually, when S_i entails Q, some parts of Q are copied or paraphrased from S_i, and several words or phrases in S_i are repeated or substituted by their synonyms or paraphrases in Q. Hence, we align corresponding words/phrases between the text S_i and the hypothesis Q. We treat the task of paraphrase identification as a monolingual machine translation task, where the source and target languages are the same and where each part of the output should be similar in meaning to a certain part of the input, and then using some machine translation techniques.
We examine a paraphrase identification approach that relies on some machine translation evaluation metrics and then build up an automatic word/phrase alignment module. BLEU [10] and TERp [11] are commonly used in machine translation to automatically evaluate the quality of output. BLEU counts n-gram matches with the reference to score the target output. TERp measures the number of edit operations (insertions, deletions, substitutions and shifts) needed to transform the machine translation output into the reference translation. METEOR [12] uses a combination of both precision and recall unlike BLEU which focuses on precision. Alignments are based on exact, stem, synonym (via WordNet), and paraphrase (via a lookup table) matches between words and phrases. We obtain a list of word/phrase alignments from each aligned pair $< S_i, Q >$. Based on the number of aligned word/phrase pair as well as the length of the text S_i and the hypothesis Q, we determine their entailment relationship. We note that when the length of Q is much longer than the total length of relevant articles S, it is more likely that Q contains some information that cannot be entailed from S. The pseudo-code of the algorithm is described below, where S_i and Q are the text and the hypothesis after removing stopwords, respectively. *numOfMatches* is the number of aligned word/phrase pairs.

$threshold1$ is the maximum value of difference between the length of query Q and the length of relevant articles S for all pairs $< S, Q >$ in the training dataset where S entails Q. Our experimental results indicate that the best values for $threshold2, threshold3, threshold4$ are $threshold2 = 10, threshold3 = 5, threshold4 = 10.5$.

3 System Development

We tested the proposed system on the COLEE-14 data set using 10 folds cross validation test by using the following features with Support Vector Machines (SVM). For SVMs, we used the LibSVM [2].

Algorithm 1 Paraphrase Identification

procedure EI
  $delta \leftarrow Q.length - S.length$
  if $delta \geq threshold1$ **then**
    return False
  $numOfMatches \leftarrow numOfMatches(S_i, Q)$
  if $(minLength \leq threshold2)$ and $(numOfMatches \geq minLength/2 - threshold3)$ **then**
    return True
  else if $(numOfMatches \geq minLength/2 + threshold4)$ **then**
    return True

  return False

The evaluation method is based on precision, recall, and F-measure.

- System 1: SVM with string kernel
- System 2: SVM with feature extraction
- System 3: Using alignment value to decide a pair sentences which entails or does not entail

We tested SVM with string rewriting kernel method [1], however it takes very long time to train the system. We will report the result of cross validation test later. System 2 is applied feature extraction with SVM. We obtained a precision of 58%. For System 3, we used a paraphrase identification method (alignment method), the result is 49%. After using pre-processing method by removing redundant sentences (i.e "(Enforcement of Performance)Article 414)". Assuming we want to check the textual entailment between t_1 and t_2, in the case of t_1 can be split into some simple sentences, we divide it into some simple sentences then conduct textual entailment recognition with each simple sentence and t_2. The results of our system is significantly improved. The difficult of the proposed method when dealing with legal data is that those sentences in the legal data is very long and complex. We may need some pre-processing or legal sentences analyzing methods to simplify sentences. In other words, the number of sentences in the training data set is not enough to train our system. This factor is also a reason the machine learning methods are not effective in COLEE-14. We hope un-supervised learning methods can deal with this.

4 Results

4.1 Results for all systems

The result of the proposed systems are shown in the Table 1.

The training data has some very similar sentence pairs, with some pairs are labeled as entailed, while others are not. The translated versions of these sentences are sometimes indistinguishable. The translation step altered the sentences and make textual entailment very difficult, even for human readers.

