Proceedings of the Tenth International Workshop on Juris-informatics (JURISIN 2016)

hosted by JSAI-isAI

Workshop Co-Chairs

Makoto Nakamura
Seiichiro Sakurai
Katsuhiko Toyama

Raiosha Building, Keio University, Kanagawa, Japan,
November 14-15, 2016

Preface

The Tenth International Workshop on Juris-Informatics (JURISIN 2016) is held with a support of the Japanese Society for Artificial Intelligence (JSAI) in association with JSAI International Symposia on AI (JSAI-isAI 2016). JSAI-isAI is a collection of several workshops and seven workshops were selected this year. Although JURISIN was organized to discuss legal issues from the perspective of informatics, the scope of JURISIN is more wide-ranging than that of the conventional AI and law. Thus, the members of Program Committee (PC) are leading researchers in various fields. The collaborative work of computer scientists, lawyers and philosophers is expected to contribute to the advancement of juris-informatics and it is also expected to open novel research areas.

Despite the short announcement period, eighteen papers were submitted. Each paper was reviewed by three or more members of PC. This year, we allow a double submission to JURIX 2016 and one paper was withdrawn because of acceptance to JURIX 2016 and fifteen papers were accepted in total. The collection of papers covers various topics such as legal reasoning, argumentation theory, legal compliance, dispute resolution, application of AI and informatics to law, application of natural language processing and so on. As invited speakers, we have Professor Georg Borges from Saarland University, Germany and Professor Harumichi Yuasa from Institute of Information Security, Japan. Moreover, we have a plenary talk by Professor Fernando Koch, ACM Distinguished Speaker, Invited Professor at Korea University, and Honorary Senior Fellow at the University of Melbourne.

Following the previous years, JURISIN have a session on the third Competition on Legal Information Extraction/Entailment (COLIEE 2016) which consists of a result report of the competition and eight papers for each participant.

Finally, I would like to thank all the members of PC and subreviewers for reviewing the submitted papers.

November, 2016
Yokohama, Japan

Makoto Nakamura
Seiichiro Sakurai
Katsuhiko Toyama

Workshop Organization

Program Co-Chairs

Makoto Nakamura	Nagoya University, Japan
Seiichiro Sakurai	Meiji Gakuin University, Japan
Katsuhiko Toyama	Nagoya University, Japan

Steering Committee Members

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

Thomas Ågotnes	University of Bergen, Norway
Floris Bex	Utrecht University, The Netherlands
Randy Goebel	University of Alberta, Canada
Guido Governatori	NICTA, Australia
Yoichi Hatsutori	IBM Japan Ltd., Japan
Tokuyasu Kakuta	Chuo University, Japan
Yoshinobu Kano	Shizuoka University, Japan
Takehiko Kasahara	Toin Yokohama University, Japan
Mi-Young Kim	University of Alberta, Canada
Beishui Liao	Zhejiang University, China
Makoto Nakamura	Nagoya University, Japan
Le-Minh Nguyen	Japan Advanced Institute of Science and Technology(JAIST), Japan
Katumi Nitta	Tokyo Institute of Technology, Japan
Paulo Novais	University of Minho, Portugal
Julian Padget	University of Bath, UK
Ginevra Peruginelli	ITTIG-CNR, Italy
Seiichiro Sakurai	Meiji Gakuin University, Japan
Katsuhiko Sano	Japan Advanced Institute of Science and Technology(JAIST), Japan
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Akira Shimazu	Japan Advanced Institute of Science and Technology(JAIST), Japan
Satoshi Tojo	Japan Advanced Institute of Science and Technology(JAIST), Japan
Katsuhiko Toyama	Nagoya University, Japan
Katsumasa Yoshikawa	IBM Japan Ltd., Japan

Additional Reviewers

Danilo S. Carvahol

Table of Contents

Electronic Person - rights for robot

Georg Borges

Professor of Law, Saarland University
Email: georg.borges@uni-saarland.de

Abstract. The European Parliament's draft report on Civil Law Rules on Robotics is understood to be aiming at the conception of a legal framework for robots classifying autonomous robots as "electronic persons" with specific rights and obligations. The concept of attributing rights and obligation to a robot instead of to it's owner or producer implicates nothing less than a revolution in law and legal philosophy. In the presenation, the concept will be discussed from a regulatory perpective and the benefits and risks of the concept examined.

Technological Development and Japanese Law regarding Artificial Intelligence

Harumichi Yuasa

Professor of Law, Institute of Information Security
Email: yuasa@iisec.ac.jp

Abstract. Thanks to the development and deepening of cutting-edge technology in AI (Artificial Intelligence) it is becoming possible for AI to create sophisticated creations and perform decision making which previously only a human could. However, the current Japanese legal system gives fact that rights and duties are both based on natural or juridical persons. Hence it is urgently necessary to consider whether to acknowledge the rights of the AI's creation and to whom (or what) responsibilities resulting from the AI's action belongs. The Japanese government is currently engaged in revising our act, rules, and guidelines for medical diagnosis systems and radio waves use by robots. They also need to fundamentally reexamine the rights and duties of AI. To consider and confirm the principles of the whole AI legislation is a challenge for Japanese laws.

A Formal Analysis of Legal Reasoning

Yasuo Nakayama

Osaka University, Graduate School of Human Sciences, JAPAN
nakayama@hus.osaka-u.ac.jp

Abstract. In this paper, we propose a formal analysis of legal reasoning based on a framework for normative systems, namely *Logic for Normative Systems* (LNS). LNS is proposed in Nakayama (2011) and further developed in Nakayama (2014, 2015a, 2015b, 2016). Many articles in a legal code have normative forms. Thus, when we try to express them in a formal style, we are faced with problems how to express normative sentences in a formal framework. For this task, LNS provides a solution. In this paper, we also propose a method for constructing a consistent interpretation of legal codes.

1. Introduction

When we interpret legal codes, there are some principles that should be taken into account. Here, we express them in connection with LNS.

(1a) [Normative system] A law is a normative system. A legal system can be expressed in LNS.

(1b) [Logical consistency of legal systems] A legal system should be interpreted as logically consistent. In this paper, we show that this requirement is not trivial. We interpret the logical consistency as the consistency of a normative system in LNS.

(1c) [Coherence as integrity] Many legal theorists are interested in the role of coherence in legal reasoning (Dickson 2014: sect. 3.1). For example, Dworkin (1986) proposes to interpret law as coherent, in the sense of speaking with one voice (Dickson 2014: sect. 3.4). In this paper, we accept this principle. It justifies applying interpretation theory for communications to interpretation of legal systems. Some legal practitioners might disagree with respect to interpretation of some articles. This difference of interpretations sometimes plays an important role in legal practices.

(1d) [Holistic principle] An appropriate interpretation of laws is holistic. To interpret an article in a legal system, you have to know the whole system. For example, to interpret an article in the Japanese Criminal Law (JCL), you have to know the whole JCL.

(1e) [Ambiguity of some legal terms] Some terms in legal codes are ambiguous with respect to their extensions and can be interpreted in different ways depending on presupposed views.

(1f) [Principle of charity[1] for legal terms] Articles in a legal code contain many legal terms. To keep consistency of an interpretation, some of these terms have to be interpreted in a specific manner. For example, the term 'kill' in Article 199 of JCL should be interpreted more restrictedly than usual. Otherwise, Article 199 for homicide becomes inconsistent with some other articles in JCL, such as Article 36 for self defense.

Based on these principles, we propose a formal analysis of legal reasoning in Section 4, 5, and 6. However, in order to prepare these discussions, we introduce some formal frameworks.

2. Logic for Normative Systems

LNS is a framework for normative reasoning proposed in Nakayama (2011). LNS is based on First-Order Logic (FOL) and it differs from the standard deontic logic that is a type of modal logic. A normative system in LNS consists of a pair $\langle BB, OB\rangle$ of two sets of FOL-sentences. BB and OB are called *Belief Base* and *Obligation Base*, respectively. A belief state is fully determined by BB, while a normative state depends on both of BB and OB. Nakayama (2014, 2015a, 2015b, 2016) show that LNS is applicable to many problems such as formal analysis of some linguistic phenomena and logical descriptions of games. LNS can be defined as follows.

Definition 1. *Let BB and OB be two consistent sets of FOL-sentences. We call $\langle BB, OB\rangle$ a Normative System. We use not, &, or, $\Rightarrow$, $\Leftrightarrow$ as abbreviations of meta-language expressions, not, and, or , if then, if and only if, respectively. Let Γ be a set of FOL-sentences.*

(2a) $cons(\Gamma)$ *means that* Γ *is consistent.*
(2b) $Cn(\Gamma) =_{def} \{\phi : \phi$ *is a FOL-consequence from* $\Gamma\}$.
(2c) $\boldsymbol{B}_{NS}\phi \Leftrightarrow_{def} \phi \in Cn(BB)$.
(2d) $\boldsymbol{M}_{NS}\phi \Leftrightarrow_{def} cons(BB \cup \{\phi\})$.
(2e) $\boldsymbol{O}_{NS}\phi \Leftrightarrow_{def} (cons(BB \cup OB)\ \&\ \phi \in Cn(BB \cup OB)\ \&\ \phi \notin Cn(BB))$.
(2f) $\boldsymbol{F}_{NS}\phi \Leftrightarrow_{def} \boldsymbol{O}_{NS}\neg\phi$.
(2g) $\boldsymbol{P}_{NS}\phi \Leftrightarrow_{def} (cons(BB \cup OB \cup \{\phi\})\ \&\ \phi \notin Cn(BB))$.

[1] The principle of charity is proposed by D. Davidson (2001b) among others. According to Marc A. Joseph (2016), this principle has two components, namely the *principle of (logical) coherence* and the *principle of correspondence*. For us, only the first one is important. Joseph (2016) summarizes this principle as follows: "Assuming that a speaker reasons in accordance with logical laws is neither an empirical hypothesis about a subject's intellectual capacities nor an expression of the interpreter's goodwill toward her subject. Satisfying the norms of rationality is a condition on speaking a language and having thoughts, and hence failing to locate sufficient consistency in someone's behavior means there is nothing to interpret. The assumption that someone is rational is a foundation on which the project of interpreting his utterances rests." (section 2.b.i).

(2h) *NS is consistent* $\Leftrightarrow_{def}$ $cons(BB \cup OB)$.
(2i) $mod(\Gamma) =_{def} \{M : M$ *is a FOL-model of* $\Gamma\}$.

We read LNS-sentences as follows:

(3a) $\mathbf{B}_{NS}\phi$: It is believed in NS that ϕ.
(3b) $\mathbf{M}_{NS}\phi$: It is believed in NS that possibly ϕ.
(3c) $\mathbf{O}_{NS}\phi$: It is obligated in NS that ϕ.
(3d) $\mathbf{F}_{NS}\phi$: It is forbidden in NS that ϕ.
(3c) $\mathbf{P}_{NS}\phi$: It is permitted in NS that ϕ.

In $\mathbf{B}_{NS}\phi$, ϕ expresses a content of the belief based on NS. In $\mathbf{O}_{NS}\phi$, ϕ expresses a content of a normative requirement based on NS. LNS is founded on FOL and the semantics of LNS can be characterized in terms of FOL-sentences. Because of the strong completeness of FOL, we obtain some semantic characterizations of LNS.

Proposition 1. *Let* $NS = \langle BB, OB\rangle$ *be a normative system. Let* Γ *be a set of FOL-sentences.*

(4a) $mod(\Gamma) \subseteq mod(\{\phi\}) \Leftrightarrow \phi \in Cn(\Gamma)$.
(4b) $\boldsymbol{B}_{NS}\phi \Leftrightarrow mod(BB) \subseteq mod(\{\phi\})$.
(4c) $\boldsymbol{M}_{NS}\phi \Leftrightarrow \exists M \in mod(BB)\ (M \in mod(\{\phi\}))$.
(4d) $\boldsymbol{O}_{NS}\phi \Leftrightarrow \exists M(M \in mod(BB \cup OB)\ \&\ mod(BB \cup OB) \subseteq mod(\{\phi\})\ \&\ \exists M \in mod(BB)(M \notin mod(\{\phi\})))$.
(4e) $\boldsymbol{P}_{NS}\phi \Leftrightarrow \exists M \in mod(BB \cup OB)(M \in mod(\{\phi\})\ \&\ \exists M \in mod(BB)(M \notin mod(\{\phi\})))$.

According to (4b), $\mathbf{B}_{NS}\phi$ means that ϕ is true in every model of belief base BB. Because of (4d), it holds: If $\mathbf{O}_{NS}\phi$, then ϕ is true in every model of $BB \cup OB$ and ϕ is not believed in NS. Thus, we interpret $\mathbf{O}_{NS}\phi$ as a normative requirement for realization of a state expressed by ϕ.

From Definition 1, we obtain some theorems of LNS. These theorems are expressed in a meta-language of FOL. They can be considered as inference schema.

Proposition 2. *Let* NS *be a consistent normative system. Then, the following LNS-theorems hold.*

(5a1) $\boldsymbol{O}_{NS}\phi \Rightarrow \boldsymbol{P}_{NS}\phi$.
(5a2) $\boldsymbol{F}_{NS}\phi \Rightarrow$ *not* $\boldsymbol{P}_{NS}\phi$.
(5a3) $\boldsymbol{P}_{NS}\phi \Rightarrow$ *not* $\boldsymbol{F}_{NS}\phi$.
(5a4) $\boldsymbol{O}_{NS}\phi \Rightarrow$ *not* $\boldsymbol{B}_{NS}\phi$.
(5a5) $\boldsymbol{B}_{NS}\phi \Rightarrow (\boldsymbol{M}_{NS}\phi$ & *not* $\boldsymbol{O}_{NS}\phi$ & *not* $\boldsymbol{F}_{NS}\phi$ & *not* $\boldsymbol{P}_{NS}\phi)$.
(5b1) $(\boldsymbol{O}_{NS}(\phi \rightarrow \psi)\ \&\ \boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{O}_{NS}\psi$.
(5b2) $(\boldsymbol{O}_{NS}(\phi \rightarrow \psi)\ \&\ \boldsymbol{O}_{NS}\phi) \Rightarrow \boldsymbol{O}_{NS}\psi$.
(5b3) $(\boldsymbol{O}_{NS}(\phi \wedge \psi)$ & *not* $\boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{O}_{NS}\phi$.
(5b4) $(\boldsymbol{O}_{NS}(\phi \wedge \psi)\ \&\ \boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{O}_{NS}\psi$.
(5b5) $(\boldsymbol{O}_{NS}(\phi \vee \psi)\ \&\ \boldsymbol{B}_{NS}\neg\phi) \Rightarrow \boldsymbol{O}_{NS}\psi$.

(5b6) $(\boldsymbol{O}_{NS}(\phi \vee \psi)\ \&\ \boldsymbol{F}_{NS}\phi) \Rightarrow \boldsymbol{O}_{NS}\psi$.
(5b7) $(\boldsymbol{O}_{NS}\phi\ \&\ \boldsymbol{B}_{NS}\neg(\phi \wedge \psi)) \Rightarrow \mathit{not}\ \boldsymbol{P}_{NS}\psi$.
(5c1) $\boldsymbol{O}_{NS}(\phi \rightarrow \neg\psi) \Leftrightarrow \boldsymbol{F}_{NS}(\phi \wedge \psi)$.
(5c2) $(\boldsymbol{O}_{NS}(\phi \rightarrow \neg\psi)\ \&\ \boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{F}_{NS}\psi$.
(5c3) $(\boldsymbol{F}_{NS}(\phi \wedge \psi)\ \&\ \boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{F}_{NS}\psi$.
(5c4) $(\boldsymbol{F}_{NS}(\phi \wedge \psi)\ \&\ \boldsymbol{O}_{NS}\phi) \Rightarrow \boldsymbol{F}_{NS}\psi$.
(5d1) $(\boldsymbol{P}_{NS}(\phi \rightarrow \psi)\ \&\ \boldsymbol{B}_{NS}\phi) \Rightarrow \boldsymbol{P}_{NS}\psi$.
(5d2) $(\boldsymbol{P}_{NS}(\phi \rightarrow \psi)\ \&\ \boldsymbol{P}_{NS}\phi) \Rightarrow \boldsymbol{P}_{NS}\psi$.
(5d3) $(\boldsymbol{P}_{NS}(\phi \vee \psi)\ \&\ \boldsymbol{P}_{NS}\neg\phi) \Rightarrow \boldsymbol{P}_{NS}\psi$.
(5d4) $(\boldsymbol{P}_{NS}(\phi \vee \psi)\ \&\ \boldsymbol{F}_{NS}\phi) \Rightarrow \boldsymbol{P}_{NS}\psi$.
(5d5) $(\boldsymbol{P}_{NS}\phi\ \&\ \mathit{not}\ \boldsymbol{B}_{NS}(\phi \vee \psi)) \Rightarrow \boldsymbol{P}_{NS}(\phi \vee \psi)$.
(5e1) $(\boldsymbol{O}_{NS}\forall x_1...x_n(\phi(x_1,...,x_n) \rightarrow \psi(x_1,...,x_n))\ \&\ \boldsymbol{B}_{NS}\phi(a_1,...,a_n)\ \&\ \mathit{not}\ \boldsymbol{B}_{NS}\psi(a_1,...,a_n)) \Rightarrow \boldsymbol{O}_{NS}\psi(a_1,...,a_n)$.
(5e2) $(\boldsymbol{O}_{NS}\forall x_1...x_n(\phi(x_1,...,x_n) \rightarrow \neg\psi(x_1,...,x_n)) \Leftrightarrow \boldsymbol{F}_{NS}\exists x_1...x_n(\phi(x_1,...,x_n) \wedge \psi(x_1,...,x_n))$
(5e3) $(\boldsymbol{F}_{NS}\exists x_1...x_n(\phi(x_1,...,x_n) \wedge \psi(x_1,...,x_n))\ \&\ \boldsymbol{B}_{NS}\phi(a_1,...,a_n)\ \&\ \mathit{not}\ \boldsymbol{B}_{NS}\neg\psi(a_1,...,a_n)) \Rightarrow \boldsymbol{F}_{NS}\psi(a_1,...,a_n)$.
(5e4) $(\boldsymbol{P}_{NS}\forall x_1...x_n(\phi(x_1,...,x_n) \rightarrow \psi(x_1,...,x_n))\ \&\ \boldsymbol{B}_{NS}\phi(a_1,...,a_n)\ \&\ \mathit{not}\ \boldsymbol{B}_{NS}\psi(a_1,...,a_n)) \Rightarrow \boldsymbol{P}_{NS}\psi(a_1,...,a_n)$.
(5f1) $(NS1 = \langle BB_1, OB\rangle\ \&\ NS2 = \langle BB_2, OB\rangle\ \&\ BB_1 \subseteq BB_2) \Rightarrow (\boldsymbol{B}_{NS1}\phi \Rightarrow \boldsymbol{B}_{NS2}\phi)$.
(5f2) $(NS1 = \langle BB_1, OB\rangle\ \&\ NS2 = \langle BB_1 \cup \{\phi\}, OB\rangle\ \&\ \boldsymbol{P}_{NS1}\phi) \Rightarrow NS2$ *is consistent.*
(5f3) $(NS1 = \langle BB_1, OB\rangle\ \&\ NS2 = \langle BB_1 \cup \{\phi\}, OB\rangle\ \&\ \boldsymbol{O}_{NS1}\phi) \Rightarrow NS2$ *is consistent.*

Proof. We show only (5b1). Suppose that $\mathbf{O}_{NS}(\phi \rightarrow \psi)\ \&\ \mathbf{B}_{NS}\phi$. Then, from Definition 1, $cons(BB \cup OB)\ \&\ (\phi \rightarrow \psi) \in Cn(BB \cup OB)\ \&\ (\phi \rightarrow \psi) \notin Cn(BB)\ \&\ \phi \in Cn(BB)$. Then, $\psi \in Cn(BB \cup OB)$. Now, suppose that $\psi \in Cn(BB)$. Then, $(\phi \rightarrow \psi) \in Cn(BB)$. This contradicts to $(\phi \rightarrow \psi) \notin Cn(BB)$. Thus, $\psi \notin Cn(BB)$. Then, we obtain $\mathbf{O}_{NS}\psi$. Hence, (4b1) holds. Other theorems can be proved in the same way. Q.E.D.