Table 1. The result of our runs for COLIEE-14 task

System	Accuracy
SVM with string kernel	not yet
SVM with feature extraction	58%
ALignment	47.49%
Alignment with pre-processing	73.74 %

Table 2. The result of alignment system

System	Macro-F1	Acc.	Y-F1	Y-Prec.	Y-Rec.	N-F1	N-Prec.	N-Rec.
BP	45.64%	47.49%	35.62%	56.52%	26.00%	55.66%	44.36%	74.68%
AP	73.51%	73.74%	76.14%	77.32%	75.00%	70.87%	69.51%	72.15%

4.2 Results for Alignment System

Table 2 shows our experimental results before (BP) and after (AP) pre-processing and operating on individual sentences of the training dataset which contains 179 pairs of query and relevant articles, using the same values of thresholds.

$Accurancy(Acc) = Number(correct)/Number(target)$

$$Precision = \frac{\#\text{correct}}{\#\text{predicted}}$$
$$Recall = \frac{\#\text{correct}}{\#\text{target}}$$
$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$macroF1 = 1/|C| \sum (2 * Prec * Recall)/(Prec + Recall)$ - C the set of classes

4.3 Failure analysis

To illustrate the performance of the proposed system, we provide here the list of cases where our system could not generate correct outputs.

Case 1 When relevant articles t_1 and a query t_2 are similar in appearance and there exists an entailment relationship that t_1 entails t_2.

Example 1:

<pair id="H21-3-3" label="Y">

$< t_1 >$(Collecting Fruits of Testamentary Gift)Article 992 A testamentary donee may collect the fruits of a testamentary gift from the time that they are able to make a claim for the performance of that gift; provided that if the testator has indicated a particular intent in his/her will, that intent shall be complied with.$< /t_1 >$

$< t_2 >$A testamentary donee may collect the fruits of a testamentary gift from the time that he/she is to able to make a claim for the performance of the gift, as

long as the testator has not indicated a particular intent in his/her will.< /t_2 >
Example 2:
<pair id="H19-13-A" label="Y">
< t_1 >(Indivisibility of Rights of Retention)Article 296 A holder of a right of retention may exercise his/her rights against the whole of the Thing retained until his/her claim is satisfied in its entirety. < /t_1 >
< t_2 >A holder of a right of retention may retain that thing until his/her claim is satisfied in its entirety.< /t_2 >

Case 2 When relevant articles t_1 and a query t_2 are similar in appearance but there does not exist an entailment relationship that t_1 entails t_2. This case is difficult to deal with the proposed system. More deeply semantic parsing is required to improve the textual entailment performance,
Example 3:
<pair id="H21-8-A" label="N">
< t_1 >(Possession by Agents)Article 181 Possessory rights may be acquired by an agent. < /t_1 >
< t_2 >An agent may even acquire a possessory right, but since the effect of possession by the agent belongs to the principle, the agent him/herself may not acquire an individual possessory right for the thing possessed. < /t_2 >
Example 4:
<pair id="H19-26-O" label="N">
< t_1 >(Requirement for Perfection of Termination of Mandate)Article 655 The grounds of termination of mandate may not be asserted against the other party unless the other party was notified of or knew of the same. < /t_1 >
< t_2 >Regardless of the other party's knowledge, the grounds of termination of mandate may not be asserted against the other party unless the other party was notified of the same. < /t_2 >

Case 3 When relevant articles t_1 and a query t_2 are not similar in appearance but there still exists an entailment relationship that t_1 entails t_2. This case is also difficult to deal with the proposed system. More deeply semantic parsing is required to improve the textual entailment performance,
Example 5:
<pair id="H18-2-2" label="Y">
< t_1 >(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. < /t_1 >
< t_2 >In cases where an individual rescues another person from getting hit by a car by pushing that person out of the way, causing the person's luxury kimono to get dirty, the rescuer does not have to compensate damages for the kimono. < /t_2 >
Example 6:

<pair id="H19-15-I" label="Y">
< t_1 > (Loans for Consumption)Article 587 A loan for consumption shall become effective when one of the parties receives money or other things from the other party by promising that he/she will return by means of things that are the same in kind, quality and quantity. < /t_1 >
< t_2 >In cases where the registration for the creation of a mortgage has been made for monetary loan claims, but the principal was not delivered in the end, the mortgagor may demand cancellation of such registration due to the non-existence of secured claim. < /t_2 >

Case 4 When relevant articles t_1 and a query t_2 are not similar in appearance and there does not exists an entailment relationship that t_1 entails t_2.
Example 7:
<pair id="H19-8-A" label="N">
< t_1 >(Requirements of Perfection of Changes in Real Rights concerning Immovable properties)Article 177 Acquisitions of, losses of and changes in real rights concerning immovable properties may not be asserted against third parties, unless the same are registered pursuant to the applicable provisions of the Real Estate Registration Act (Law No. 123 of 2004) and other laws regarding registration. < /t_1 >
< t_2 >X, who purchased a real estate P from decedent A before the commencement of inheritance, may assert his ownership of P against Y without registration, if Y, who is the obligee of A's sole heir B, seized P in subrogation to B after B's inheritance registration of P.< /t_2 >
Example 8:
<pair id="H18-29-4" label="N">
< t_1 > (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 oblige 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 majesty with respect to the damages referred to in paragraph 1. < /t_1 >
< t_2 >In a demand for compensation based on delayed performance of loan claim, if person Y gives proof that the delayed performance is not based on the reasons attributable to him or herself, person Y shall be relieved of the liability.< /t_2 >

5 Conclusions

This paper has described the textual entailment system of the competition on Legal Information Extraction/Entailment task (COLIEE-14). In this paper, we have experimented with a number of machine learning methods for the textual recognition task. The method which uses machine learning models is better than

using alignment method. However, when we conduct a very simple method for analyzing the structure of legal text, the performance of alignment method improved very much. This leads to an interesting work in the future that we can exploit analyzing legal sentences for textual entailment recognition. We also plan to add a number of rules for the task, instead of relying entirely on the machine learning methods.

6 Additional Authors

Additional authors: Tam Duc Hoang (VNU-UET, Vietnam. email: tamhd1990@gmail.com) and Dang Hai Tran (VNU-UET, Vietnam. email: dangth010589@gmail.com).

References

1. Bu, Fan and Li, Hang and Zhu, Xiaoyan, "String Re-writing Kernel," Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Vol. 1, 2012.
2. C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intelligent Systems and Technology, Vol. 2, 2011.
3. I. Dagan, O. Glickman, and B. Magnini. "The Pascal Recognising Textual Entailment Challenge," In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.
4. M. Denkowski and M. Lavie. 2010. "Extending the METEOR Machine Translation Metric to the Phrase Level," In Proceedings of NAACL. 2010.
5. Heilman, M. and Smith, N.A. "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010.
6. Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu "BLEU: a method for automatic evaluation of machine translation," In Proceedings of the 40th ACL. Philadelphia, PA, pages 311-318. 2002.
7. Snover, M.G., Madnani, N., Dorr, B., Schwartz, R.: "Ter-plus: Paraphrase, semantic, and alignment enhancements to Translation Edit Rate," 2010.
8. Zhang, Y. and Patrick, J., "Paraphrase identification by text canonicalization," Proceedings of the Australasian Language Technology Workshop. 2005.
9. Klein, Dan and Christopher D. Manning. "Fast exact inference with a factored model for natural language parsing," In Advances in Neural Information Processing Systems 15 (NISP02), pages 310, 2003.
10. K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. "BLEU: A Method for Automatic Evaluation of Machine Translation," In Proceedings of ACL 2002.
11. M. Snover, N. Madnani, B. Dorr, and R. Schwartz. "TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate," Machine Translation, 23(23):117127, 2009.
12. M. Denkowski and M. Lavie. "Extending the METEOR Machine Translation Metric to the Phrase Level," In Proceedings of NAACL 2010.

Alberta-KXG: Legal Question Answering Using Ranking SVM and Syntactic/Semantic Similarity

Mi-Young Kim, Ying Xu, and Randy Goebel

Dept. of Computing Science, University of Alberta, Edmonton, Canada

{ miyoung2,yx2,rgoebel}@ualberta.ca

Abstract. We provide a summary and analysis of our submission for the Competition on Legal Information Extraction/Entailment (COLIEE) 2014 Task system. The task consists of two phases: legal ad-hoc information retrieval and textual entailment. The first phase requires the identification of relevant Japan civil law articles in the collection to a query. In addition to two unsupervised baseline models (tf-idf and LDA-based IR), we implement a supervised model, Ranking SVM, for the task. The features of the model are set of words, scores of an article based on the corresponding baseline models. The results show that the Ranking SVM model nearly doubles the Mean Average Precision compared with both baseline models. The second phase is to answer "Y" or "N" to previously unseen queries, by comparing the meanings of queries and relevant articles. The features for the phase two are syntactic/semantic similarities and use of negation. The results show that our method, combined with rule-based model and the unsupervised model, outperforms the SVM-based supervised model.