Nakayama (2014, 2015b, 2016) propose *Dynamic Nomative Logic* (DNL). In DNL, the belief base can be updated. A normative system in DNL involves information about its stage. We write a normative system of DNL as follows: $NS(n) = \langle BB(n), OB\rangle$. Any normative system in DNL is a normative system in LNS. Thus, Definition 1, Proposition 1, and Proposition 2 hold for DNL.

3. Application of Dynamic Normative Logic

LNS and DNL are based on the following ideas:

(6a) If ϕ is not the case and ϕ is required, we should perform an action in order to make ϕ true.

(6b) If ϕ is the case, there is no need for normative actions with respect to ϕ (See Proposition (5a5)).

This dependency of normative states on belief states is needed to describe dynamic changes of normative requirements. To clarify these features of DNL, we consider a simple scene in a restaurant described by van Benthem (2011: p. 4). We shorten the story and we assume that two guests instead of three guests ordered their meals.

> In a restaurant, your Mother ordered Vegetarian, and you have Meat. Out of the kitchen comes some new person carrying the two plates. What will happen?

To clarify update processes, we divide belief base $BB(n)$ into two parts, namely elementary theory ET and a set of facts $FACT(n)$. Thus, it holds, $BB(n) = ET \cup FACT(n)$ & $ET \cap FACT(n) = \emptyset$. In this example, only $FACT(n)$ is updated.

The following sequence describes one of possible developments of the story.

1. The waiter asks, 'Who has the Meat?'
2. The boy says 'Me'.
3. The waiter serves him with the meat plate.
4. The waiter serves the mother with the vegetarian plate without asking.

To describe this scene within DNL, each component of NS in this story must be made explicit.[2]

(7) Elementary Theory: $ET_G = \{$(7a), (7b), (7c), (7d)$\}$ & $ET_w = ET_G \cup \{(7e)\}$.
(7a) Simple Set Theoretical Principles.
(7b) $\forall x((guest(x) \wedge x \in Family) \rightarrow \exists^{=1} y(plate(y) \wedge ordered(x, y)))$.
(7c) $\forall x \forall y((guest(x) \wedge \exists n\ \underline{answer}(x, y, n)) \rightarrow ordered(x, y))$.
(7d) $\forall x \forall G_1 \forall n((served(G_1, n) \wedge guest(x) \wedge \exists y \exists z(waiter(y) \wedge plate(z) \wedge \underline{serve}(y, x, z, n)) \rightarrow served(G_1 \cup \{x\}, n + 1))$.
(7e) $\forall x \forall y \forall D \forall n((waiter(x) \wedge plate(y) \wedge have\text{-}plate(x, D, n) \wedge y \in D \wedge \exists z(guest(z) \wedge \underline{serve}(x, z, y, n))) \rightarrow have\text{-}plate(x, D - \{y\}, n + 1))$.

(8) Obligation Base: $OB = \{(8a), (8b)\}$
(8a) $\forall x \forall y \forall z((guest(x) \wedge ordered(x, y) \wedge waiter(z)) \rightarrow \forall n\ (\underline{ask\text{-}order}(z, Family, y, n) \rightarrow \underline{answer}(x, y, n + 1)))$.
(8b) $\forall x \forall y \forall z((waiter(x) \wedge guest(y) \wedge ordered(y, z)) \rightarrow \forall D \forall n\ ((have\text{-}plate(x, D, n) \wedge z \in D) \rightarrow \underline{serve}(x, y, z, n)))$.

(9a) Initial State:

[2] Underlined relation symbols in this section stand for actions. Actions are special in the sense that they can change some states.

$FACT_G(0) = \{Family = \{b, m\}, guest(b), guest(m), Plate = \{meat, vegetarian\}, served(\emptyset, 0), waiter(w)\}$.
$FACT_b(0) = \{ordered(b, meat)\}$.
$FACT_m(0) = \{ordered(m, vegetarian)\}$.
$FACT_w(0) = \{have\text{-}plate(w, Plate, 0)\}$.

(9b) Normative systems on state n: In this story, $FACT_G(n)$ is updated along the development of the situation.
$G(n) = \langle BB_G(n), OB\rangle$, where $BB_G(n) = ET_G \cup FACT_G(n)$.
$boy(n) = \langle BB_G(n) \cup FACT_b(0), OB\rangle$.
$mother(n) = \langle BB_G(n) \cup FACT_m(0), OB\rangle$.
$waiter(n) = \langle BB_G(n) \cup \{(7e)\} \cup FACT_w(0), OB\rangle$.

Based on (9b), we can easily show that $\mathbf{B}_{G(n)}$ expresses a shared belief among three people in the story, namely it holds: $\mathbf{B}_{G(n)}\phi \Rightarrow (\mathbf{B}_{waiter(n)}\phi \,\&\, \mathbf{B}_{boy(n)}\phi \,\&\, \mathbf{B}_{mother(n)}\phi)$. The consistency of $G(n)$ and so on can be proved by constructing small models.

We propose to interpret the restaurant story as a game played by the waiter and two guests who are cooperative with the waiter. We assume here that each of players obeys and performs any obligation that is required in each situation. It is the goal of this game that the waiter correctly distributes all plates he had at the initial state. Now, we can describe the development of the game with help of DNL.

(10a) At first, the waiter does not know which meal the boy and the mother ordered ($not\ \mathbf{B}_{waiter(0)}ordered(b, meat)$ & $not\ \mathbf{B}_{waiter(0)}ordered(b, vegitarian)$ & $not\ \mathbf{B}_{waiter(0)}ordered(m, meat)$ & $not\ \mathbf{B}_{waiter(0)}ordered(m, vegitarian)$). By constructing a finite model, we can prove: $\mathbf{P}_{waiter(0)}$ $ask\text{-}order(w, Family, meat, 0)$. Thus, the waiter asks, 'Who has the Meat?': $FACT_{G(1)} = FACT_{G(0)} \cup \{ask\text{-}order(w, Family, meat, 0)\}$.[3] Then, because of (8a) and (9a): $\mathbf{O}_{boy(1)}$ $answer(b, meat, 0)$. Following this obligation, the boy says 'Me': $FACT_{G(2)} = FACT_{G(1)} \cup \{answer(b, meat, 0)\}$. Now, because of (7c) and (9b): $\mathbf{B}_{waiter(2)}$ $ordered(b, meat)$, which means that the waiter realizes that the boy ordered meat. Then, from (8b) follows: $\mathbf{O}_{waiter(2)}$ $serve(w, b, meat, 0)$. Following this obligation, the waiter serves the boy with the meat plate: $FACT_{G(3)} = FACT_{G(2)} \cup \{serve(w, b, meat, 0)\}$. Now, from (7d) and (7e) follows: $\mathbf{B}_{G(3)}served(\{b\}, 1)$ & $\mathbf{B}_{waiter(3)}$ $have\text{-}plate(w, \{vegetarian\}, 1)$.

(10b) In the second stage, the waiter infers who ordered the vegetarian plate without asking. Because of (7b) and (9a): $\mathbf{B}_{waiter(3)}ordered(m, vegetarian)$. Thus, $\mathbf{O}_{waiter(3)}serve(w, m, vegetarian, 2)$. Following this obligation, the waiter serves the mother with the vegetarian: $FACT_{G(4)} = FACT_{G(3)} \cup$

[3] When the water asks who ordered the meat, he get an answer so that he can know the answer to this question. This can be seen as an *abductive* inference An abduction is also called *inference to the best explanation* (Douven 2011). In a game, it is usual that players try to perform the best method for achieving their goals in the given situation. When an agent believes that action act_k is the method to achieve a goal, it is rational for him to perform act_k.

$\{serve(w, m, vegetarian, 1)\}$. Then, we obtain: $\mathbf{B}_{G(4)}served(\{b, m\}, 2)$ & $\mathbf{B}_{waiter(4)}$*have-plate*$(w, \emptyset, 2)$. This shows that the waiter realized that he had accomplished his current task.

Now, you may recognize that this interaction in the restaurant is similar to many *language games* described in Wittgenstein (1953). Actually, simple language games can be described within DNL (Nakayama 2016). Nakayama (2011, 2016) suggest that a trial can be interpreted as a game. We may consider that rules of games in Japanese criminal trials are described in the *Japanese Code of Criminal Procedure*.

4. Interpretation of Japanese Criminal Law

In this section, we propose a LNS-interpretation of some articles of JCL. To translate JCL articles into FOL sentences, we accept an event ontology proposed by Donald Davidson (2001a), because it is useful for a compact FOL-formulation of articles. We consider events as entities and we quantify over events. Furthermore, following Davidson (2001a), we consider actions as a type of events.

At first, we assume a normative system for JCL, namely $NS(JCL) = \langle BB_{JCL}, OB_{JCL}\rangle$. As an example, we consider an application of Article 199 and Article 41.

[**Infancy** Article 41]
An act of a person less than 14 years of age is not punishable.
[**Homicide** Article 199]
A person who kills another shall be punished by the death penalty or imprisonment with work for life or for a definite term of not less than 5 years.

We interpret these articles as obligation statements for the Japanese Legal Institution (JLI). Furthermore, we consider $[kill]_{JCL}$ as a special term for JCL. Then, we require for $NS(JCL)$ the following conditions:

(11a) BB_{JCL} includes the following FOL-sentences as its elements:
$\forall x \forall y \forall e([kill]_{JCL}(x, y, e) \rightarrow kill(x, y, e))$.
('kill' in sense of JCL is more restricted than 'kill' in the ordinary language.)
$\neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge ([act : A41]_{JCL}(x, e) \wedge age(x) < 14years))$.
(Any action of $[kill]_{JCL}$ is not an action described in Article 41.)
(11b) OB_{JCL} includes the following FOL-sentence as an element:
(A41) $\forall x((\exists e[act : A41]_{JCL}(x, e) \wedge age(x) < 14years) \rightarrow \neg\exists s\ punish(JLI, x, s))$.
(JLI does not punish any teenager who is less than 14 years of age and performed an action described in Article 41.)
(A199) $\forall x(\exists y \exists e[kill]_{JCL}(x, y, e) \rightarrow (punish(JLI,\ x,\ \textit{death-penalty}) \vee punish(JLI,\ x,\ \textit{imprisonment-for-life}) \vee \exists t(punish(JLI,\ x, imprisonment(t)) \wedge t \geq 5years))$.
(Any person who killed another person in sense of Article 199 is punished by the death penalty or imprisonment with work for life or for a definite term of not less than 5 years.)

(11a) expresses an interpretation of JCL-articles based on the principle of charity, while (11b) expresses contents of JCL-articles. When we literally read Article 41 and 199, we can think up some incidents that make both articles mutually inconsistent. If a boy of 13 years of age killed a person, then both articles seem simultaneously not applicable to this incident. In other words, when we apply both to it, we obtain two mutually inconsistent obligations with respect to punishment. Thus, to avoid this kind of inconsistency, we need the constraints expressed in (11a).

We supplement this list by simplified formulations of related articles in JCL. They are Articles 35, 36, 37, 202, 205, and 210.

[**Justifiable Acts** Article 35]
An act performed in accordance with laws and regulations or in the pursuit of lawful business is not punishable.

[**Self Defense** Article 36]
(1) An act unavoidably performed to protect the rights of oneself or any other person against imminent and unlawful infringement is not punishable.
(2) An act exceeding the limits of self-defense may lead to the punishment being reduced or may exculpate the offender in light of the circumstances.

[**Averting present Danger** Article 37]
(1) An act unavoidably performed to avert a present danger to the life, body, liberty or property of oneself or any other person is not punishable only when the harm produced by such act does not exceed the harm to be averted; provided, however, that act causing excessive harm may lead to the punishment being reduced or may exculpate the offender in light of the circumstances.
(2) The preceding paragraph does not apply to a person under special professional obligation.

[**Inducing or Aiding Suicide; Homicide with Consent** Article 202]
A person who induces or aids another to commit suicide, or kills another at the other 痴 request or with other 痴 consent, shall be punished by imprisonment with or without work for not less than 6 months but not more than 7 years.

[**Injury Causing Death** Article 205]
A person who causes another to suffer injury resulting in death shall be punished by imprisonment with work for a definite term of not less than 3 years.

[**Causing Death through Negligence** Article 210]
A person who causes the death of another through negligence shall be punished by a fine of not more than 500,000 yen.

Main ideas of these articles can be expressed in LNS. We focus on the aspect how these articles constrain applicability of Article 199.

(12a) BB_{JCL} includes the following FOL-sentences as its elements:
$\forall x \forall y \forall e([kill]_{JCL}(x, y, e) \rightarrow kill(x, y, e))$.
$\neg \exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge\ [\textit{justifiable act: A35}]_{JCL}(x, e))$.

$\neg\exists x\exists e(\exists y[kill]_{JCL}(x,y,e)\wedge$ [*self defense: A36.(1)*]$_{JCL}(x,e))$.
$\neg\exists x\exists e(\exists y[kill]_{JCL}(x,y,e)\wedge$ [*self defense: A36.(2)*]$_{JCL}(x,e))$.
$\neg\exists x\exists e(\exists y[kill]_{JCL}(x,y,e)\wedge$ [*Averting present Danger: A37.(1a)*]$_{JCL}(x,e))$.
$\neg\exists x\exists e(\exists y[kill]_{JCL}(x,y,e)\wedge$ [*Averting present Danger: A37.(1b)*]$_{JCL}(x,e))$.
Let $* = a$ or $* = b$. Let [*A37.(1*)*]$_{JCL(}x,e) :=$ [*Averting present Danger: A37.(1*)*]$_{JCL}(x,e)$ and *person-spo*]$_{JCL}(x,e) :=$ [*person-with-special-professional-obligation*]$_{JCL}(x,e)$.
$\neg\exists x\exists e([$*person-spo*$]_{JCL}(x,e)\wedge$ [*A37.(1*)*]$_{JCL}(x,e))$.[4]
$\neg\exists x\exists e(\exists y[kill]_{JCL}(x,y,e)\wedge([act:A41]_{JCL}(x,e)\wedge age(x) < 14years))$.
$\neg\exists x\exists e\exists y([kill]_{JCL}(x,y,e)\wedge$ [*Homicide with Consent: A202*]$_{JCL}(x,y,e))$.
$\neg\exists x\exists e\exists y([kill]_{JCL}(x,y,e)\wedge$ [*Injury Causing Death: A205*]$_{JCL}(x,y,e))$.
$\neg\exists x\exists e\exists y([kill]_{JCL}(x,y,e)\wedge$
[*Causing Death through Negligence: A210*]$_{JCL}(x,y,e))$.

(12b) OB_{JCL} includes the following FOL-sentences as its elements:

(A35) $\forall x(\exists e$ [*justifiable act: A35*]$_{JCL}(x,e) \rightarrow \neg\exists s\ punish(JLI,x,s))$.
(A36.(1) $\forall x(\exists e$ [*self defense: A36.(1)*]$_{JCL}(x,e) \rightarrow \neg\exists s\ punish(JLI,x,s))$.
(A36.(2)) $\forall x(\exists e$ [*self defense: A36.(2)*]$_{JCL}(x,e) \rightarrow$
$\exists s\ reduced[punish](JLI,x,s))$. [5]
(A37.(1a)) $\forall x(\exists e$ [*Averting present Danger: A37.(1a)*]$_{JCL}(x,e) \rightarrow$
$\neg\exists s\ punish(JLI,x,s))$.
(A37.(1b)) $\forall x(\exists e$ [*Averting present Danger: A37.(1b)*]$_{JCL}(x,e) \rightarrow$
$\exists s\ reduced[punish](JLI,x,s))$.
(A39.(1a)) $\forall x(\exists e$ [*Insanity: A39.(1a)*]$_{JCL}(x,e) \rightarrow \neg\exists s\ punish(JLI,x,s))$.
(A39.(1b)) $\forall x(\exists e$ [*Diminished Capacity: A39.(1b)*]$_{JCL}(x,e) \rightarrow$
$\exists s\ reduced[punish](JLI,x,s))$.
(A41) $\forall x((\exists e[act:A41]_{JCL}(x,e)\wedge age(x) < 14years) \rightarrow$
$\neg\exists s\ punish(JLI,x,s))$.
(A199) $\forall x(\exists y\exists e[kill]_{JCL}(x,y,e) \rightarrow (punish($*JLI*, x, *death-penalty*$)\vee$
$punish($*JLI*, x, *imprisonment-for-life*$)\vee$
$\exists t(punish($*JLI*$, x, imprisonment(t))\wedge t \geq 5years))$.
(A202) $\forall x(\exists y\exists e$ [*Homicide with Consent: A202*]$_{JCL}(x,y,e) \rightarrow$
$\exists t(punish(JLI,x,$ *imprisonment*$(t))\wedge$ *6months* $\leq t <$ *5years*$))$.
(A205) $\forall x(\exists y\exists e$ [*Injury Causing Death: A205*]$_{JCL}(x,y,e) \rightarrow$
$\exists t(punish(JLI,x,$ *imprisonment*$(t))\wedge$ *3years* $\leq t))$.
(A210) $\forall x(\exists y\exists e$ [*Causing Death through Negligence: A210*]$_{JCL}(x,y,e) \rightarrow$
$\exists$*fine* $(punish(JLI,x,$ *fine*$)\wedge$ *fine* $\leq$ *500,000yen*$))$.

[4] Article 37 (2) says: "The preceding paragraph does not apply to a person under special professional obligation". This kind of cancelation is expressed by the incompatibility of the antecedents of Article 37.(1) and *person-with-special-professional-obligation*.

[5] The expression 'reduced' is used as an adverb. It has a function of reducing the punishment that is determined without consideration of the given circumstances.

As you can see in (12a), 'homicide' in JCL has a very restricted meaning. In principle, any interpretation of [kill]$_{JCL}$ that satisfies all constraints expressed in (12a) is logically consistent with JCL and permissible in this sense.