Keywords: legal text mining, question answering, recognizing textual entailment, information retrieval, ranking SVM, Latent Dirichlet Allocation (LDA)

1 Task Description

The Competition on Legal Information Extraction/Entailment (COLIEE) 2014 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. Here **relevant** means, based on the articles, lawyers are able to infer the question's answer.

We implement two unsupervised 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 by a function with term frequency and document frequency as parameters.

The other unsupervised model is a topic model-based IR system. The topic model we employ is the LDA (Latent Dirichlet Allocation) model [2]. An LDA model assumes that there is a set of latent topics for a corpus. A document can have several topics, and words in the document are generated based on the topic distribution of a model. The more similar two documents are, the more similar their topic distributions will be. While the tf-idf model only considers words that are both in the query and the document, the LDA based IR model also considers words that are in the query but not in the document. For example, if "seller" is only in the query and the document is about trading and contains words such as "buy", "customer" etc., then the probability that the document generates "seller" will be high.

Besides implementing both models for the task, we also incorporate them into a Ranking SVM model [3]. That model re-ranks documents that are retrieved by the tf-idf model. The model's features include word types, and both tf-idf score and LDA model score of a document for a query. The intuition is that, by including word types, the SVM model can re-assign weights to word types based on the training instances instead of solely on statistic information. By including both statistical models' scores, the SVM model can take advantage of both models.

Our experiments for phase 1 show that all features contribute to the improvement of IR results. The Ranking SVM model nearly doubles the Mean Average Precision compared with both baseline models.

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 meanings between queries and relevant articles.

Our system uses features that depend on the syntactic structure, the presence of negation, and the semantic similarities of the words. The combined model with rules and unsupervised learning model shows reasonable performance, which improves the baseline system, and outperforms an SVM-based supervised machine learning model.

2. Phase 1: Legal Information Retrieval

2.1 IR Models

In this section, we will introduce our tf-idf model, our LDA-based IR model, and our Ranking SVM model. Queries and articles are all tokenized and parsed by the Stanford NLP tool. For IR task, the similarity of a query and an article is based on the terms in them. Our terms can be a word or a dependency linked word bigrams.

2.1.1 TF-IDF

One of our baseline models is a tf-idf model implemented in Lucene, an open source IR system[1].

The simplified version of Lucene's similarity score of an article to a query is:

[1] Lucene can be downloaded from http://lucene.apache.org/core/.

$$tf - idf(Q, A) = \sum \sqrt{(tf(t, A)} \times (1 + \log(idf(t)))^2 \quad (1)$$

where t is a term in query Q, $tf(t,A)$ is the number of times the term occurs in article A, and idf is the number of articles that contain the term t. The real version has some normalized parameters in terms of an article's length.

2.1.2 LDA-based IR

Our LDA-based IR model was first proposed in [4]. For more details about LDA model please refer to [2]. The score function is:

$$LDA(Q, A) = \prod_{t \in Q} P(t \mid A) \quad (2)$$

$P(t|A)$ is the probability of the term t given the article A, specified as:

$$P(t \mid A) = \sum_{z \in K} P(t \mid z) P(z \mid A) \quad (3)$$

where z is a latent topic in topic set K, $P(t|z)$ is the probability of the term t given topic z, and $P(z|A)$ is the probability of the topic z given the article A. The two probabilities are from an LDA-model trained with the Mallet package[2].

2.1.3 Ranking SVM

The Ranking SVM model was proposed by [3]. He ranked the documents based on user's click through data. Given the feature vector of a training instance, i.e. retrieved article set given a query, $<\Phi(Q,Ai)>$, the model tries to find $\vec{w}$ that satisfies constraints:

$$\vec{w} \times \Phi(Q, A_i) > \vec{w} \times \Phi(Q, A_j) \quad (4)$$

where A_i is a relevant article for the query Q, while A_j is not.

We incorporate four types of features.

- Word types: the lemmatized words in both the retrieved article and the query. In the conversion to the SVM's instance representation, this type is converted into binary features whose values are one or zero, i.e. 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 [5]. This feature type is also converted into binary values.
- tf-idf score.
- LDA-based IR score.

2.2 Experiments

[2] Mallet is a machine learning software package. It can be downloaded from http://mallet.cs.umass.edu/

The following experiments compare different IR models on the legal IR task, verifying whether the ensemble SVM-Ranking model improves the IR performance.

The legal IR task has 4 sets of queries, and we use the corresponding 4-fold leave-one-out cross validation evaluation. The metrics for measuring our IR models is Mean Average Precision (MAP):

$$MAP(Q)=\frac{1}{|Q|}\sum_{q\in Q}\frac{1}{m}\sum_{k\in(1,m)}precision(R_k) \quad (5)$$

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 two values of m, which are 5 and 10, for all queries, corresponding to the column MAP@5 and MAP@10 in Table 1.

The LDA model is trained on the whole Japan civil law dataset. The parameters such as number of training iterations and number of topics are set according to the 10-fold cross validation performance. SVM's parameters are set in a similar manner. The LDA values in the SVM models are from the best LDA-based IR system.

Table 1 presents the results of different models.

Id	models	MAP@10	MAP@5
1	tf-idf with lemma	0.108	0.154
2	LDA-based	0.105	0.147
3	tf-idf with lemma and dependency pair	0.111	0.158
4	SVM-ranking with lemma	0.199	0.287
5	SVM-ranking with lemma and dependency pair	0.204	0.294
6	model 5 plus tf-idf score	0.211	0.306
7	model 5 plus LDA-based score	0.207	0.298
8	model 5 plus LDA-based score plus tf-idf score	**0.214**	**0.312**

Table 1. IR results with different models.

The result shows that including dependency information slightly improves IR. Ranking SVM almost doubles the MAP. It does learn a better term weight for IR systems. Including the baseline tf-idf scores as features slightly improves the performance, and including LDA-based score further improves the performance.

2.3 Discussion

The baseline tf-idf model's performance confirms that legal case information retrieval is a difficult task. One problem is language variability, i.e. different expressions convey the same meaning. For explanation purposes, consider the following example:

Question: In a demand for compensation based on delayed performance of loan claim, if person Y gives proof that the delayed performance is not based on the reasons attributable to him or herself, person Y shall be relieved of the liability.

Related Article: (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.

There is no common content word between the question and the article. An expert can consider several alignments to help decide that the two paragraphs are related: "force majeure'" can be aligned to the negative of "reasons attributable to him or herself;" "agreed interest rate shall prevail" can be aligned to "person Y shall be relieved of the liability;" and "failure to perform" can be aligned to "delay;" etc.

From this example, we can see the difficulty of the task. It should not solely depend on lexical information, but also take semantics into consideration. Any approach to address this problem will require at least a larger corpus from which to build improved semantic models.

3. Phase 2: Answering 'Yes'/'No' Questions based on Textual Entailment

Our system combines different sources of syntactic/semantic sources to predict textual entailment. We exploit syntactic/semantic similarity features, and negation. For negation and antonym patterns, we use M-Y.Kim et al.[6]'s approach.

3.1 Our System

3.1.1 Condition and Conclusion Detection

Previous Textual Entailment systems compute lexical or syntactic similarities between two whole sentences [7-10]. However, the sentences in this task are more complex; for example they often contain components like condition(premise), conclusion, and exceptional case description. Even when two sentences have similar word occurrences, if condition and conclusion are placed in a different order, or if a condition part matches an exceptional case, then the result of textual entailment should be 'no.' Therefore, we first need to identify condition, conclusion, and exceptional sentences in civil law articles, before computing similarity.

We also exploit keywords indicating a condition within training data. The keywords are as follows: 'in case(s) ", "if", "unless", "with respect to", "when", and comma.

The sentences which explain exceptional cases also have keywords such as "provided", and "not apply", and those sentences exist in the last position in an article.