Properly, we should take Article 38 [Intent], Article 66, 67 [Reduction of Punishment in Light of Extenuating Circumstances], and Article 211 [Causing Death or Injury through Negligence in the Pursuit of Social Activities] into consideration. However, we skip discussions on these articles for lack of space.

(12a) shows that an interpretation of a legal code should be holistic. Suppose that you do not know Article 202 [Inducing or Aiding Suicide; Homicide with Consent] and believe $\exists e[kill]_{JCL}(B, A, e)$, even if B killed A because of A's requirement for killing himself. In this case, your judgment is wrong, because Article 202 should be applied to this case. As this example shows, you have to know the whole JCL, in order to be sure that your judgment is justified.

5. Conclusions from *NS*(JCL)

From the formal description of JCL in section 4, we obtain some LNS-consequences. Here, we express some of them. To simplify notations, we use $\mathbf{B}_{JCL}\ \phi$ and $\mathbf{O}_{JCL}\ \phi$ instead of $\mathbf{B}_{NS(JCL)}\ \phi$ and $\mathbf{O}_{NS(JCL)}\ \phi$. Furthermore, we assume here that NS(JCL) is consistent.

(13a) We have the following consequences for the belief state of $NS(JCL)$.
$\mathbf{B}_{JCL}\ \forall x \forall y \forall e([kill]_{JCL}(x, y, e) \rightarrow kill(x, y, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge [justifiable\ act:\ A35]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge [self\ defense:\ A36.(1)]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge [self\ defense:\ A36.(2)]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge$
$[Averting\ present\ Danger:\ A37.(1a)]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge$
$[Averting\ present\ Danger:\ A37.(1b)]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e([person\text{-}spo]_{JCL}(x, e) \wedge [A37.(1^*)]_{JCL}(x, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e(\exists y[kill]_{JCL}(x, y, e) \wedge ([act : A41]_{JCL}(x, e) \wedge age(x) < 14 years))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e \exists y([kill]_{JCL}(x, y, e) \wedge [Homicide\ with\ Consent:\ A202]_{JCL}(x, y, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e \exists y([kill]_{JCL}(x, y, e) \wedge [Injury\ Causing\ Death:\ A205]_{JCL}(x, y, e))$.
$\mathbf{B}_{JCL}\ \neg\exists x \exists e \exists y([kill]_{JCL}(x, y, e) \wedge$
$[Causing\ Death\ through\ Negligence:\ A210]_{JCL}(x, y, e))$.

(13b) We have the following consequences for the normative state of $NS(JCL)$.
$\mathbf{F}_{JCL}\ \exists x(\exists e\ [justifiable\ act:\ A35]_{JCL}(x, e) \wedge \exists s\ punish(JLI, x, s))$.
$\mathbf{F}_{JCL}\ \exists x(\exists e\ [self\ defense:\ A36.(1)]_{JCL}(x, e) \wedge \exists s\ punish(JLI, x, s))$.
$\mathbf{O}_{JCL}\ \forall x(\exists e\ [self\ defense:\ A36.(2)]_{JCL}(x, e) \rightarrow$
$\exists s\ reduced[punish](JLI, x, s))$.
$\mathbf{F}_{JCL}\ \forall x(\exists e\ [Averting\ present\ Danger:\ A37.(1a)]_{JCL}(x, e) \wedge \exists s\ punish(JLI, x, s))$.
$\mathbf{O}_{JCL}\ \forall x(\exists e\ [Averting\ present\ Danger:\ A37.(1b)]_{JCL}(x, e) \rightarrow$
$\exists s\ reduced[punish](JLI, x, s))$.
$\mathbf{F}_{JCL}\ \exists x(\exists e\ [Insanity:\ A39.(1a)]_{JCL}(x, e) \wedge \exists s\ punish(JLI, x, s))$.

$\mathbf{O}_{JCL}\ \forall x(\exists e\ [\textit{Diminished Capacity: A39.(1b)}]_{JCL}(x,e) \rightarrow$
$\exists s\ \textit{reduced}[\textit{punish}](JLI,x,s)).$
$\mathbf{F}_{JCL}\ \exists x((\exists e[\textit{act}: A41]_{JCL}(x,e) \wedge\ \textit{age}(x) < 14\textit{years}) \wedge\ \exists s\ \textit{punish}(JLI,x,s)).$
$\mathbf{O}_{JCL}\ \forall x(\exists y \exists e[\textit{kill}]_{JCL}(x,y,e) \rightarrow$
$(\textit{punish}(JLI,x,\ \textit{death-penalty}) \vee$
$\textit{punish}(JLI,x,\ \textit{imprisonment-for-life}) \vee$
$\exists t(\textit{punish}(JLI,x,\textit{imprisonment}(t)) \wedge t \geq 5\textit{years})).$
$\mathbf{O}_{JCL}\ \forall x(\exists y \exists e\ [\textit{Homicide with Consent: A202}]_{JCL}(x,y,e) \rightarrow$
$\exists t(\textit{punish}(JLI,x,\ \textit{imprisonment}(t)) \wedge\ \textit{6months} \leq t < \textit{5years})).$
$\mathbf{O}_{JCL}\ \forall x(\exists y \exists e\ [\textit{Injury Causing Death: A205}]_{JCL}(x,y,e) \rightarrow$
$\exists t(\textit{punish}(JLI,x,\ \textit{imprisonment}(t)) \wedge\ \textit{3years} \leq t)).$
$\mathbf{O}_{JCL}\ \forall x(\exists y \exists e\ [\textit{Causing Death through Negligence: A210}]_{JCL}(x,y,e) \rightarrow$
$\exists \textit{fine}\ (\textit{punish}(JLI,x,\ \textit{fine}) \wedge\ \textit{fine} \leq\ \textit{500,000yen})).$

In this paper, we assume that practitioners of JCL can perform epistemic and normative judgments described in (13a) and (13b).

6. Analysis of Legal Syllogism

Many legal theorists consider that the legal syllogism is sound and useful (MacCormick 2009). Indeed, the legal syllogism has certain applicability in legal practices. However, as the discussion in section 5 shows, the matter is not so simple. For we have to interpret articles in a legal code so that they become mutually consistent. In this section, we show how to analyze the legal syllogism based on LNS and DNL.

Two LNS-theorems in Proposition 2, namely (5e1) and (5e3) are important for the legal syllogism. Note that (5e1) validates the following inference schema:

$\mathbf{O}_{NS}\ \forall x_1 ... \forall x_n(\phi(x_1,...,x_n) \rightarrow \psi(x_1,...,x_n))$
$\mathbf{B}_{NS}\ \phi(a_1,...,a_n)$
$\textit{not}\ \mathbf{B}_{NS}\ \psi(a_1,...,a_n)$

$\mathbf{O}_{NS}\ \psi(a_1,...,a_n))$

In a legal inference, the third condition, $\textit{not}\ \mathbf{B}_{NS}\ \psi(a_1,...,a_n)$, is always fulfilled, because this inference is performed before a decision of the court is declared. In other words, at that time, the punishment with respect to the incident in dispute is not yet determined. Thus, we obtain, from (5e1) and (5e3), the following schema for valid legal syllogism:

$\mathbf{O}_{NS}\ \forall x_1 ... \forall x_n(\phi(x_1,...,x_n) \rightarrow \psi(x_1,...,x_n))$
$\mathbf{B}_{NS}\ \phi(a_1,...,a_n)$

$\mathbf{O}_{NS}\ \psi(a_1,...,a_n)$

$\mathbf{F}_{NS}\ \exists x_1 ... \exists x_n(\phi(x_1, ..., x_n) \wedge \psi(x_1, ..., x_n))$
$\mathbf{B}_{NS}\ \phi(a_1, ..., a_n)$

$\mathbf{O}_{NS}\ \psi(a_1, ..., a_n)$

To illustrate applications of legal reasoning, let us take an example. Suppose that a body of a wealthy businessman A was found in his house in Tokyo. On the next day, his wife B was arrested. Prosecutor C believes that B killed A. Lawyer D of B believes that B defended herself against A's violation. Judge E is not yet sure who is right. This situation can be described within DNL.

Let $NS(X, n) = \langle BB(X, n), OB(X)\rangle$ be the normative system of person X at the stage n. We assume that the prosecutor, the lawyer, and the judge know the whole JCL. Thus, we assume: For any X in $\{C, D, E\}$, $BB_{JCL} \subseteq BB(X, 0)$ & $OB_{JCL} \subseteq OB(X)$. C believes that B killed A in the sense of JCL, while D believes that B defended herself in the sense of JCL. This means: $[kill]_{JCL}(B, A, Incident_1) \in BB(C, 0)$ & $[\textit{self defense: A36.(1)}]_{JCL}(B, Incident_1) \in BB(D, 0)$. Thus, we have the following valid syllogism for prosecutor C.

$\mathbf{O}_{NS(C,0)}\ \forall x(\exists y \exists e[kill]_{JCL}(x, y, e) \rightarrow (\textit{punish}(JLI,\ x,\ \textit{death-penalty}) \vee \textit{punish}(JLI, x, \textit{imprisonment-for-life}) \vee \exists t(\textit{punish}(JLI, x, \textit{imprisonment}(t)) \wedge t \geq \textit{5years}))$.
$\mathbf{B}_{NS(C,0)}\ [kill]_{JCL}(B, A, Incident_1)$.

$\mathbf{O}_{NS(C,0)}\ (\textit{punish}(JLI, B,\ \textit{death-penalty}) \vee \textit{punish}(JLI, B, \textit{imprisonment-for-life}) \vee \exists t(\textit{punish}(JLI, B, \textit{imprisonment}(t)) \wedge t \geq \textit{5years}))$.

This inference in DNL can be seen as a formal representation of the following syllogistic inference in English.

A person who kills another shall be punished by the death penalty or imprisonment with work for life or for a definite term of not less than 5 years.
B killed A in $Incident_1$.

B shall be punished by the death penalty or imprisonment with work for life or for a definite term of not less than 5 years.

For lawyer D, we have the following syllogism:

$\mathbf{F}_{NS(D,0)}\ \exists x(\exists e\ [\textit{self defense: A36.(1)}]_{JCL}(x, e) \wedge \exists s\ \textit{punish}(JLI, x, s))$
$\mathbf{B}_{NS(D,0)}\ [\textit{self defense: A36.(1)}]_{JCL}(B, Incident_1)$

$\mathbf{F}_{NS(D,0)}\ \exists s\ \textit{punish}(JLI, B, s))$

This inference in DNL can be seen as a formal presentation of the following syllogistic inference in English.

An act unavoidably performed to protect the rights of oneself or any other person against imminent and unlawful infringement is not punishable.

B protected herself against violence by A and A incidentally died as a result of B's protection.

B should not be punished.

When the trial starts, prosecutor C provides evidences for his claim, to prove $[kill]_{JCL}(B, A, Incident_1)$. Then, lawyer D criticizes C and try to falsify $[kill]_{JCL}(B, A, Incident_1)$ and to prove $[$*self defense: A36.(1)*$]_{JCL}(B, Incident_1)$. Before a trial starts, judge E does not know which claim is right. Thus, $[kill]_{JCL}$ $(B, A,\text{Incident}_1) \notin BB(E, 0)$ & $[$*self defense: A36.(1)*$]_{JCL}(B, Incident_1)$ $\notin BB(E, 0)$. When judge E makes a final judgment, E follows the *principle of innocent until proven guilty*. When the trial ends after k stages, the belief state of E in k is important. If E is persuaded that B killed A in the sense of JCL ($[kill]_{JCL}(B, A, Incident_1) \in BB(E, k)$), E should decide to punish B. Otherwise, E should declare that B is not guilty. As you can see, this procedure is dynamic. During a trial, the epistemic and the normative state of relevant persons change. The whole process in a court can be interpreted as a game with real consequences (Nakayama 2016). Indeed, a part of the (Japanese) Code of Criminal Procedure can be seen as a rulebook for plays in a court.

We interpret that the second premise of a legal syllogism is related with a classification problem. Because some legal terms contain ambiguity, there are often some classification possibilities for an incident. In a court, a prosecutor and a lawyer sometimes dispute about which interpretation is more appropriate. Our approach reflects this fact more faithfully than many approaches with non-monotonic reasoning.

7. Concluding Remarks

In this paper, based on LNS and DNL, we have analyzed the legal reasoning. Our analysis has shown that legal practitioners need a consistent interpretation of a legal code in order to justify their proposals. For this purpose, some terms in a legal code must be interpreted more restricted than usual. This interpretive task requires a holistic interpretation of a legal code. Another conclusion from our analysis is a certain applicability of the legal syllogism. This fact is based on the structure of legal codes. Many articles in a legal code has a form 'If $\phi(x)$, then $Should[\exists s\ punish(x, s)]$'. In Section 4 and 5, we have shown that LNS can formally express this kind of normative sentences. In Section 6, we have shown that DNL can be used to describe changes of normative states of a prosecutor, a lawyer, and a judge. We have suggested that the whole process of a trial can be interpreted as a game and such a game is describable in DNL.

In this paper, we could not explicitly show what kind of results our approach provides fro researches in Juris Informatics. This problem remains our future task.[6]

References

1. Davidson, D. (2001a) *Essays on Actions and Events*, 2nd ed., Clarendon Press.
2. Davidson, D. (2001b) *Inquiries into Truth and Interpretation*, 2nd ed., Clarendon Press.
3. Dickson, J. (2014) "Interpretation and Coherence in Legal Reasoning," *The Stanford Encyclopedia of Philosophy* (Summer 2014 Edition), Edward N. Zalta (ed.), URL = ⟨http://plato.stanford.edu/archives/sum2014/entries/legal-reas-interpret/⟩.
4. Douven, I. (2011) "Abduction", *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition), E. N. Zalta (ed.), URL = ⟨http://plato.stanford.edu/archives/spr2011/entries/abduction/⟩.
5. Dworkin, R. (1986) *Law's Empire*, Fontana Press.
6. Joseph, M. A. (2016) "Davidson: Philosophy of Language, " *The Internet Encyclopedia of Philosophy*, ISSN 2161-0002, http://www.iep.utm.edu/
7. MacCormick N. (2009) *Rhetoric and the Rule of Law: A Theory of Legal Reasoning*, Oxford University Press.
8. Nakayama, Y. (2011) *Norms and Games: An Introduction to the Philosophy of Society*, Keiso shobo. (In Japanese)
9. Nakayama, Y. (2013) "Dynamic Normative Logic and Information Update," T. Yamada (ed.) *SOCREAL 2013: 3rd International Workshop on Philosophy and Ethics of Social Reality*, Abstracts, Hokkaido University, Sapporo, JAPAN, pp. 23-27.
10. Nakayama, Y. (2014) "Speech Acts, Normative Systems, and Local Information Update," In: Y. I. Nakano, et al. (eds.) *New Frontiers in Artificial Intelligence (JSAI-isAI 2013 Workshops, Kanagawa, Japan, Selected Papers from LENLS10, JURISIN2013, MiMI2013, AAA2013, DDS13)*, LNAI 8417, Springer, pp. 98-114.
11. Nakayama, Y. (2015a) "Formal Analysis of Epistemic Modalities and Conditionals based on Logic of Belief Structures," In: T. Murata, K. Mineshima, D. Bekki, (eds.) *New Frontiers in Artificial Intelligence: JSAI-isAI 2014 Workshops, LENLS, JURISIN, and GABA*, Kanagawa, Japan, October 27-28, 2014, Revised Selected Papers (Lecture Notes in Computer Science Vol.9067), Springer, pp. 37-52.
12. Nakayama, Y. (2015b) "Logic of Modalities and Updates of Propositional Attitudes," *Proceedings of the Twelfth International Workshop of Logic and Engineering of Natural Language Semantics 12 (LENLS 12)*, pp. 173-186. 978-4-915905-68-1 C3004(JSAI):LENLS, USB Memory.
13. Nakayama, Y. (2016) "Chapter 12 Norms and Games as Integrating Components of Social Organizations," In: H. Ishiguro, et al. (eds.) *Cognitive Neuroscience Robotics B*,Springer, pp. 253-271.
14. van Benthem, J. (2011) *Logical Dynamics of Information and Interaction*, Cambridge University Press.
15. Wittgenstein, L. (1953) *Philosophische Untersuchungen.*

[6] This research was supported by Grant-in-for Scientific Research, Scientific Research C (24520014) : The Construction of Philosophy of Science based on the Theory of Multiple Languages. Finally, I would like to thank two reviewers for useful comments.

A Framework to Reason about the Legal Compliance of Security Standards*

Cesare Bartolini[1], Andra Giurgiu[1], Gabriele Lenzini[1], and Livio Robaldo[1]

University of Luxembourg,
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
{*firstname.lastname*}@uni.lu

Abstract. Achieving compliance with legal regulations is no easy task. Normally, laws state general requirements but do not provide clear parameters to determine when such requirements are met. On a different level, industrial standards and best practices define specific objectives that can be certified by means of auditing procedures from qualified bodies. Implementing a standard does not *per se* guarantee legal compliance, with the rare exception when the standard is also endorsed by the law itself. But standards and laws in the same domain may have overlaps and correlations, so adopting the former may provide an argument to demonstrate that adequate measures were taken to achieve legal compliance. In this paper, we introduce a framework that, using state-of-the-art Natural Language Semantics techniques, helps process legal documents and standards to build a knowledge base to store their logic representations, and the correlations between them. The knowledge base will help legal experts assess what requirements of the law are met by the standard and, consequently, recognize what requirements still need to be implemented to fill the remaining gaps. An application of the framework is exemplified by comparing a provision of the European General Data Protection Regulation against the ISO/IEC 27001:2013 standard.

Keywords: Legal compliance; legal requirements; security standards; General Data Protection Regulation.

1 Introduction

Security standards have contributed to shaping the quality of services and have promoted a security-oriented culture by establishing best practices. Like standards in general, they can also create in favour of the implementing party a "*presumption of conformity with the specific legal provision they address*" [12].

In themselves, standards do not guarantee legal compliance. In order to have a direct effect on legal compliance, standards would need to be endorsed by governments as key elements in their framework for policies and regulations. This is

* Livio Robaldo has received funding from the EU's H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 661007. Cesare Bartolini has received funding from the EU's H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974.

a solution that ISO and IEC are advocating in specific sectors, such as that Medical Device[1], but it would have severe limitations if applied at an international level. In the European Union, for instance, where there is the need to promote a single market while letting each country establish its own legal framework, supporting a standard in a legislative source (Regulation or Directive alike) would immediately exclude from conformity the countries where that standard is not commonly adopted, weakening their position in the single market [10]. This is why the European Union prefers to emit a legislation that defines an abstract level of safety/security of products and systems, and makes the adoption of standards to demonstrate compliance a voluntary choice[2].

That stated, standards nevertheless remain a viable way to support arguments for conformity with regulations. When widely adopted and subject to repeated audits by conformity-assessing bodies, a standard can give an organization that implements it an argument of compliance. Of course, such an argument is a presumption, giving the possibility to demonstrate that the organization failed to comply with the legal framework [12]. Still, this can offer an inversion of the burden of proof that the organization would not benefit from, if it simply adopted a personalized solution.