We can split the articles into three parts using these keywords. Condition and conclusion are in the first sentence, but an exceptional case is explained in the seconds sentence. There also exist articles that do not mention exceptional cases.

$$conclusion := segment_{last}(first\ sentence, keyword_{condition}),$$
$$condition := \sum_{i \neq last} segment_i(first\ sentence, keyword_{condition}),$$
$$exception := \text{sec}ond\ sentence\ which\ includes\ keyword_{exception}$$

Condition is the sum of segments which include the keywords for conditions. Conclusion is the last segment in the first sentence. Exceptional case is the second sentence which includes the keywords for exceptional cases.

For example,

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

(1) Condition=> If a possessor of a Thing belonging to another person has a claim that has arisen with respect to that Thing,
(2) Conclusion=> he/she may retain that thing until that claim is satisfied.
(3) Exception=> Provided, however, that this shall not apply if such claim has not yet fallen due.

3.1.2 Textual Entailment between a query sentence and several articles

Previous textual entailment systems compare two sentences and deduce a result [7-10]. However, in our task, a query sentence can have multiple relevant articles, and one article consists of several sentences. So, in order to propose the appropriate textual entailment result, we have to compare the meanings of one query sentence and multiple article sentences (one sentences vs. multiple articles).

Our approach assumes that there exists one most relevant article amongst relevant articles which can answer 'yes' or 'no' for a query sentence. For simplicity, we select the most relevant article and then try to deduce a textual entailment result by comparing between a query sentence and most relevant article (one sentence vs. one article). Here, we will describe how we choose the most relevant article with a query, from previously identified relevant articles:

most relevant article sentence :=

$$\arg\max_{article_{n,j} \in \text{articles relevant with } query_n} \{ overlap(condition_{article_{n,j}}, condition_{query_n}) + overlap(conclusion_{article_{n,j}}, conclusion_{query_n}) \}$$

The basic idea is that sentence $article_{n,j}$ is the most relevant article if it has maximum word overlap with the $query_n$ sentence. When we compute the word overlap, we separately compute the word overlap in condition parts and conclusion parts, and then use their sum.

For this approach to textual entailment, we have tried three methods: applying hand-constructed rules, creating a simple model with unsupervised learning, and then using an SVM-based supervised learning method.

3.1.3 Applying rules

if ($neg\_level(condition_{article\ n}) + neg\_level(conclusion_{article\ n})$
$= neg\_level(condition_{query\ n}) + neg\_level(conclusion_{query\ n})$),
$Answer_n := yes$,
otherwise, $Answer_n := no$,
where *neg_level() := 1 if negation and antonym occur odd number of times.*
neg_level() := 0 otherwise.

Figure 1. Answering rule for easy questions

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

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

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

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

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

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

A more concise description for this rule is shown in Figure 1. In this Figure, $article_n$ is the most relevant article of the query $query_n$. The output of our rule-based system is also used below in an unsupervised learning model for assigning labels of condition (conclusion) clusters.

3.1.4 Unsupervised learning

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

The types of features we use are as follows:

Word matching Having the same lemma.

Tree structure features Considering only the dependents of a root.

Lexical semantic features Having the same Kadokawa [11] thesaurus concept code.

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

$$\text{Feature 1}: if\ w_{root}(condition_{query_n}) = w_{root}(condition_{article_n})$$
$$\text{Feature 2}: if\ w_{root}(conclusion_{query_n}) = w_{root}(conclusion_{article_n})$$
$$\text{Feature 3}: if\ w_{dep\_i}(conclusion_{query_n}) \in W_{dep}(conclusion_{article_n})$$
$$\text{Feature 4}: if\ c_{root}(condition_{query_n}) = c_{root}(article_n, condition_{article_n})$$
$$\text{Feature 5}: if\ c_{root}(conclusion_{query_n}) = c_{root}(article_n, conclusion_{article_n})$$
$$\text{Feature 6}: if\ neg\_level(condition_{query_n}) = neg\_level(condition_{article_n})$$
$$\text{Feature 7}: if\ neg\_level(conclusion_{query_n}) = neg\_level(conclusion_{article_n})$$

In the features above, $article_n$ is the most relevant article of the query $query_n$. Features 1, 2, 3 consider both lexical and syntactic information, and Features 4 and 5 consider semantic information. Features 6 and 7 incorporate negation and antonym information. Features 1 and 2 are used to check if conditions (conclusions) of a question and corresponding article share the same root word in the syntactic tree. Feature 3 is to determine if each dependent of a root in the conclusion of a question appears in the article. We heuristically limit the number of dependents as those three nearest to the root. Features 4 and 5 confirm if the root words of conditions (conclusions) of the question and corresponding article share the same concept code. We use some morphological and syntactic analysis to extract lemma and dependency information. Details of the morphological and syntactic analyzer are given in Section 3.2.

The inputs for our unsupervised learning model are all the questions and corresponding articles. The outputs are two clusters of the questions. The yes/no outputs based on rules described in Section 2.2.3 are used as a key for assigning a yes/no label of each cluster. The cluster which includes higher portion of "yes" of the easy questions is assigned the label "yes," and the other cluster is assigned "no." We determine their yes/no answers using their clustering labels.

3.1.5 Supervised learning with SVM

We compare our method with SVM, as a kind of supervised learning model. Using the SVM tool included in the Weka [12] software, we performed cross-validation for the 179 questions using 7 features explained in Section 2.2.4. We used a linear kernel SVM because it is popular for real-time applications as they enjoy both faster training and classification speeds, with significantly less memory requirements than non-linear kernels because of the compact representation of the decision function.

3.2 Experimental setup for Phase 2

In the general formulation of the textual entailment problem, given an input text sentence and a hypothesis sentence, the task is to make predictions about whether or not the hypothesis is entailed by the input sentence. We report the accuracy of our method in answering yes/no questions of legal bar exams by predicting whether the questions can be entailed by the corresponding civil law articles.

There is a balanced positive-negative sample distribution in the dataset (55.87% yes, and 44.13% no), so we consider the baseline for true/false evaluation is the

accuracy when returning always "yes," which is 55.87%. Our data has 179 questions, with total 1044 civil law articles.

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 [13], which extracts enriched information including the use of the Kadokawa thesaurus for lexical semantic information. We use a simple unsupervised learning method, since the data size is not big enough to separate it into training and test data.

3.3 Experimental results

Our method	Accuracy (%)
Baseline	55.87
Rule-based model	53.77
Unsupervised learning (K-means)	60.85
Cross-validation with Supervised learning (SVM)	59.43
Rule-based model for easy questions + unsupervised learning for non-easy questions (combined)	61.96

Table 2. Experimental results for phase 2

Evaluation of question answering systems is in general almost as complex as question-answering itself. So one must make the choice to consider several features of QA systems in the evaluation process, e.g., query language difficulty, content language difficulty, question difficulty, usability, accuracy, confidence, speed and breadth of domain [14].

Table 2 shows our results. A rule-based model showed accuracy of 53.77%. We also use a K-means clustering algorithm with K=2 for unsupervised learning for the rest of the questions, and it showed accuracy of 60.85%. The overall performance when combining the use of rules and unsupervised learning showed 61.96% of accuracy which outperformed unsupervised learning for all questions, and even SVM (59.43%), the supervised learning model we use with a linear kernel. According to p-value measures (p=0.01) between the baseline and the combined model in the true/false determination, the combined model with rule-based model for easy questions and unsupervised learning model for non-easy questions significantly outperformed the baseline. Since previous methods use supervised learning with

syntactic and lexical information, we consider the supervised learning experiment with SVM in Table 2 approximately represents the performance of previous methods.

3.4 Discussion

From unsuccessful instances, we can see that the challenge of paraphrasing causes most of the errors of our system. It could be addressed by exploiting expert knowledge and much larger corpora. We also found cases that indicate the need to do more extensive temporal analysis.

It will be interesting if we compare our performance using Korean-translated sentences with that using original sentences. We would expect the system using original sentences will show better performance than ours, because there exist no translation errors.

4 Related work

SemEval 2014 Task 1 [15] evaluates system predictions of semantic relatedness (SR) and textual entailment (TE) relations on sentence pairs from the SICK dataset, which consist of 750 images and two descriptions for each image. The top ranked system (UIUC) [16] in this task uses distributional constituent similarity and denotational constituent similarity features. Their method of comparing constituents between the whole two sentences are not useful in our task, because our task consists of one legal query and multiple corresponding article sentences.