However, that argument of compliance needs to be reassessed when new laws reshape the legal landscape. In this case, a company needs to know whether or not the standards it has adopted will preserve the company's argument of conformity. Failing that, it may want to know what changes must be implemented to keep preserving that argument. Waiting for the adoption of new standards would take a significant amount of time and may lead to interim problems. In the meanwhile, businesses face the risk of liability for not being compliant and may, on that ground, be sanctioned. A passive business strategy like this may have the benefit of being effortless but can backfire with high costs if an incident occurs.

A safer strategy would be to identify what provisions in a standard interpret or implement the provisions in the law. This allows to determine what gaps need to be filled in order to benefit from the argument of compliance. Such *correlations* can express either a *formal compliance* based on the semantic of terms of the language in the standard and the law, or a *substantive compliance* based on decisions of courts and other competent authorities. This strategy has drawbacks as well: security standards, laws, and court decisions are written in natural language. Reading and analysing them by hand is inefficient and largely impractical. But this activity can be supported by computer processing.

We advocate such a pathway and propose a software framework that aids in determining the formal and substantive correlations between provisions in a standard and in a law. The framework's core is a logic-based methodology to represent, in a machine-processable format, (a) the relevant syntactical concepts in the provisions, and (b) the relevant correlations between them. In this paper,

[1] See for example `http://www.iso.org/sites/policy/sectorial_examples.html`.

[2] Regulation (EU) No 1025/2012 of the European Parliament and of the Council of 25 October 2012 on European Standardization, Article 2.1

we describe in detail this logic-based methodology, and exemplify how it works using an excerpt from the ISO/IEC 27001 security standard and a provision in the new European General Data Protection Regulation (GDPR)[3].

The framework depends on two auxiliary functional blocks (see Section 2):

1. a *logic knowledge base* that stores the history of the machine-processable logic correlations, based on collaborative work; experts can not only use the knowledge base but also correct and extend it;
2. a *set of Natural Language Semantics (NLS) and Natural Language Processing (NLP) techniques* that allow a user to browse a XML representation of the documents, search and retrieve the words, terms, and phrases that he and others have found relevant for correlation.

The NLS and NLP techniques will help users show the established correlations within the knowledge base. Any expert user can contribute to reinforce, correct, justify, and expand the correlations. Due to differences in legal interpretation, it may happen that some of the correlations will be in contradiction. Initially, these will remain as contradictions within the knowledge base; over time however they will be overridden by interpretations from more authoritative sources (such as those given by courts of a higher instance).

Technically, the different correlations are expressed in a deontic and defeasible logic for legal semantics called Reified Input/Output Logic (see Section 3), while the *modus operandi* of the framework is similar to that of the collaborative tools for document translation "SDL Trados Studio". It must be stressed that selecting the relevant terms and establishing the correlations is an activity that still requires human reading, processing and decision-making. The analysis of documents is therefore semi-automatic. It will be supported by the extensible knowledge base; the process of browsing, aided by the tools and techniques from the NLS and NLP, adds efficiency and precision.

The fast pace of technological innovation calls for new regulation. The European Union (EU)'s GDPR is the perfect example of a new legislation which is trying to address the challenges posed by new technologies. It will be applied from 25 May 2018[4], and it poses significant challenges for undertakings in terms of ensuring compliance with it, as it brings significant changes to the regulatory framework for personal data protection in Europe. Security standards, on the other hand, can be regarded as a building block [12] that helps data-processing organizations comply with the principle of accountability and with their obligations resulting from data protection laws. Considering the novelties of the GDPR, as well as the partial overlap between security and data protection, this article will illustrate a methodology based on the correlations between provisions of the GDPR and security standards, specifically ISO/IEC 27001.

[3] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[4] GDPR, Article 99.2.

2 The framework at a glance

The framework we propose is schematically summarized in Figure 1. Users (who may be lawyers, regulators, auditors, or other legal experts), access a digital and annotated XML representation of the normative texts (both laws and standards). While browsing a document and selecting the relevant concepts—some which may have been previously annotated—NLP and NLS tools help traverse the rest of the documents, find related terms, and recall the correlations that already exist between them. Not all these correlations have the same degree of importance, as they come from different sources and may have different relevance in specific contexts. This subsidiary information will be stored in appropriate metadata. Potential contradictions and ambiguities among the different interpretations can be solved, as detailed in Section 4. Besides, the user can find new terms and define new correlations, or correct existing ones. The framework thus implements a collaborative, self-correcting, strategy to evaluate the stored correlations. The user's decisions are stored in the knowledge base for later use, after having been appropriately represented in a logic for legal semantics. The details of this transformation are also explained in Section 4.

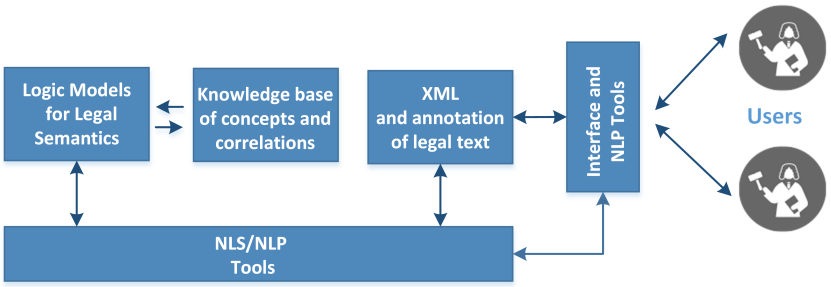


Fig. 1. The framework at a glance.

Our framework offers a computer-aided methodology to anyone who intends to analyze standards to build an argument of compliance with respect to a specific piece of legal text. The framework, and in particular its knowledge base, does not pretend to be complete. Rather, it provides the expert user with the updated knowledge that helps him or her take autonomous and informed decisions, either when confirming the correlations the tool suggests or when choosing to define new correlations.

The knowledge base is designed to tolerate apparent inconsistencies of different interpretations of terms which may lead to different correlations and create conflicts. Conflicts are especially frequent in legal interpretation, but can be solved, at least in principle, considering that the interpretations of higher instances, such as the Court of Justice of the European Union (CJEU), prevail over others. In order to cope with interpretations of different legal weights, which

may supersede one another, the logic formalism that the framework embeds is defeasible: correlations can be updated, modified, rewritten and weighted. If, despite the defeasible mechanism, conflicts do remain, the framework still embeds known strategies that help the user take a decision (see Section 4).

In the following, this paper will exemplify the core part of the methodology, which relates to building correlations.

3 Background

The background is structured along two main topics. 1) Natural Language Processing (NLP) [22,8] and Natural Language Semantics (NLS) [15,20]) in legal analysis, techniques which we invoke in the computer-assisted steps of our methodology. 2) Reified Input/Output Logic, the defeasible logic that we propose as the formal language to express correlations.

Natural Languages Processing in Law. The application of NLP and NLS to the legal domain is a research trend that has received a lot of attention and investments in recent years, as demonstrated by the several EU funded projects on the topic such as `Openlaws`[5], `Legivoc`[6], `EUCases`[7], `ProLeMAS`[8], `BO-ECLI`[9], and `MIREL`[10]. Modern NLP technologies [5] are able to classify and discover interlinks between legal documents thanks to parsers [2], statistical algorithms [7], and legal terminological databases or legal ontologies [23,7]. This is often done by transforming the legal documents into XML standards, such as *Akoma Ntoso*[11], where relevant information are tagged. An example of commercial legal document management system employing these technologies is *Eurocases*[12] which collects EU case law and uses NLP techniques to classify the documents on the basis of their topic [6].

Although these systems help navigate legislative documents and retrieve information, their overall usefulness is limited because they process words disregarding their possible different semantic interpretations. The latter would allow for legal reasoning, *e.g.*, correlating laws among them and determining whether they lead to inconsistencies. Semantic processing of documents like laws/regulations and security standards is what we are going to propose as a component of the methodology we present in Section 4.

Reified Input/Output Logic. This logic has been designed as an attempt to go further in state-of-the-art research in legal informatics, by investigating the *logical architecture* of the provisions, which are available in natural language only.

[5] `https://info.openlaws.com/openlaws-eu/`.
[6] `http://www.legivoc.eu`.
[7] `http://eucases.eu/start.html`.
[8] `http://www.liviorobaldo.com/Prolemas.htm`.
[9] `http://www.bo-ecli.eu/`.
[10] `http://www.mirelproject.eu/`.
[11] `http://www.akomantoso.org/`.
[12] `http://eurocases.eu/`.

Reified Input/Output logic is a logical framework [21] for representing the meaning of norms. Contrary to the great majority of its competitors [14,16,13], Reified Input/Output logic integrates modern insights from the NLS literature. Specifically, it merges Input/Output logic [17], a well-known formalism in Deontic Logic (*i.e.*, a logic that expresses concepts like permissions, obligations, prohibitions), with the first-order logic for NLS proposed by prof. J.R. Hobbs, which is grounded on the concept of *reification*. Reification is a concept originally introduced by the philosopher D. Davidson in [11]. It allows to move from standard notation in first-order logic such as "$(give\, a\, b\, c)$", asserting that "a" gives "b" to "c", to another notation in first-order logic "$(give\, e\, a\, b\, c)$", where "e" is the *reification* of the giving action. "e" is a first-order logic term denoting the giving event by "a" of "b" to "c". Thanks to reification, Hobbs's logic is able to express a wide range of phenomena in NLS such as named entities, quantification, anaphora, causality, modality, time, and more[13]. In particular, the simplified version of Reified Input/Output logic [21] that we use in our example is an extension of first-order logic that distinguishes three kinds of implication: "$\rightarrow$", "$\rightsquigarrow$", and "$\Rightarrow$".

The implication "$\rightarrow$" is the standard trust-value implication of first-order logic. The second, "$\rightsquigarrow$", is its *defeasible* version. Defeasible [19] here is to be understood in the sense that an implication "$\Phi \rightsquigarrow \Psi$" holds by default unless overridden by "stronger" implications. When instantiated properly, this notion of "stronger" implications resolves the potential contradictions emerging because of the non-monotonic nature of the defeasible reasoning. Reified Input/Output logic also includes other mechanisms to deal with unresolvable conflicts, such as the well-known conundrum about whether or not Nixon is a pacifist, considering that he is both a Quaker (therefore pro-peace) and a Republican (pro-war)[14]. A possible solution to deal with this type of conflicts is to leave the conflict open until more evidence will help the reasoner take a decision. This is, actually, what better fits a situation with conflicts due to multiple legal interpretations. The third implication, "$\Rightarrow$", is a deontic implication. Taken a formula Φ, referring to a set of pre-conditions, and another formula Ψ, referring to a set of actions, the meaning of "$\Phi \Rightarrow \Psi$" is "given the pre-condition Φ, actions Ψ are obligatory", that is, the actions must be undertaken if the pre-conditions hold. Note that, according to the interpretation of "$\Rightarrow$", the formula "$\Phi \Rightarrow \Psi \wedge \Phi \wedge \neg\Psi$" is not inconsistent: it only means obligation Ψ has been violated.

The framework [21] further distinguishes between the formulæ belonging to the assertive contextual statements (ABox), *i.e.*, the formulæ denoted by the norms, from those belonging to the terminological declarative statements (TBox), *i.e.*, the definitions, axioms, and constraints on the predicates used in the ABox formulæ. Formulæ in the ABox are in the form "$\forall_{x_1} \ldots \forall x_n\, [\Phi(x_1 \ldots x_n) \Rightarrow \Psi(x_1 \ldots x_n)\,]$", where arguments "$\Phi(x_1 \ldots x_n)$" and "$\Psi(x_1 \ldots x_n)$" are conjunc-

[13] More information are available on line at the following URLs:
`www.isi.edu/~hobbs/csk.html`
`www.isi.edu/~hobbs/csknowledge-references/csknowledge-references.html`.

[14] See `plato.stanford.edu/entries/logic-nonmonotonic/#sec-2-2`.

tions in standard first-order logic. Formulæ in the TBox can be any formula in standard first-order logic augmented with the operator "$\rightsquigarrow$".

4 Methodology

Our methodology, after minor adaptations, can be applied to find matches between any laws and any standards. However, in order to explain it in a manner as precise and specific as possible, we will refer only to a security standard in ISO/IEC 27000 family, more specifically ISO/IEC 27001, and to the GDPR. The application example in Section 5 will also refer to these two documents.

4.1 Overview

The methodology we propose is highly interdisciplinary, in that it involves a strict interaction between law and computer science. When implemented, it will partition the provisions of the the law into three sets:

set of covered provisions: these are the provisions that have a full match with the provisions in the standard. A match, or correlation, is expressed in terms of a logic implication, one of the three implications that the Reified Input/Output logic admits (for an example seeSection 5);

set of partially-covered provisions: the provisions in this set have a partial match with provisions in the standard, that is, not all the requirements of the provision have corresponding elements in the standard. To guarantee compliance with these provisions, the (certified) compliance with the standard is helpful but not sufficient, and additional requirements must be addressed;

set of uncovered provisions: these are the provisions for which there does not appear to be any correspondence in the standard. Concerning compliance with these provision, a certified compliance with the standard does not provide any useful information.

The types of coverage stated above can be further divided into two different categories: *formal* and *substantive* correlations.

Formal correlations entail a mere textual overlap between concepts. For example, formal correlations would allow us to observe that both the GDPR and the ISO/IEC 27001 standard use the term "transfer" (*e.g.*, GDPR, Article 46 and ISO/IEC 27001, Article. 13.2.1, shown in Section 5).

On the other hand, substantive correlations are more complex and entail the analysis of the actual meaning of terms. To assert a correlation of this kind, requirements must be met in a concrete way. In other words, and following the previous example, to assert a correlation between a provision of the GDPR and one of the standard concerning transfer, it is necessary to verify the actual difference/similarity in meaning.

It should be pointed out that the correlations that define a match are *defeasible*, *i.e.*, they can be superseded by new correlations. A superseding correlation

might be the consequence of a decision of a court or Data Protection Authority (DPA), or simply the evolution of the interpretation in doctrinal analysis. This way, the output of the methodology can be easily kept up to date by introducing the new correlations as they are developed by the legal actors.

4.2 Building Correlations

To build the correlations and define the types of coverage, the methodology follows three steps, which involve both a legal and a technical approach. The legal approach is focused on the interpretation of the provisions of laws (the GDPR in our example) and security standards, whereas the technical approach consists of modelling those provisions into an ontology, and expressing the interpretation by means of logical formulas. The sequence of steps is meant to be iteratively repeated. At each iteration, existing correlations between provisions in the law and provisions in the standard are presented to the user, but new correlations may be established and existing correlations may be updated. The knowledge base of correlations is thus used and updated at the same time.

The next paragraphs detail the steps.

Step 1: analysis of the provisions. The provisions of the law and of the security standard are analyzed by legal experts. They provide a legal interpretation of the terms used in the provisions and compare them in search of semantic correlation. There is no need for this interpretation to be final—more interpretations can be added later, and old interpretations can be overridden by newer ones—but this start requires a significant manual activity. The analysis also builds a structured model of the provisions, using an annotation tool called LIME[15]. LIME does not modify the text of the law, but splits it into its components, *i.e.*, articles, paragraphs, and so on. This annotation converts the law and the security standard into the XML legal format `Akoma Ntoso`. The latter is a formalism that provides specific XML tags for linking the textual spans with separate semantic annotations. So, for example, relevant concepts can be embraced by `<concept eId=`x`>` ... `</concept>`, where "x" indexes an ontological concept (see next step).

To support the execution of this step (and especially its successive iterations), we link LIME to external NLP procedures so that it suggests (semi-automatically and while the legal expert browses the documents) previous translations and correlations on the basis of the ones currently stored in the knowledge base. Ultimately, it is the legal expert that must decide whether and how the correlations shown need to be overridden. In that case, together with the updating of the knowledge base, we can annotate the new correlations with the source which contributed to define it (*e.g.*, Court of Justice of the European Union, *Dapreco and Copreda Corp.*, C-XYZ/16). Correlations by more authoritative sources must override defeasible correlations by less authoritative ones, reshaping the partitions in the three coverage sets as described in Subsection 4.1.

[15] `http://lime.cirsfid.unibo.it/`.

Step 2: creation of legal ontologies. The legal interpretations are mapped onto legal ontologies of the law and the security standard. Legal ontologies [4] model the legal concepts, the parties and stakeholders affected by the law, the duties and rights of each stakeholder, and the sanctions and penalties for violating the duties. As per ontologies in general, legal ontologies must be expressed in a knowledge representation language. For this work, we chose the popular abstract language OWL [1]. It can be serialized using various XML notations, making it well fit for machine processing. For example, the OWL representation of the data protection ontology will contain concepts such as "controller", "data subject", "personal data", "processing" and so on.

Concerning the GDPR used in our example, a preliminary version of a legal ontology has been defined already [3]. Albeit partial and based on an older text of the GDPR, it was designed to express the duties of the controller. As such, it can be used to find the correspondences between the requirements expressed in the GDPR and in security standards, until an improved ontology is built.

Step 3: generation of logic formulæ. The third and final step of the methodology consists of generating the logical formulæ representing the set of provisions in the law and the set of provisions in the security standard, as well as the correlations between them. These formulæ are expressed in Reified Input/Output logic [21], the first-order language illustrated above in Section 3.

Associating textual provisions to logical formulæ amounts to converting ambiguous and vague terms into non-ambiguous items (predicates and terms). Also, as extensively explained above, these predicates, as well as the correlations connecting them, are subject to different legal interpretations. To this end, words in the provisions are represented via predicates reflecting their ambiguity/vagueness. For example, the word "appropriate", included in the sample GDPR provision used below in Section 5, will be represented via the homonym predicate "appropriate". These "vague" predicates may be defined by adding correlations and further constraints (axioms). Those correlations will be *defeasible*, so that they can account for different legal interpretations.

5 Generation of Logic Formulæ: example

We exemplify step 3 of our methodology by using a provision from the GDPR and an article of the ISO/IEC 27001 security standard. Namely:

(a) GDPR, Article 46.1: *[. . .] a controller or processor may transfer personal data to a third country or an international organization only if the controller or processor has provided appropriate safeguards [. . .]*

(b) ISO 27001, Article. 13.2.1: *Formal transfer policies, procedures and controls shall be in place to protect the transfer of information through the use of all types of communication facilities.*

We underline that this example is based only on merely formal correlations between the two texts and relies on the formal conceptual identity of the term

"transfer". In this paper we did not take into account the substantive interpretations, according to which the same term would be used with different meanings in the two contexts. The example has the sole purpose of exposing how formulæ are build; it does not mean to assess the substantive correlation between the two texts, which might lead to a different conclusion in terms of compliance.

The formalization of the two provisions in the simplified version of Reified Input/Output logic we use in this paper is rather straightforward. As explained above in Section 4.2, the formulæ will include predicates reflecting the vagueness of the terms occurring in the sentences. Thus, for instance, the main verb "transfer" is formalized into an homonym predicate "transfer" and the adjective "appropriate" is formalized into an homonym predicate "appropriate".