There was a textual entailment method from W. Bdour et al. [17] 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 not appropriate for our task. If a false question sentence is constructed by replacing named entities with terms of different meaning in the legal article, a logic representation can be helpful. However, false questions are not simply constructed by substituting specific named entities, and any logical representation can make the problem more complex. Kouylekov and Magnini [18] experimented with various cost functions and found a combination scheme to work the best for RTE. Vanderwende et al. [19] used syntactic heuristic matching rules with a lexical-similarity back-off model. Nielsen et al. [20] extracted features from dependency paths, and combined them with word-alignment features in a mixture of an expert-based classifier. Zanzotto et al. [21] proposed a syntactic cross-pair similarity measure for RTE. Harmeling [22] took a similar classification-based approach with transformation sequence features. Marsi et al. [23] described a system using dependency-based paraphrasing techniques. All previous systems uniformly conclude that syntactic information is helpful in RTE: we also use syntactic information combined with lexical semantic information. As further research, we can enrich our knowledge base with deeper analysis of data, and add paraphrasing dictionary getting help from experts.

5 Conclusion

We have described our systems for the Competition on Legal Information Extraction/Entailment (COLIEE) 2014 Task.

For phase 1, legal information retrieval, we implemented a Ranking-SVM model for the legal information retrieval task. By incorporating features such as word types, dependency links, tf-idf score, and LDA-based IR score, our model doubles the mean average precision.

For phase 2, we have proposed a method to answer yes/no questions from legal bar exams related to civil law. We used the knowledge base of M-Y. Kim et al.[6] by analyzing negation patterns and antonyms in the civil law articles. To make the alignment easy, we first segment questions and articles into condition, conclusion, and exception. We then extract deep linguistic features with lexical, syntactic information based on morphological analysis and dependency trees, and lexical semantic information using the Kadokawa thesaurus. Our method uses a hybrid model which combines a rule-based model for easy questions and unsupervised learning model for non-easy questions. This achieved quite encouraging results in both true and false determination. To improve our approach in future work, we need to create deeper representations (e.g., to deal with paraphrase), and analyze the temporal aspects of legal sentences. In addition, we will complement our knowledge base with paraphrasing dictionary with the help of experts.

Acknowledgements

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

References

1. 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. D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation. J. Mach. Learn. Res .3, 993-1022, March 2003
3. 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
4. X. Wei, W.B. Croft, Lda-based document models for ad-hoc retrieval. In: Proceeding of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 178-185. SIGIR '06, ACM, New York, NY, USA, 2006
5. 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
6. M-Y. Kim, Y. Xu, R. Goebel, K. Satoh, “Answering Yes/No Questions in Legal Bar Exams”, JURISIN 2013

7. V. Jikoun and M. de Rijke. Recognizing textual entailment using lexical similarity. In Proceedings of the PASCAL Challenges Workshop on RTE, 2005
8. B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In Proceedings of HLT-NAACL, 2006
9. R. Sno, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In Proceedings of HLT-NAACL, 2006
10. A. Lai, J. Hockenmaier, "Illinois-LH: A Denotational and Distributional Approach to Semantics", In Proceedings of SemEval 2014: International Workshop on Semantic Evaluation. 2014
11. S. Ohno and M. Hamanishi, New Synonym Dictionary. Kadokawa Shoten, Tokyo, 1981
12. 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
13. 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
14. M. Walas, How to answer yes/no spatial questions using qualitative reasoning?, Proc. of the International Conference on Computational Linguistics and Intelligent Text Processing, pp. 330-341, 2012
15. M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment", In Proceedings of SemEval 2014: International Workshop on Semantic Evaluation. 2014
16. A. Lai, J. Hockenmaier, "Illinois-LH: A Denotational and Distributional Approach to Semantics", In Proceedings of SemEval 2014: International Workshop on Semantic Evaluation. 2014
17. 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
18. 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
19. L. Vanderwende, A. Menezes, and R. Snow. Microsoft research at rte-2: Syntactic contributions in the enta ilment task: an implementation. In Proceedings of the second PASCAL Challenges Workshop on RTE .2006
20. 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
21. 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
22. 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
23. 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

Author Index