Note that the sentences may refer to *tacit knowledge*, *i.e.*, they may contain ellipses. For instance, the GDPR uses the term "data subject" to generically refer to individuals to which the personal data pertains. However, in many GDPR provisions, such as the one shown above, it is not specified that the "personal data" are the "personal data of a data subject". Tacit knowledge ought to be restored in the formula; thus we will represent "personal data" as "$(\mathsf{personalData}\, y) \wedge (\mathsf{of}\, y\, z) \wedge (\mathsf{dataSubject}\, z)$", where "$y$" is a (generic) unit of personal data and "z" is the (generic) data subject that owns "y".

Elliptical/tacit knowledge is only one of the issues we must deal with when translating natural language into logical formulæ. In this paper, we cannot illustrate all major problems in NLS and how it is possible to address them in a reification-based framework such as the one of prof. Hobbs.

The GDPR provision in (a) is formalized as follows:

$$\begin{aligned}&\forall_e \forall_x \forall_y \forall_z \forall_w \forall_k [\\ &\quad ((\mathsf{dataController}\, x) \wedge (\mathsf{personalData}\, y) \wedge (\mathsf{of}\, y\, z) \wedge (\mathsf{dataSubject}\, z) \wedge \\ &\quad (\mathsf{thirdCountryOrIntOrg}\, w\, z) \wedge (\mathsf{transfer}\, e\, x\, y\, w) \wedge (\mathsf{instrOf}\, e\, k)) \\ &\quad \Rightarrow (\mathsf{appropriate}\, k)] \end{aligned} \tag{1}$$

In (1), "e" is a first-order variable referring to the event of transfer of "y" (patient) performed by "x" (agent) to "w" (receiver). "x" has to be a data controller[16] while "y" refers to personal data of a data subject "z". "k" is the instrument of the action "e", *i.e.*, the communication facility through which data are transferred. The predicate "thirdCountryOrIntOrg" is a convenient predicate to state that its first argument is either an international organization or a country different from the one where the data subject lives. This is enforced by adding the following definition of "thirdCountryOrIntOrg" to the TBox:

$$\begin{aligned}&\forall_w \forall_y \forall_c [(\mathsf{thirdCountryOrIntOrg}\, w\, z) \leftrightarrow \\ &\quad ((\mathsf{IntOrg}\, w) \vee ((\mathsf{country}\, w) \wedge (\mathsf{diffFrom}\, w\, c) \wedge (\mathsf{countryOf}\, c\, z)))] \end{aligned} \tag{2}$$

[16] Or a data processor, but in this example we assume for simplicity that the provision only refers to data controllers.

The crucial predicate in (1) is the predicate "appropriate", which may be true or false with respect to a communication facility "k".

The TBox of the ontology will include defeasible axioms that define when a communication facility is "appropriate". For instance, an undertaking certified to ISO 27001 has adopted a formal transfer policy according to which data transfers must take place via email with electronic signature or registered (hard) mail. Assuming that such data transfers are to be considered appropriate, the following two (defeasible) axioms are added to the knowledge base.

$$\begin{aligned} &\forall_x[\,(\mathsf{emailWithES}\,x) \;\rightsquigarrow\; (\mathsf{appropriate}\,x)\,],\\ &\forall_x[\,(\mathsf{regMail}\,x) \;\rightsquigarrow\; (\mathsf{appropriate}\,x)\,] \end{aligned} \tag{3}$$

In case a judicial authority decides, for example, that emails are no longer appropriate means, an additional higher-priority axiom will be added to the TBox in order to override the first axiom in (3). However, axioms may be associated to time stamps (although this is not shown in (3), so that the new axiom will override the old one only for transfer actions performed after a certain date, *i.e.*, the one from which the new law enters into force. On the other hand, the old axiom will still assert that email with electronic signature is an "appropriate" communication facility for all transfer actions performed before that date.

Formula (4) models the ISO/IEC 27001 provision in (b). This provision refers to any kind of information, not only personal data. It does not specify the agent or the receiver of the transfer actions, so those are assumed to be any entity. The formal transfer policies, procedures and controls to protect the transfer of information are simply formalized in terms of a predicate *secure*, which is true for any communication facility that is deemed to be "secure".

$$\begin{aligned} &\forall_e\forall_x\forall_y\forall_w\forall_k[\\ &\quad(\,(\mathsf{transfer}\,e\,x\,y\,w) \;\wedge\; (\mathsf{information}\,y) \;\wedge\; (\mathsf{instrOf}\,e\,k)\,)\\ &\quad\Rightarrow\; (\mathsf{secure}\,k)\,] \end{aligned} \tag{4}$$

Since personal data are a particular subclass of information, the TBox includes the following axiom:

$$\forall_y[\,(\mathsf{personalData}\,y) \;\rightarrow\; (\mathsf{information}\,y)\,] \tag{5}$$

On the other hand, since in (4) the agent and the receiver may be any entity, the formula may be obviously evaluated also when these are data controllers and third countries or international organizations.

So far, the example serves the purpose of illustrating how formal correlations work. To take it one step further we would have to assume a substantive correlation between the terms in the two norms. This would mean that *transfer* is used in the two text with the same meaning. Under this hypothesis, if there is a

legal interpretation stating that whenever a means of communication is "secure" with respect to the ISO/IEC 27001 standard, then it is "appropriate" for transferring personal data with respect to the GDPR, such an interpretation can be expressed as shown in (6).

$$\forall_k [\, (\mathsf{secure}\, k) \;\rightsquigarrow\; (\mathsf{appropriate}\, k) \,] \tag{6}$$

Based on such an interpretation, it would then be possible to automatically infer that, whenever the ISO/IEC 27001 standard is respected by an agent "x", so is (part of) the corresponding GDPR's provision.

Again, the correlation formalized in axiom (6) is defeasible. In case a court or a data protection authority later decides that not all secure communication facilities are appropriate from the point of view of provision (a), it is possible to add further higher-priority axioms to the knowledge base in order to override (6), thus keeping track of that specific legal interpretation.

6 Discussion and Conclusion

This paper introduces a semi-automated methodology to reuse existing security standards to help ensure legal compliance. By following the illustrated methodology, one can build a machine-processable *knowledge base* of logic formulæ that model relevant concepts from the text of a law and a security standard, together with their possible correlations. Once the knowledge base has been populated with a sufficient number of correlations, since standards can be certified by auditors, an enterprise that implements the standard can have an argument for compliance, at least with the provisions to which the standard correlates.

The knowledge base will thus allow to detect the legal provisions covered by the standard, and can then be used to assess to what extent the enterprise that has implemented the standard(s) complies with the law.

Although this paper focuses on the methodology only, several technical challenges related to building and updating the knowledge base are raised.

First off, the translations from natural language to logical formulæ must be uniform for excerpts of text that are similar to each other. To achieve this, we must overcome the limitation of a manual translation, which would be time-consuming and error-prone. For this reason, our work must rely on current NLP technologies. However, even at the best of their performances, NLP algorithms are still unable to automatically carry out the translation with a reasonable level of accuracy, so we advocate a *semi-automatic* translation of the provisions.

Similar approaches are applied to translations in general. Here, translators are helped by collaborative tools such as the "SDL Trados Studio"[17], which suggests, via pattern-recognition text-similarity NLP techniques [18,9], how to translate a sentence on the basis of the translations of similar sentences that the translators have previously stored in the tool. Using a feature used in tools

[17] `http://www.translationzone.com/products/trados-studio/`.

to learn foreign languages such as Duolingo[18], some translations acquire more importance the more they are chosen by highly-qualified end users.

Inspired by that approach, we will extend the LIME text editor to assist the manual translation of provisions into formulæ. For each provision, the editor will display the translations of similar provisions found via NLP procedures applied to the provisions already stored in the knowledge base.

Secondly, the knowledge base must be consistent, *i.e.*, without contradictions. To check for consistency, we plan to store formulæ in an XML-based data model, and employ/extend reasoners[19], to monitor the consistency of the knowledge base, whenever new formulæ are added to it.

The methodology herein is currently a work in progress and not fully implemented yet. The complete methodology requires the definition of a detailed taxonomy of concepts extracted from the law and the security standards.

In the future, we envision significant developments of the research presented in this paper. The knowledge base will be populated with *legal* interpretations that will be translated into defeasible formulæ. To this end, and along the GDPR example, web crawlers can be used to parse documents from relevant portals, *e.g.*, those of national DPAs in the EU, in order to check (via NLP) whether new interpretations of the GDPR provisions are available. The knowledge engineers can then update the knowledge base, overriding some defeasible implications occurring therein. This task can be more easily achieved if the the methodology and the techniques we have illustrated here are embedded in a web service. By accessing the service, users, such as legal experts, judicial courts and the like, will benefit from the collaborative expertise brought by the knowledge base; at the same time, they can help build and update the substantive correlations. In addition to the technical skills needed to manage the knowledge base, this step would greatly benefit from a close interaction with legal authorities. We envision that in a small country like Luxembourg that could be more easily achieved.

References

1. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, chap. 4, pp. 67–92. International Handbooks on Information Systems, Springer Berlin Heidelberg (2004)
2. Arora, C., Sabetzadeh, M., Briand, L.C., Zimmer, F.: Automated checking of conformance to requirements templates using natural language processing. IEEE Transactions on Software Engineering 41(10), 944–968 (2015)
3. Bartolini, C., Muthuri, R., Santos, C.: Using ontologies to model data protection requirements in workflows. In: Proceedings of the Ninth International Workshop on Juris-informatics (JURISIN). pp. 27–40 (November 2015), extended version to be published in LNAI book.
4. Benjamins, R., Selic, B., Gangemi, A. (eds.): Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications, Lecture Notes in Artificial Intelligence, vol. 3369. Springer-Verlag Berlin Heidelberg (2005)

[18] `https://www.duolingo.com/`.

[19] `https://www.w3.org/2001/sw/wiki/OWL/Implementations`.

5. Boella, G., Di Caro, L., Humphreys, L., Robaldo, L., Rossi, R., van der Torre, L.: Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artificial Intelligence and Law to appear (2016)
6. Boella, G., Di Caro, L., Graziadei, M., Cupi, L., Salaroglio, C.E., Humphreys, L., Konstantinov, H., Marko, K., Robaldo, L., Ruffini, C., Simov, K., Violato, A., Stroetmann, V.: Linking legal open data: Breaking the accessibility and language barrier in european legislation and case law. In: Proc. of the 15th International Conference on Artificial Intelligence and Law. ACM, New York, USA (2015)
7. Boella, G., Di Caro, L., Rispoli, D., Robaldo, L.: A system for classifying multi-label text into eurovoc. In: Proceedings of the 14th International Conference on Artificial Intelligence and Law. ICAIL '13, ACM, New York, USA (2013)
8. Boella, G., Di Caro, L., Robaldo, L.: Semantic Relation Extraction from Legislative Text Using Generalized Syntactic Dependencies and Support Vector Machines, pp. 218–225. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
9. Boella, G., Di Caro, L., Ruggeri, A., Robaldo, L.: Learning from syntax generalizations for automatic semantic annotation. The Journal of Intelligent Information Systems 43(2), 231–246 (2014)
10. Chalmers, D., Davies, G., G.Monti: European Union Law: Cases and Materials. Cambridge University Press (2008)
11. Davidson, D.: The logical form of action sentences. In: Rescher, N. (ed.) The Logic of Decision and Action. Univ. of Pittsburgh Press (1967)
12. De Hert, P., Papakonstantinou, V., Kamara, I.: The cloud computing standard ISO/IEC 27018 through the lens of the EU legislation on data protection. Computer Law & Security Review 32(1), 16–30 (February 2016)
13. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing strong and weak permissions in defeasible logic. Journal of Philosophical Logic 42(6), 799–829 (2013), http://dx.doi.org/10.1007/s10992-013-9295-1
14. Hansen, J.: Prioritized conditional imperatives: problems and a new proposal. Autonomous Agents and Multi-Agent Systems 17(1), 11–35 (2008)
15. Hobbs, J.R.: Deep lexical semantics. In: Proc. of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (2008)
16. Horty, J.: Reasons as Defaults. Oxford University Press (2012)
17. Makinson, D., van der Torre, L.W.N.: Input/output logics. Journal of Philosophical Logic 29(4), 383–408 (2000)
18. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. AAAI Press (2006)
19. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
20. Robaldo, L.: Interpretation and inference with maximal referential terms. The Journal of Computer and System Sciences 76(5), 373–388 (2010)
21. Robaldo, L., Humphreys, L., Sun, L., Cupi, L., Santos, C., Muthuri, R.: Combining input/output logic and reification for representing real-world obligations. In: Post-proceedings of the 9th International Workshop on Juris-informatics. LNCS (2016)
22. Robaldo, L., Caselli, T., Russo, I., Grella, M.: From italian text to timeml document via dependency parsing. In: Proc. of 12th International Conference 'Computational Linguistics and Intelligent Text Processing'. (2011)
23. Vibert, H., Jouvelot, P., Pin, B.: Legivoc - connectings laws in a changing world. Journal of Open Access to Law 1(1), 165–174 (2013)

A Method to Estimate Document Structure from Unstructured Documents

Yoichi Hatsutori, Katsumasa Yoshikawa, and Haruki Imai

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

Abstract. Text analytics is used to analyze diverse documents. For example, legal documents (such as contracts, ordinances, regulations, and global standards) must be analyzed for corporations to manage their business risk and meet compliance requirements. However, since documents are often stored or published as unstructured documents, they need to be preprocessed to analyze them in subsequent text analytics. In particular, the following two forms of preprocessing are useful for text analytics: (1) extracting text, and (2) estimating document structure (such as chapters, sections, and subsections), which is used to define the range of topics or articles in a document. This paper presents a preprocessing method to estimate document structure from unstructured documents. The proposed method is rule-based approach, and consists of three algorithms: (1) one is based on style information, such as bold font; (2) another is based on numbered objects, such as sections; and (3) the other is based on a document's Table of Contents, which summarizes the document's structure. The accuracy of the proposed method is also evaluated by using 102 documents. The proposed method was found to be able to estimate document structure with 96.6% accuracy.

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

1 Introduction

Electronic text documents (e.g. office documents, web contents, articles, and technical papers) are widely read and available online. The growth in available electronic text documents has been accompanied by active research on text analytics, like natural language processing (NLP) and text mining. Text analytics is important to analyze legal documents (such as contracts, ordinances, regulations, or global standards), for example, so that corporations can identify, evaluate, and manage their business risk and meet compliance requirements. However, since most documents are stored as unstructured documents, they need to be converted into a structured format for text analytics. Thus, processing unstructured and diverse documents raises challenges in the preprocessing phase of text analytics.

Key tasks in the preprocessing phase of text analytics are; (1) extracting text from unstructured documents, and (2) estimating document structure (chapters, sections, subsections, and paragraphs), which is used to define the range of topics in a document. For example, Mao et al. [1] mentioned that structural information can be very useful in indexing and retrieving information contained in a document. Pasetto et al. [2] pointed out that document structure needs to be obtained to be able to compare semantics in legal document analysis.

For extracting text, open source libraries can be used to extract text descriptions and style information from diverse documents. For example, Apache Tika[1] can extract text descriptions and style information from over 1,000 different file types. Therefore, we focus on the other task: estimating document structure.

To estimate document structure, estimation algorithms need to consider characteristics of unstructured and diverse documents. For example, unstructured documents do not have a common structure. In addition, the definition of document structure can differ even if documents are stored in the same file format because ways to define document structure vary in accordance with the document's objective, its writers, and so on. Thus, document structure should be estimated from contents of a document, e.g. text description or style information.

We have already proposed a method to estimate document structure from text description, which is the uniquely common information among diverse documents [3]. The method is rule-based and estimates document structure by searching for numbered objects like "Section 1." and "Section 2." or "(1)," "(2)," and "(3)" as shown in Fig. 1. This method is applicable when document structure is defined by numbered objects. However, while some documents use numbered objects to define document structure, others use style information like **bold** or *italic* font as shown in Fig. 2. When document structure is defined by style information, our previous approach cannot estimate document structure. In addition, we also consider the "Table of Contents," which outlines a document structure, as shown in Fig. 3, since it is another clue for estimating document structure.

The contribution of this paper is to expand the method described in the previous study [3] and present a novel preprocessing method considering style information, numbering information, and the Table of Contents. The proposed method is designed to preprocess unstructured and diverse documents and enables us to add structure information to them.

The rest of this paper is as follows. Section 2 introduces types of document structure definition. Section 3 summarizes related work on estimating document structure. Section 4 presents our new preprocessing method. Section 5 evaluates and discusses the accuracy of the proposed method. In Section 6, the proposed method is used to analyze some Terms of Use published on websites of web-service companies. Finally, Section 7 concludes this paper.

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

2 Definitions of Document Structure

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

In the first way, serial numbers are used for defining document structure. For example, numbered objects such as “1.” and “2.” or numbered objects with a prefix such as “Chapter 1” and “Chapter 2” are added before each chapter, section, or subsection. Fig. 1 shows an example in which additional text (i.e. “Section 1” and “Section 2”) is used for structural definition.

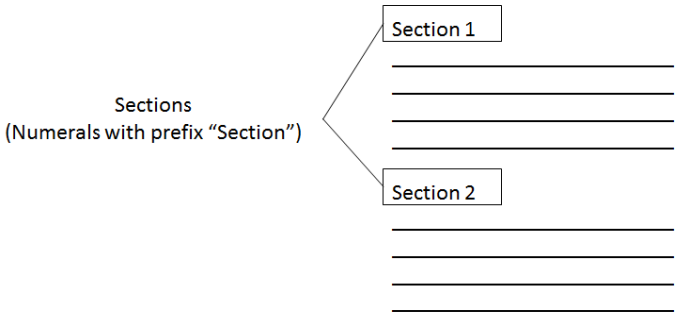


Fig. 1. Example of structural definition by additional text

Changing style is another way to define structure. For example, font size, font style (e.g. bold, and italics), alignment, indent, and so on can be used. Fig. 2 shows an example in which bold font is used to define the document structure.

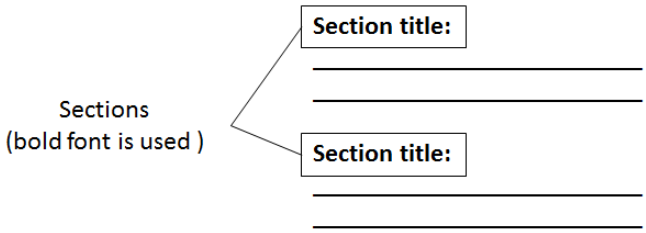


Fig. 2. Example of structural definition by style information

In addition, some documents have a Table of Contents, which outlines document structure. Since the structure shown in a Table of Contents is the same as that of the body text, this information is useful for estimating document structure. Fig. 3 shows an example in which “Article 1” and “Article 2” are defined in body text and “Table of Contents” shows the same structure.

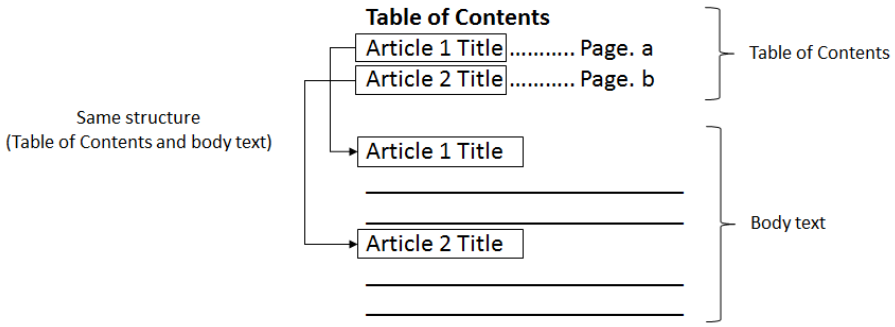


Fig. 3. Example of Table of Contents and body text

In this paper, both additional text and style information are considered to estimate document structure. Moreover, "Table of Contents" are also used for improve the results of the estimation. Details are described in Section 4.

3 Related Work

As described in Section 2, there are many ways to define document structures and algorithms to estimate them. In this section, related work is summarized and the difference of this paper is described. The first approach is based on text description. The second approach is based on additional information attached to text, e.g. font information or tag information describing structural information. The third approach is based on syntactic rules.

3.1 Text-based approach

Fukui et al. [4] proposed a text-description-based approach to extract document structure. In their research, pattern match is used for each line to extract document structure. Each line is assumed to conform to their knowledge-based model called the "Intra-line Structure Model," which consists of heading, delimiter, and content parts. They prepared about 300 patterns for the heading part, which consists of a numeric part, punctuation, and a reserved heading word. If a line has a heading part that is matched to a predefined pattern, then the line is extracted as a start point of document structure. The average accuracy of their structure extraction is 89.0%. However, their intra-line structure model assumes that only the heading part can be the start point of a document structure and that reserved heading words are required for detecting structure.

Déjean [5] proposed a robust method to detect long-range numbered sequences in different types of documents. He specifically focused on scanned, optical character recognition (OCR), and PDF documents. Only text description is used for estimating document structure. In Déjean's research, incremental types (e.g. digits, upper- and lower-case letters, Roman letters, Chinese characters, or "Bis" sequences) are detect-

ed and extracted by predefined regular expressions. The document is scanned line by line, and numbered objects are extracted by using pattern match. For every match, a pattern is associated with the line. A numbered object is recognized as a header part of the line, and the remaining part is recognized as the body part of the line. Document structure is estimated from extracted objects. The originality of this research is the notation of missing items (often due to OCR errors), and the notation of multi-increment items. This research also addresses intra-line structure, like that of Fukui et al.

3.2 Markup-based approach

Some documents have useful information for estimating document structure. For example, "heading" tags are available in Hyper Text Markup Language (HTML). If a document has meaningful tag information describing structural information, document structure can be extracted from the document. Some open source libraries are available for extracting document structure from markups. For example, Apache Tika can extract metadata and text. However, users can use not only heading tags but also other tags such as style information, i.e. bold font as shown in Fig. 2, for defining document structure. Therefore, even if a document contains markups, an algorithm to estimate document structure from style information is required for analyzing diverse documents.

3.3 Syntactic approach

Igari et al. [6] proposed an approach based on syntactic rules. They claim that legal judgments are described in a common format, often have structures that are highly similar, and often include the same type of information in similar parts [6]. When documents are described in a common format, the document structure can be expressed by using syntax rules for the document text. Because Igari et al. focus on analyzing and parsing legal judgments, which have a well-defined common format, their method has generality, extensibility, and high accuracy. This approach is very useful and highly accurate for documents that have a common format. However, we assume that diverse documents do not share a common format.

3.4 Scope of this paper

As described in Subsection 3.3, we focus on analyzing diverse documents. This means that there are no common structural formats in documents, so the syntactic approach is not considered but the text-based and markup-based approaches are. In particular, style information is considered in the markup-based approach.

In addition, there are two ways for extracting document structure: a machine learning approach and a rule-based approach. In the machine learning approach, a large amount of training data is required to train the model for estimating document structure. However, preparing training data is an obstacle for practical use. On the other hand, the rule-based approach does not need training data to be prepared. Therefore,

the rule-based approach is used for extracting document structure. Since this paper focuses on a method to estimate document structure, we assume that text is extracted from diverse documents preliminarily.

4 Proposed Method

This section describes a three-algorithms method to estimate document structure from unstructured documents. As mentioned in Section 2, there are two types of definitions of document structure: document structure is defined by additional text or style information. Since the proposed method considers these two definitions, the method includes numbering-based and style-based algorithms. In addition, since some documents have a Table of Contents describing document structure, the proposed method also utilizes this additional information to estimate the document structure. Subsection 4.1 details the style-based algorithm, Subsection 4.2 the numbering-based algorithm, and Subsection 4.3 the Table-of-Contents-based algorithm. Then the whole processing flow and some configuration parameters are described in Subsection 4.4.

4.1 Style-based algorithm

This subsection describes the style-based algorithm to estimate document structure that can be used when document structure is defined by using style information.

First, users define style information for each document structure. For example, relations such as bold for sections or italics for subsections are defined by users.

Second, text having the same style information is searched for in a document. When style is detected in the body text, the text is extracted as a starting point of each document structure. Note that searching text is processed line-by-line. When style information matches only a part of a line, the line is skipped.

Third, text belonging to each section is estimated. Basically, text at the starting point of a section is assumed to be text of that section. Similarly, text at the end of a document is assumed to be text of the last section. For subsections, a similar algorithm is used to estimate the range of subsections. Text at the starting point of a subsection is assumed to be a text of that subsection. Text at the end of the section is assumed to be text of the last subsection.

Figs. 4 and 5 show examples of style-based estimation. In these figures, bold is used for sections, and italics for subsections. These relationships are predefined by users. In Fig. 4, two instances of “Section title” in bold are extracted as a section-level structure. Next, text from the first starting point to the second is extracted as text of the first section. Text from the second starting point to the end of a document is extracted as text of the second section. In Fig. 5, four instances of “Subsection” in italics are extracted as a subsection-level structure. Text at the starting point or the end of the subsection is extracted as a text of the subsection.

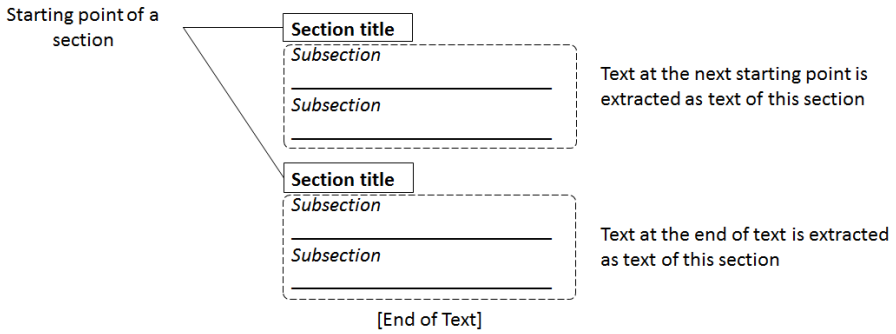


Fig. 4. Example of style-based estimation (section level)

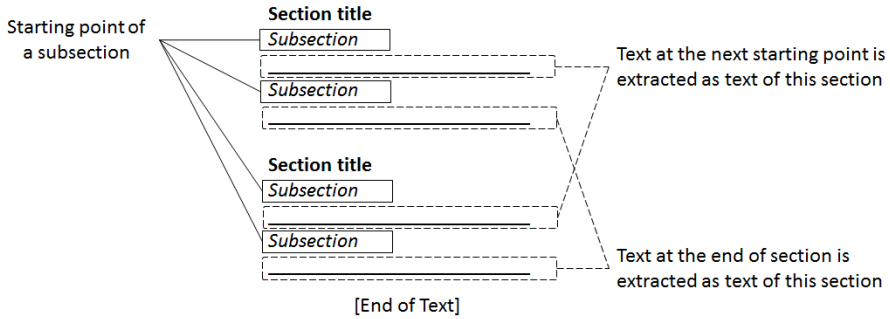


Fig. 5. Example of style-based estimation (subsection level)

4.2 Numbering-based algorithm [3]

This subsection briefly introduces a numbering-based algorithm to estimate document structure [3] that can be used when document structure is defined by numbered objects. First, regular expressions to search for numbered objects are predefined by users. Next, numbered objects matching regular expressions are searched for in a document. Tree structures are built by using all candidates of the searched objects, and a pruning technique is applied to them to remove unnecessary leaves.

An example is shown in Fig. 6. In the left image, numbered objects are extracted by predefined regular expressions: "\d\." and "\([a-z]\)." In the center image, extracted objects are classified and grouped by using the left-hand-side characters. Candidates of tree structures are built in each group and leaves of the tree are pruned. In this example, since all candidates are unbranched trees, the pruning technique is not needed. Then, the super-sub relationships among unbranched trees are estimated in the right image. Since (a) and (b) are covered by 1. and 2., we can estimate (a) and (b) to be a lower level structure than 1. and 2.

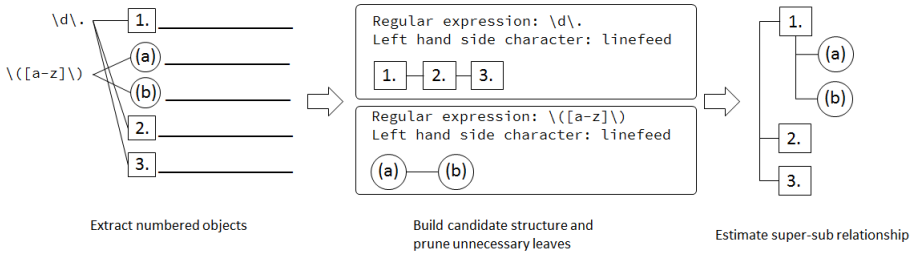


Fig. 6. Example of numbering-based estimation

4.3 Table-of-Contents-based algorithm

As mentioned in Section 2, some documents have a Table of Contents to outline document structure. Since the structure in a Table of Contents is the same as that of body text, document structure of body text can be estimated from the Table of Contents. The merits of using a Table of Contents are (1) document structure can be estimated even if document structure is not defined by using either style information or numbered objects and (2) the results estimated by style- and numbering-based algorithms can be verified. Therefore, this subsection explains an algorithm that uses a Table of Contents to estimate document structure. This algorithm consists of four steps.

In the first step, the following parameters are predefined to estimate a Table of Contents and its range: (1) key-words representing the starting point of a Table of Contents are defined, such as Table of Contents, Contents, and Summary; (2) regular expressions to detect each line contained in a Table of Contents, for example, "(Article\s\d)([a-zA-Z0-9\s]+)(\s\.+\sPage\.\s[ab])," are defined as in the case in Fig. 3; (3) style information described in Subsection 4.1 is defined to detect document structure; and (4) regular expressions described in Subsection 4.2 are defined to analyze numbered objects in a Table of Contents. Note that regular expressions defined in (2) consist of three parts. The first is the prefix text part. The text in this part can be used in the body text. In Fig. 3, "Article 1" and "Article 2" are used in the body text and the regular expression "(Article\s\d)" is the first part of the regular expression. The second is the main text part. This is mainly used as a title of each section. The third is the additional text part. The text in this part is mainly used in the Table of Contents only. For example, in Fig. 3, "……Page. a" and "…… Page. b" are the additional text part describing a page number and used only in the Table of Contents.

In the second step, the scope of the Table of Contents is estimated by using predefined parameters: (1) key-words and (2) regular expressions. The text matching the prefix and main text parts of regular expressions is extracted from the Table of Contents. In Fig. 3, for example, "Article 1 Title" and "Article 2 Title" are extracted from the Table of Contents.

In the third step, text in the Table of Contents is analyzed by using predefined parameters: (3) style information for document structure and (4) regular expressions for

numbered objects. As a result, document structure in the Table of Contents is estimated.

Finally, by using the text extracted in the second step, corresponding text in body text is searched for. For example, in Fig. 3, “Article 1 Title” and “Article 2 Title” are extracted in the second step and then, searched for in body text. Note that we empirically find that some documents use the same text in the Table of Contents and the body text while others do not, e.g. “Article 1 Title” appears in the Table of Contents but only “Title” appears in the body text. Therefore, first, we search for text containing a prefix. If we cannot find any such sections, the text without a prefix is searching to find corresponding text.

4.4 Processing flow and configuration parameters

In this subsection, processing flow of the whole process and necessary configuration parameters are described.

Since a Table of Contents is an outline of a document structure, the Table-of-Contents-based algorithm is applied to estimate the document structure first. If we cannot extract any document structure from a Table of Contents, e.g. a document does not have a Table of Contents, the style- and numbering-based are applied to estimate the document structure. Here, to select the style- or numbering-based algorithm, a configuration parameter is used.

Configuration parameters required to estimate document structure are listed in Table 1. These parameters should be predefined by users. In our implementation, a user can set configuration parameters for each document.

Table 1. Configuration parameters required to estimate document structure

Parameters	Description
Algorithm selection	Select style-based or numbering-based
Key words for "Table of Contents"	Search for the starting point of Table of Contents
Regular expressions	Search for lines in Table of Contents
Style information	Estimate document structure in Table of Contents
Regular expressions	Estimate document structure in Table of Contents
Style information	Estimate document structure in body text
Regular expressions	Estimate document structure in body text

5 Evaluation

In this section, the proposed three-algorithms method is evaluated using legal documents available online. First, a correct dataset, i.e. ground truth, is created manually. Next document structure is estimated by using the proposed method. Third, the result of document structure estimation is compared with the correct data set. Types of input legal documents, the number of documents, and the number of sections are listed in Table 2. As shown in Table 2, we use 102 documents for the evaluation. Here, the accuracy and the true positive rate are evaluated using legal documents. The confu-

sion matrix [7], showing the predicted and actual classifications, is considered in the evaluation. A confusion matrix and the meaning of each cell are shown in Table 3.

Table 2. Statistics of documents used in the evaluation

Definition of document structure	Number of documents	Number of sections
Style information	12	292
Numbering	90	2430
Total	102	2722

Table 3. Confusion matrix [7] and meaning of each cell

		Predicted	
		Negative	Positive
Actual	Negative	True negative (TN)	False positive (FP)
	Positive	False negative (FN)	True positive (TP)

Kohavi and Provost [7] define the following terms for a 2 x 2 confusion matrix;

- Accuracy: (TN + TP) / (TN + FP + FN + TP)
- True positive rate (recall, sensitivity): TP / (FN + TP)

In this study, true positive means that the proposed method estimates a correct section. False positive means that the proposed method estimates an incorrect section. False negative means that the proposed method does not estimate a correct section. True negative means that the proposed method does not estimate an incorrect section. In accordance with the above definitions, the accuracy and the true positive rate are calculated to evaluate the proposed method. The confusion matrix generated from the results of document structure detection is shown in Table 4. As a result of evaluation with 102 documents, the proposed method estimated 2669 sections including 20 false positives. However, it failed to estimate 73 sections that should have been estimated.

Table 4. Results of document structure detection

		Predicted	
		Negative	Positive
Actual	Negative	0	20
	Positive	73	2649

As a result, the accuracy and the true positive rate are calculated as follows:

- Accuracy: (0 + 2649) / (0 + 20 + 73 + 2649) = 96.6%
- True positive rate (recall, sensitivity): 2649 / (73 + 2649) = 97.3%

False positives were mainly occurred at style-based algorithm. Some captions of figures and tables, which have only style information, such as bold style, and don't have numbered object, such as "Fig." or "Table", are recognized as a title. On the other hand, false negatives are caused by skipped numbers. Some documents have irregular numbering, e.g. "Section 1", "Section 2" and "Section 4". "Section 4" is not recognized as an element of document structure.

6 Application

In this section, the proposed method is applied to analyze legal documents. We focused on analyzing and comparing some Terms of Use of web services. Terms of Use is a legal document, so comparing some Terms of Use among web services is useful for obtaining characteristics of each service, comparing each service, and identifying risks of the service. However, Terms of Use published online have two types of document structure definitions: some use numbering-based format while others use style-based format. We can assume these Terms of Use to be examples of diverse and unstructured documents. The proposed method is used to analyze both types of definitions.

First, in Subsection 6.1, sample Terms of Use are analyzed by using the proposed method and articles defined in them are extracted. Second, in Subsection 6.2, extracted articles are compared among web services. Through this application, the effectiveness of proposed method is discussed.

6.1 Sample Terms of Use and results of document structure estimation

In this section, the Terms of Use of five web services are targeted. Two of them use style-based format while others use numbering-based format. Parts of Terms of Use are shown in Fig. 7. In Fig. 7, one uses style-based format while another uses numbering-based format. In both, each article has a title and its description.

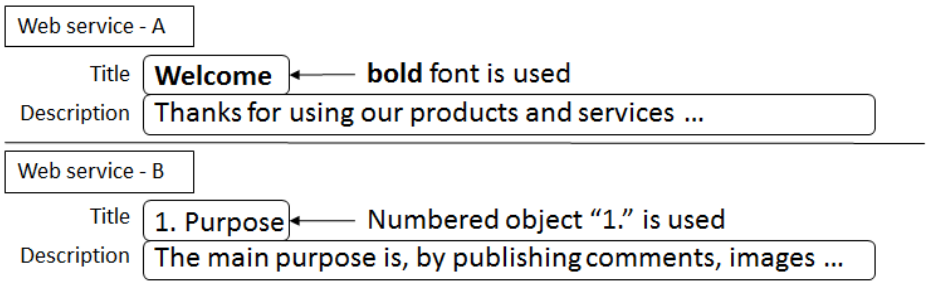


Fig. 7. Part of Terms of Use of two web-services

First, starting points of each section are detected by predefined style or regular expressions for numbering objects. For example, the regular expression "([\s\n\r]+|^)([0-9]+)(\.\s+)" is used for numbering-based format. Bold font are searched in Terms of

Use written in style-base format. The results of document structure estimation are shown in Table 5. We can see that both style- and numbering-based algorithms can estimate document structure correctly.

Table 5. Results of document structure estimation

	Truth	Extracted (TP)
Web service - A	12	12
Web service - B	13	13
Web service - C	16	16
Web service - D	16	16
Web service - E	22	22

6.2 Comparison of articles

In this subsection, articles extracted in the previous subsection are compared. First, each article extracted by using the proposed method is represented by a Vector Space Model with TF-IDF term weighting [8, 9]. Second, an article is selected as a query and similar articles are searched for by calculating cosine similarity [10] between vectorized articles in other services. Here, the range of similarity scores is 0.0 - 1.0, and higher scores are better. Since there is no common format of Terms of Use, contents of Terms of Use or granularity of articles can differ. Therefore, we use two types of clues to obtain similar articles: a title and its text description. This means that similar articles are searched for by using title-title similarity and text-text similarity. Finally, similar articles are aligned to the query article, and a comparison table is created.

A part of the comparison table is shown in Table 6, which lists two results. Note that a part of description is shown to sanitize in Table 6. The query article is the same. However, a similar article searched for by using its title is aligned in the “result 1” row, and a similar article searched for by using its description is aligned in the “result 2” row. In “result 1”, although both articles have similar title, descriptions are different from each other. For example, “Web service - E” mentions its jurisdiction, “Web service - C” claims disputes between users. On the other hand, in “result 2”, though two articles have different title, both descriptions claim their jurisdiction. Actually, “Web service - E” is related to online shopping, while “Web service - C” is related to social network service. Then, the characteristics of web services are appeared in comparison table.

Table 6. A part of comparison table

		Query article	Similar article
Result 1	Company	Web service - E	Web service - C
	Title	Disputes	Disputes between You and Other Users
	Description	Any dispute or claim relating in any way to your use of any our services will be resolved by binding arbitration ... Conflicts that arise from the Service will be governed primarily under the exclusive jurisdiction of the District Court of Tokyo or the Tokyo Summary Court.	You are solely responsible for your interaction with other users of the Service and other parties that you come in contact with through your use of the Service...
Result 2	Company	Web service - E	Web service - D
	Title	Disputes	Governing Law and Jurisdiction
	Description	Any dispute or claim relating in any way to your use of any our services will be resolved by binding arbitration ... Conflicts that arise from the Service will be governed primarily under the exclusive jurisdiction of the District Court of Tokyo or the Tokyo Summary Court.	These Terms and Conditions will be governed by the laws of Japan. Conflicts that arise from the Service or conflicts between Users and the Company related to the Service will be governed primarily under the exclusive jurisdiction of the District Court of Tokyo or the Tokyo Summary Court.

7 Conclusion

Active research on text analytics is accompanying the growth in available electronic documents. One critical challenge for obtaining meaningful insights from these document in text analytics is preprocessing of unstructured and diverse documents. In this paper, we presented a novel preprocessing method to estimate document structure that uses numbering-based and style-based algorithms. In addition, a new Table-of-Contents-based algorithm was also used. Proposed method refers style information, numbering object, and Table-of-Contents. These clues are commonly used for documents, and don't depend on domains. It means that our method can be applied to not only legal domain, but also other domains.

The proposed three-algorithms method was evaluated by using 102 documents, which had 2722 sections, and both numbering-based and style-based definitions of document structure were used. The proposed method was found to be able to analyze both types of document structure with high accuracy (96.6%) and a high true positive rate (97.3%). It is thus accurate enough to be used for practical use.

In addition, one application for analyzing legal documents was demonstrated. Terms of Use published online on five web services were analyzed by using the proposed method, a Vector Space Model with TF-IDF term weighting, and a cosine similarity score. The advantages of proposed method are (1) it estimates both numbering- and style-based document structure and (2) based on the results of document structure detection, it extracts article title and its test description, information that is useful for analyzing diverse documents.

As a result, we conclude the proposed method can preprocess unstructured and diverse documents for text analytics.

References

1. Song Mao, Azriel Rosenfeld, and Tapas Kanungo: Document Structure Analysis Algorithms: A Literature Survey. Proc. SPIE 5010, Document Recognition and Retrieval X, 11 pages (2003)
2. Davide Pasetto, Hubertus Franke, Weihong Qian, Zhili Guo, Honglei Guo, Dongxu Duan, Yuan Ni, Yingxin Pan, Shengua Bao, Feng Cao, and Zhong Su: RTS - an integrated analytics solution for managing regulation changes and their impact on business compliance. Proceedings of the ACM International Conference on Computing Frontiers (CF '13), Article 24 , 8 pages (2013) DOI=http://dx.doi.org/10.1145/2482767.2482798
3. Yoichi Hatsutori and Katsumasa Yoshikawa: A Method to Estimate Document Structure from Text Document and its Application to Law Articles, JURISIN 2015 (2015)
4. Isamu Iwai, Miwako Doi, Koji Yamaguchi, Mika Fukui, and Yoichi Takebayashi: A Document Layout System Using Automatic Document Architecture Extraction, CHI'89 Proceedings, pages 369-374, (1989)
5. Hervé Déjean: Numbered Sequence Detection in Documents. Proc. SPIE 7534, Document Recognition and Retrieval XVII, 753405 (18 January 2010), 12 pages (2010)
6. Hirokazu Igari, Akira Shimazu, and Koichiro Ochimizu: Document Structure Analysis with Syntactic Model and Parsers: Application to Legal Judgments, Jurisin 2011 (2011)
7. Ron Kohavi and Foster Provost: Glossary of Terms, Machine Learning, Vol. 30, No. 2-3, pp. 271-274 (1998)
8. Yiming Yang and Xin Liu: A reexamination of text categorization methods, SIGIR-99, 8 pages (1999)
9. Salton, G., Wong, A., and Yang, C. S.: A Vector Space Model for Automatic Indexing. Commun. ACM, Vol. 18, Num. 11, pp. 613-620 (1975)
10. Sandeep Tata and Jignesh M. Patel: Estimating the selectivity of tf-idf based cosine similarity predicates, ACM SIGMOD Record, Volume 36, Issue 2, June 2007, Pages 7-12 (2007)

Voluntary Manslaughter? Intention-to-Kill in Meta-Argumentation with Supports

Ryuta Arisaka and Ken Satoh

National Institute of Informatics, Tokyo, Japan
ryutaarisaka@gmail.com, ksatoh@nii.ac.jp

Abstract. In a criminal case, the judge's decision making often involves proving, beyond any reasonable doubt, the defendant's intention to commit a crime from material evidence. A valid decision should be supported by some material evidence, and neither the material evidence itself nor the support that it gives to the conclusion should be invalidated by any other material evidence. Luckily, this sounds a familiar topic in abstract argumentation with supports. We show an argumentation theory which roughly follows the tradition of evidential support, but which is a meta-argumentation (or extended argumentation) where an argument can attack an attack/support arrow. We model our example of intention-to-kill in the theory.

1 Introduction

The process to establish the defendant's intention to commit a crime from material evidence, an essential task in a criminal case judgement, is often non-monotonic. Suppose the judge is presented just two pieces of factual evidence by eye witnesses: (1) that the defendant threw a sharp Japanese chef knife which has a blade 13.3 centimeters in length; and (2) that it hit the victim's back of head to cause cerebellum stab wound and intracranial hemorrhage, then the defendant's intention to kill is most certainly inferred from them. However, we now add yet other evidence - (3) The victim is his daughter, and the defendant was very fond of her. (4) He threw the knife 3.3 meters away from her. (5) He was in a habit of throwing the knife at his wife's foot once drunk and irritated. And (6) she did not die on the spot, but he did not chase her leaving the scene. Taken together with (3), (4), (5) and (6), the first two evidence seem no longer self-sufficient for proving his intention to kill [1].

Accumulation of evidence being in this manner, the reasoning process as taken by a judge tends to be complex with potentials to errors. Still, no errors are admissible. Take manslaughter cases, while voluntary

manslaughter carries a sentence of death penalty, indefinite imprisonment or imprisonment for a definite term of 5 years or longer under the Criminal Justice System of Japan, involuntary manslaughter carries a much lighter sentence of imprisonment for a definite term of 3 years or longer. Hence, any error in judgement has a severe consequential implication to those involved in manslaughter cases.

In order to support the judge's decision making, in this work we investigate abstract argumentation theory [11] which is a theory for reasoning about arguments and attacks, appropriate for evidential reasoning [5, 23]. As the decision in a criminal case must be a certain one, however, there is a reason to be cautious when we face the following situation: an argument a_1 is attacked by another argument a_2, which is in turn attacked by an argument a_3. Under the original theory [11], we get the conclusion that a_3 defends and accepts a_1 because a_1 counters a_2's attack on a_1. In a way, a_3 is indirectly giving a support to a_1. But real examples are not always that clean, and one argument could be attacking another without giving any support to those the latter is attacking. Consider for instance: (x) the defendant did not have any intention to kill his daughter. (1) the defendant threw a sharp Japanese chef knife which has a blade 13.3 centimeters in length. (y) the defendant did not have any Japanese chef knife. Here (y) attacks (1) which attacks (x), and yet it is really not the case that (y) defends (x), for the one who states (y) mentions nothing of the defendant's intention to kill (that is, he/she may be in a belief that the defendant fired a gun instead of throwing a knife). Contrast this with (x)-(1)-(z) where (z) is: the defendant was in London when his daughter was killed in Tokyo, which forms a 'clean' case where the standard notion of defence applies. In situations such as intention-to-kill inference, it is not safe to take a support for granted out of indirect supports, and such distinction as seen between (x)-(1)-(y) and (x)-(1)-(z) should be expressed. We make use of a support relation [7, 9, 20–22] along with the attack relation for that purpose. Also, we allow a situation where some evidence is for confirming or rejecting an attack or a support from some evidence to some evidence, which occurs naturally in situations similar to our earlier example. For instance, (1) attacks (x), and (5) is better considered attacking the attack than (1) directly. Our argumentation theory will be consequently both bipolar [7, 9, 20–22], i.e. containing the two relations, and meta-argumentational [7, 12, 17], i.e. facilitating the higher-order attacks and supports. The exact interpreta-

tion of support varies from a work to a work, as classified by Cayrol and Lagasquie-Schiex [10]. If some a_1 supports some a_2, then: deductive support [7] ensures that acceptance of a_1 licences that of a_2; necessary support [20, 21] ensures that acceptance of a_2 necessitates that of a_1; and evidential support [22] guarantees: that a set of arguments, if acceptable, contains special non-attackable arguments that are always accepted; and that all the other arguments are connected by a chain of the support relation from (one or more of) the special arguments. Our interpretation will be roughly in the evidential tradition; however, our framework will be meta-argumentational and will contain no special arguments.

One significant advantage of argumentation theory in general is that it is very intuitive even for non-specialists of formal methods. Verification of relations among evidence for or against verdicts can be simplified. In fact, as we are to show later, it may be even possible to use this theory to study concluded past cases deeper and test their soundness.

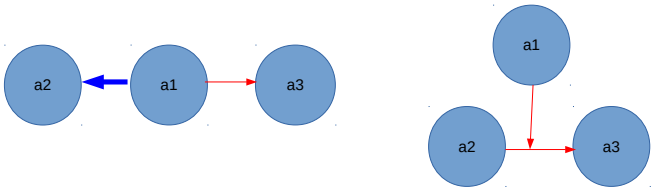
In the rest, we will: touch upon technical background (in Section 2); present our meta-argumentation with supports (in Section 3); and model one actual criminal case we saw earlier (from (1) to (6)) in a little more details than given here (in Section 4), before drawing conclusions.

2 Technical Background

Argumentation frameworks allow one to reason about attacks and defences among arguments. In [11] an argument is an abstract entity, and an attack relation is a binary relation over a pair of the set of arguments. Suppose a_1 and a_2 are arguments, then a_1 is considered to be attacking a_2 just when the binary relation is defined for (a_1, a_2). The following definitions are taken from the original theory. An argumentation framework is a pair (A, R) where A is a finitary set of abstract entities *arguments*, and R is a binary relation defined over them. An argument a_1 is said to *attack* an argument a_2 iff $(a_1, a_2) \in \mathrm{R}$. A set of arguments $A_1 (\subseteq A)$ is *conflict-free* iff no $a_1 \in A_1$ is attacking any $a_2 \in A_1$. A set of arguments A_1 *defends* a_1 iff for any a_2 attacking a_1, there is some $a_3 \in A_1$ that attacks a_2. A set A_1 *accepts* a iff A_1 defends a. A set A_1 is admissible iff A_1 accepts all its members. A set A_1 is a preferred set (extension) iff A_1 is admissible and there exists no $A_1 \subset A_2$ such that A_2 is admissible. There are other notions such as complete sets (extensions), stable sets (extensions), and the grounded set (extension). More information for these semantics can

be found elsewhere in the literature.

There are many extensions to the original theory. Two that are of particular importance to our technical development are: the support relation which characterises arguments supporting - instead of attacking - arguments [7, 9, 20–22]; and the argument-to-argument-or-attack relation [16] or even more relaxed a relation [12]. In the left drawing below, a thin-lined arrow is an attack, while a thick-lined arrow is a support: an argument a_1 supports a_2 and attacks a_3, or we just say $(a_1, a_2) \in \mathrm{R}_{\mathrm{sup}}$ and $(a_1, a_3) \in \mathrm{R}$, treating $\mathrm{R}_{\mathrm{sup}}$ as the support relation, and R as the attack relation. In the right drawing below, a_1 attacks the attack of a_2 on a_3, i.e. (a_2, a_3). Any argumentation theory that allows an attack or a support to an arrow (an edge) as well as an argument (a node) is considered a meta-argumentation theory, which in the literature is also called extended argumentation.



There are several proposals for the exact semantics of support, such as deductive support, necessary support and evidential support:

Deductive support [7] $(a_1, a_2) \in \mathrm{R}_{\mathrm{sup}}$ means: (1) if a_1 is accepted, then so is a_2; and (2) if a_2 is not accepted, then a_1 is also not accepted.

Necessary support [20, 21] $(a_1, a_2) \in \mathrm{R}_{\mathrm{sup}}$ means if a_2 is accepted, then so must a_1 be.

Evidential support [22] There are special kinds of arguments $\mathfrak{n}$ with or without a subscript. They are always unattacked, are unsupported and are accepted. Let $\mathrm{R}^{+}_{\mathrm{sup}}$ be such that: (1) $(\mathfrak{n}, a) \in \mathrm{R}_{\mathrm{sup}}$ materially implies $(\{\mathfrak{n}\}, a) \in \mathrm{R}^{+}_{\mathrm{sup}}$; and (2) $(A, a) \in \mathrm{R}^{+}_{\mathrm{sup}}$ only if, for each $a_x \in A$, $(A_x, a_x) \in \mathrm{R}^{+}_{\mathrm{sup}}$ for some $A_x \subseteq A \backslash \{a_x\}$. Then other arguments a that are accepted in A must satisfy $(A_1, a) \in \mathrm{R}^{+}_{\mathrm{sup}}$ for some $A_1 \subseteq A$. Note that in [22] an attacker is assumed to be a set of arguments in the manner similar to Nielsen-Parsons argumentation theory [19].

Among the three, our interpretation of support will be evidential, broadly construed. However, it will be a meta-argumentation and will contain no

explicit set of ns. While very useful in some other applications, in this work we do not require Nielsen-Parsons group attacks or supports.

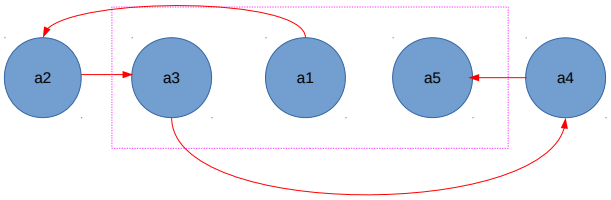
3 Our Meta-Argumentation with Supports

We assume that $\mathbb{N}$ is the class of natural numbers including 0. We assume a set of evidence made available to a judge and a set of possible decisions he/she may make. We denote the former by $\mathsf{Evidence}$ and the latter by $\mathsf{Verdict}$. Now, let Rel^0 be $\mathsf{Evidence} \times (\mathsf{Evidence} \cup \mathsf{Verdict}) \rightharpoonup \{-1, 1\}$, and let Rel^{k+1} be $\mathsf{Evidence} \times \mathsf{dom}(\mathsf{Rel}^k) \rightharpoonup \{-1, 1\}$ for any $k \geq 0$. They form our meta-argumentation framework which is: $(\mathsf{Evidence}, \mathsf{Verdict}, \mathsf{Rel})$ for $\mathsf{Rel} = \bigcup_{k \in \mathbb{N}} \mathsf{Rel}^k$. The two values 1 and -1 indicate attack and support respectively.

We denote an element of $\mathsf{Evidence}$ by e, an element of $\mathsf{Verdict}$ by v, an element of $\mathsf{Evidence} \cup \mathsf{Verdict}$ by a, and an element of $\mathsf{dom}(\mathsf{Rel})$ by rel, all with or without a subscript. For any ordered set Γ (not the particular symbol Γ but any ordered set), $\pi(n, \Gamma)$ is: the n-th element of Γ if $0 < n \leq |\Gamma|$; or else undefined.

3.1 Acceptability semantics

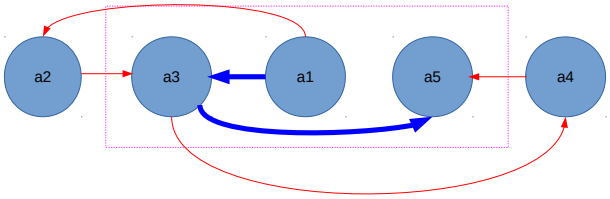
That the decision in a criminal case should be first of all a certain one imposes a requirement on the attack and the support relations with respect to acceptability of arguments. Let us illustrate our notion of support by going through a few examples.



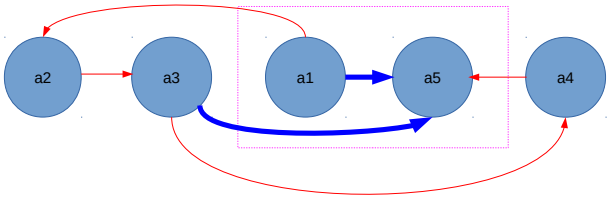
Here a_k is attacking a_{k+1} for each $1 \leq k \leq 4$. The argument a_1 is not attacked by any argument, and so it is an acceptable argument. Although a_3 is being attacked by a_2, it in turn is being attacked by a_1. In the classic acceptability semantics (Section 2), it means that a_1 is nullifying the allegation that a_2 raises against a_3. Since no other arguments are attacking a_3, a_1 indirectly supports a_3 which to a_1 is acceptable. In a similar

manner, a_5 is also accepted by a_1.

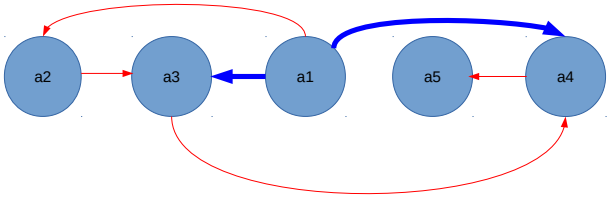
As we mentioned in Section 1, however, an argument may be attacking another argument a_x without giving any support to an argument a_x is attacking. There are numerous examples of these kinds in every-day argumentation, not really limited to legal cases. Hence, there is a good reason why we should make any support explicit in our argumentation framework.



a_1 supports a_3 here, by which we more specifically mean that a_1 supports whatever attacks and whatever supports a_3 is making: a_3's attack on a_5 and a_3's support for a_5, in this example. Consequently a_1 supports a_5 through its support for a_3. Take a contrast against another example.



Here, a_1 cannot be considered to be supporting a_3. Finally what if we have the following:



Then there is a known problem: a_1 both supports (directly) and attacks (via a_3) a_4. This we, like others, consider inconsistent.

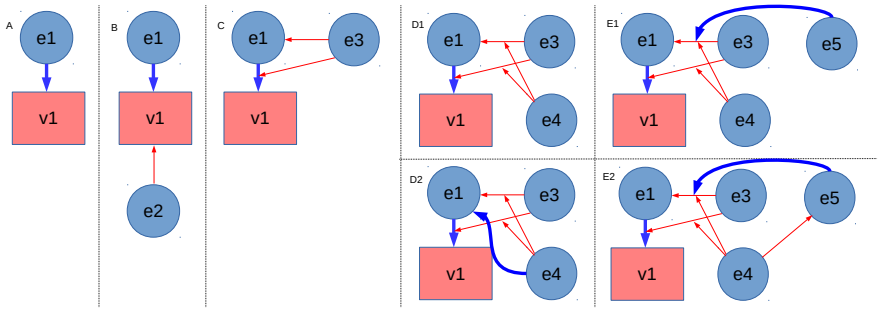
Definition 1 (Supports and attacks).
We say that a_1 supports $\pi(2,rel)$ iff $\mathsf{Rel}(\pi(1,rel),\pi(2,rel)) = 1$ **and** ei-

ther a_1 is $\pi(1, rel)$ or else there exists a sequence $a_1, \ldots, a_n$, $n \geq 1$, such that $\mathsf{Rel}(a_i, a_{i+1}) = 1$ for all $1 \leq i \leq n-1$ **and** $\mathsf{Rel}(a_n, \pi(1, rel)) = 1$. We say that a_1 attacks $\pi(2, rel)$ iff $\mathsf{Rel}(\pi(1, rel), \pi(2, rel)) = -1$ **and** either a_1 is $\pi(1, rel)$ or else there exists a sequence $a_1, \ldots, a_n$, $n \geq 1$, such that $\mathsf{Rel}(a_i, a_{i+1}) = 1$ for all $1 \leq i \leq n-1$ **and** $\mathsf{Rel}(a_n, \pi(1, rel)) = 1$.

In this definition and elsewhere, we may explicitly use **and** to indicate formal 'and' with semantics of classic logic conjunction for a disambiguation purpose. **or** is its dual.

Definition 2 (Inconsistent framework).
We say that $(\mathsf{Evidence}, \mathsf{Verdict}, \mathsf{Rel})$ is inconsistent iff there exists $\pi(2, rel)$ and $e \in \mathsf{Evidence}$ such that $\mathsf{Rel}(rel)$ is defined **and** e both attacks and supports $\pi(2, rel)$.

We assume in the rest that an argumentation framework is not inconsistent. Let us now go through some examples, as shown below, to connect these supports and attacks to the judge's decisions. The idea is to decide, given a set of verdicts and a set of evidence, which verdicts are acceptable, and, moreover, which evidence actually support them.



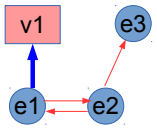
In A, we have some verdict v_1 which is supported by an evidence e_1. Intuition speaks that v_1 is acceptable in this case. But it is not sufficient just to prove existence of a supporting evidence to conclude that v_1 is acceptable in general. For example, there may be another evidence e_2 as in B attacking the same verdict. Or like in C, e_1 or the attack arrow (e_1, v_1) may be attacked which is not in turn attacked. Consequently, we need also examine all evidence attacking v_1 or supports for it, and need show that their attacks are nullified by other evidence. An example is D1 where we can conclude that v_1 be acceptable because e_4 nullifies the attacks by e_3. In D2, again v_1 is acceptable, but here it is supported not just by e_1 but also by e_4. Now, a little more complicated cases could be E1 and E2

which both extend D1. In those cases like E1, we shall conclude that v_1 be *not* acceptable, the reason being that e_5 in a way, by supporting the attack (e_3, e_1), attacks e_1, and that there is no evidence attacking e_5. If, however, there is an attack arrow (e_4, e_5) as in E2, then we conclude that v_1 is acceptable, in comparison. In contrast to D2, in this case it is not that e_4 supports v_1.

We make our intuition explicit.

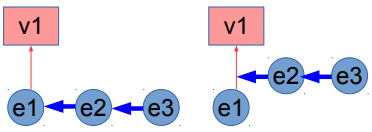
Definition 3 (Conflict-freeness). Let $\mathsf{Test}(rel, A)$ be a recursive predicate: $\mathsf{Test}((a_1, a_2), A)$ iff $a_1, a_2 \in A$; and $\mathsf{Test}((a_1, rel), A)$ iff $a_1 \in A$ **and** $\mathsf{Test}(rel, A)$. Let $\delta(A)$ be $A \cup \{rel \in \mathsf{Rel} \mid \mathsf{Test}(rel, A)\}$. We say that $A \subseteq \mathsf{Evidence} \cup \mathsf{Verdict}$ is conflict-free iff there is no $a \in A$ and $x \in \delta(A)$ such that a attacks x.

$\delta(A)$ contains, along with the elements of A, all the arrows (support or attack) that occur in A. Consider the following graph, $\delta(\{v_1, e_1, e_2\})$ contains all but e_3 (because it is not in the set) and the attack from e_2 to e_3 (because, while e_2 is in the set, e_3 is not in the set). $\delta(\{v_1, e_1, e_3\})$ contains v_1, e_1 and e_3 as well as the support arrow, and is conflict-free.



Definition 4 (Support set). For $x \in \mathsf{Evidence} \cup \mathsf{Verdict} \cup \mathsf{dom}(\mathsf{Rel})$, we say that $A \subseteq \mathsf{Evidence} \cup \mathsf{Verdict}$ is its support set iff each $a_1 \in A$ supports x **and** A is conflict-free. We say that a support set A of x is maximal iff there is no greater support set of x.

In the below diagrams, $\{e_2, e_3\}$ is a maximal support set for e_1 (left) and respectively for the support of e_1 for v_1 (right).



The first purpose of a maximal support set is to know all the nodes that are indirectly attacking some node through a node that directly attacks it. See E1 in the earlier figure. The second purpose is to know, for any acceptable verdict, which evidence actually support it.

Definition 5 (Defence). $A \subseteq \mathsf{Evidence} \cup \mathsf{Verdict}$ defends $a \in A$ iff: for every $a_1 \in \mathsf{Evidence} \cup \mathsf{Verdict}$ that attacks a, there is no $a_x \in A$ that supports a_1 (i.e. no set member supports its attacker) **and** either: a_1 and every member of every maximal support set A_1 of a_1 is attacked by some member of A; or (a_1, a) and every member of every maximal support set A_2 of (a_1, a) is attacked by some member of A.

Definition 6 (Admissible and preferred set). We say that $A \subseteq \mathsf{Evidence} \cup \mathsf{Verdict}$ is admissible iff A is conflict-free **and** A defends every $a \in A$. We say that it is preferred iff there is no greater admissible set of A.

Definition 7 (Acceptability of verdicts). We say that v is accepted by $E \subseteq \mathsf{Evidence} \cup \mathsf{Verdict}$ iff there is a preferred set which contains both v and E **and** E is a maximal support set of v.

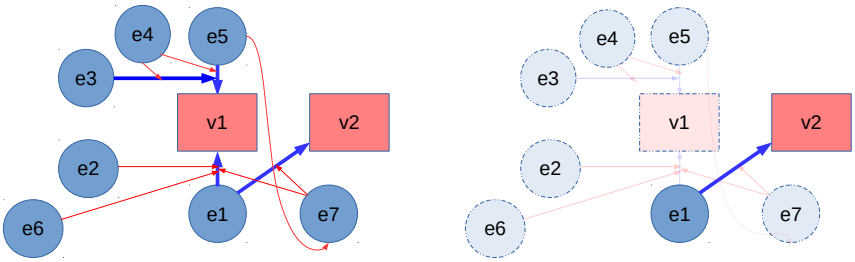
As we can see from this definition, preferred sets are important because we judge acceptance of a verdict based on them. However, it is not necessary that every member of a preferred set is connected by supports. See E2 in the earlier figure. Hence, we extract from a preferred set a maximal support set for the verdict.

4 An Example

In Section 1, we illustrated our motivation with one actual criminal case. Here we show more complete picture of the case (some details are simplified) within our argumentation framework. There are two verdicts: v_1

and v_2, and several evidence.

v_1 : Man is guilty of voluntary manslaughter.
v_2 : Man is guilty of involuntary manslaughter.
e_1 : Man threw a sharp knife at his daughter walking down a staircase, and the knife hit her back of head causing her to sustain cerebellum stab wound and intracranial hemorrhage.
e_2 : Usually in knife-throwing, aim is poor at 3.3 meters distance from a target.
e_3 : Man was throwing the knife at his wife when drunk and irritated.
e_4 : Man was in a habit of shouting "I kill you!" when drunk and irritated, and was throwing the knife at the floor near his wife's foot just to scare his wife and daughter.
e_5 : Man shouted "I kill you!" at his daughter.
e_6 : Daughter did not die on the spot and started to move away but man did not chase her.
e_7 : Man loved his daughter.



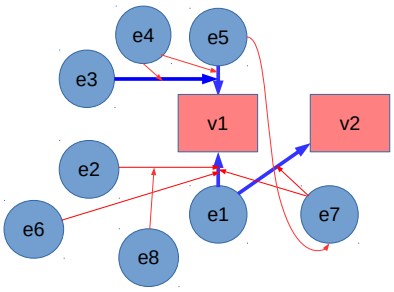
The left graph represents this criminal case. As is clear, v_1 is supported by two material evidence: e_1 and e_5. Further, the support of e_5 for v_1 is supported by e_3. However, firstly, the support of e_5 for v_1 as well as that of e_3 for the support of e_5 for v_1 is attacked by e_4 which is not countered. Therefore, there is no preferred set that contains both v_1 and

e_5. Similarly, e_2 and e_5 both attack the support of e_1 for v_1; it is not possible for a preferred set containing v_1 and e_1 to exist. But consider v_2 now. It is supported by e_1. While the support is attacked by e_7, we have e_5 that attacks it. Neither e_5 nor its attack is attacked by any nodes. Here, $\{e_1, e_5, v_2\}$ forms a preferred set. But among them, the only node that gives a support to v_2 is e_1, which derives the right drawing above showing that e_1 is the support for v_2.

4.1 For analysis of the past cases

It may be possible to inspect the past cases with the aid of this argumentation framework, testing their soundness with additional and unconsidered evidence.

One observation which did not enter into the judge's verdict, which nonetheless might have been relevant, is the following fact: if an object is right below ourselves, it is much easier to hit it. The daughter in the above case was walking down a staircase and, according to the original case document, her head was about the man's feet hight. Therefore, there could have been e_8 : It is easier to hit an object which is down below. The modified argumentation framework is shown below, where e_2 can no longer attack the support without getting countered.



As they say, drawings help, and we believe the same for the judge's decision making.

5 Conclusion

We presented a meta-argumentation with supports to model an intention-to-kill criminal case judgement. Though roughly in the evidential support tradition [22], our framework is a meta-argumentation (as it is more natural in the judge's decision making process to presume that some evidence act for or against attack or support arrows but not directly on evidence), does not involve the set of special arguments and does not demand an acceptable argument to be supported directly or indirectly by those arguments that are by default accepted, leading to a new interpretation of support. These consideration are reasonable because we begin with a set of verdicts and start to backward-trace supports from them, unlike in [22] where the task is rather to forward-trace (in the direction of arrows) supports from special nodes. Note also in the example in Section 4, $\{e_1, e_5, v_2\}$ is a preferred set even though e_5's support arrow is being attacked. We believe that further research in meta-argumentation with supports will aid the judge's decision making by simplifying their reasoning process. Dealing with criminal intention itself is not new: there are for instance works [13, 14] in defeasible logic. The appeal of argumentation theory, nonetheless, is the very intuitive character it has. Outside abstract argumentation, there are many other works in structured argumentation [8, 11, 18, 24] in which an argument is expressed more concretely as a pair of a set of premises and a conclusion. There is a support relation holding in each pair from the premises to the conclusion. An argument may be rebutted (the conclusion is attacked), undercut (the support is attacked) or undermined (some premise is attacked), and such greater expressiveness is an appeal. However, it is also true that concretisation of abstract entities can entail introduction of more assumptions about the nature of an argument. We have for instance pointed out [2] that not every natural argument may be clear-cuttingly divided into supporting premises and a conclusion. There is also the difficult issue of contrariness in the modern logic [15], which is, as far as our awareness extends, being studied [3], but which has not been fully resolved, yet. Much of these difficulties can be encapsulated in abstract argumentation. As such, there is a benefit in working within a sufficiently abstract framework if identification of some key matters around attacks and supports is an issue. Lastly, we mention that our approach, as well as argumentation-theoretic approaches, is treated evidential reasoning. There are also studies into

abductive reasoning purporting to find the best possible explanations to causes [4] or those that combine evidential and abductive reasoning [6].

References

1. Hanrei times no. 985, page 300, 1999.
2. Ryuta Arisaka and Ken Satoh. Balancing Rationality and Utility in Logic-Based Argumentation with Classical Logic Sentences and Belief Contraction. In *PRIMA*, pages 168–180, 2016.
3. Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. Dealing with Generic Contrariness in Structured Argumentation. In *IJCAI*, pages 2727–2733, 2015.
4. Floris Bex, Trevor J. M. Bench-Capon, and Katie Atkinson. Did He Jump or Was He Pushed? Abductive Practical Reasoning. *Artificial Intelligence and Law*, 17(2):79–99, 2009.
5. Floris Bex, Henry Prakken, Chris Reed, and Douglas Walton. Towards a Formal Account of Reasoning about Evidence: Argumentation Schemes and Generalisations. *Artificial Intelligence and Law*, 11(2):125–165, 2003.
6. Floris Bex, Peter J. van Koppen, Henry Prakken, and Bart Verheij. A Hybrid Formal Theory of Arguments, Stories and Criminal Evidence. *Artificial Intelligence and Law*, 18(2):123–152Bex10.pdf, 2010.
7. Guido Boella, Dov M. Gabbay, Leendert van der Torre, and Serena Villata. Support in Abstract Argumentation. In *COMMA*, pages 111–122, 2010.
8. Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.
9. C. Cayrol and M. C. Lagasquie-Schiex. On the Acceptability of Arguments in Bipolar Argumentation Frameworks. In *ECSQARU*, pages 378–389, 2005.
10. Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in Argumentation Graphs: Towards a Better Understanding. In *Scalable Uncertainty Management*, pages 137–148, 2011.
11. Phan M. Dung. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games. *Artificial Intelligence*, 77(2):321–357, 1995.
12. Dov M. Gabbay. Semantics for Higher Level Attacks in Extended Argumentation Frames Part 1: Overview. *Studia Logica*, 93(2-3):357–381, 2009.
13. Guido Governatori and Vineet Padmanabhan. A Defeasible Logic of Policy-Based Intention. In *AI*, pages 414–426, 2003.
14. Guido Governatori and Antonino Rotolo. BIO Logical Agents: Norms, Beliefs, Intentions in Defeasible Logic. *JAAMAS*, 17(1):36–69, 2008.
15. Laurence R. Horn. *A Natural History of Negation*. The University Chicago Press., 2nd edition, 2001.
16. Sanjay Modgil. Reasoning about preferences in argumentation frameworks. In *Artificial Intelligence*, volume 173, pages 901–934, 2009.
17. Sanjay Modgil and Trevor J. M. Bench-Capon. Metalevel argumentation. *Journal of Logic and Computation*, 21(6):959–1003, 2011.
18. Sanjay Modgil and Henry Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.
19. Søren Holbech Nielsen and Simon Parsons. A generalization of Dung's Abstract Framework for Argumentation. In *Argumentation in Multi-Agent Systems*, pages 54–73, 2006.
20. Farid Nouioua and Vincent Risch. Bipolar Argumentation Frameworks with Specialized Supports. In *ICTAI*, pages 215–218, 2010.

21. Farid Nouioua and Vincent Risch. Argumentation Frameworks with Necessities. In *SUM*, pages 163–176, 2011.
22. Nir Oren, C. Reed, and M. Luck. Moving between Argumentation Frameworks. In *COMMA*, pages 379–390, 2010.
23. Henry Prakken. Analysing reasoning about evidence with formal models of argumentation. *Law, Probability & Risk*, 3(1):33–50, 2004.
24. Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.

Modeling Attempted Crime in Criminal Law

Jiraporn Pooksook[1] Phan Minh Dung[1]
Ken Satoh[2]

[1] Computer Science and Information Management, Asian Institute of Technology, Pathumthani, Thailand
jpooksook@gmail.com
dung.phanminh@gmail.com
[2] National Institute of Informatics, Tokyo, Japan
ksatoh@nii.ac.jp

Abstract. In the court of law, a person can be punished for attempting to commit a crime. There are distinct legal rules for determining attempted crime whereas the last act rule (also called proximity rule) represents the strictest approach that is also widely used in many courts. In this paper we provide a formal modelling of the proximity rule for a conviction of attempted crime using argument-based knowledge bases system. We illustrate our model using a well-known court case on obtain money by false pretenses in UK.

Keywords: Structured Argumentation, Attempted Crime, Attempt and Intention, Lact-Act (or Proximity)-Rule

1 Introduction

Attempt to commit a crime is punishable by law. Attempt is defined as an act that has been carried out purposely but failed in completion of an indictable offence[3] [2, 16, 23]. Section 6 of Massachusetts General Laws [17] stated that "*whoever attempts to commit a crime by doing any act toward its commission,but fails in its perpetration,...,be punished as follows:...*" Section 80 of Thai Criminal Code [21] also stated that "*whoever commences to commit an offence, but does not carry it through, or carries it through, but does not achieve its end...*" The purpose in the punishment of inchoate crime is to direct remedial action to those people who have explicitly expressed their dangerousness. Also the police can legally intervene when they find a man attempting to commit any indictable offence [16, 23].

Two fundamental elements that are in any legal proof of attempt are an intention (mens rea) and not mere preparatory acts (actus reus). The important issue in proving actus reus is that how to distinguish between preparatory acts and attempt. To solve this problem different courts employ distinct tests like the proximity rule, the substantial step rule, the probable desistance, etc [2,

[3] (Criminal Attempts Act 1981 of UK law)[22] "... a person does an act which is more than merely preparatory to the commission of the offence..."

16, 23]. The strictest approach is the proximity rule, also called proximity test or last act test stating that "*if a person has engaged the last proximate act which is immediately connected with the substantive crime then his/her acts are considered to be an attempt to commit the offence* [3, 16, 19]." The proximity rule is applied in UK and Thai courts while the substantial step rule is more liberal and widely used in US courts [16, 23, 25]. To exemplify the proximity rule, let us consider a court case in UK.

Example 1. R v Eagleton (1855)[19]. Mr.Eagleton had a contract with a local soup kitchen to supply bread to the poor for free. For each loaf of bread Mr.Eagleton asked a ticket from a needy person. Then he reported the number of loaves supplied with collected tickets to an officer who would credit his account and pay him later. Before the first payment it was later found that Mr.Eagleton had baked loaves to the poor weighting much less than the contracted specification. Hence, he was convicted for an attempt to obtain money by false pretenses.[4]

In this paper we model the proximity rule where the heart of the work lies in the model of the last act. We also model the legal rule for deriving the intention of the defendant from his acts. Attempt to commit a crime is then defined as an unfinished the last act with intent.

The paper is organized as follows. In section 2 we describe the necessary language used in the representation of the formalization. In section 3, we first model the last proximate act and intention in the proximity rule, and then represent the rule of attempted crime. We illustrate in section 4 the application of our proposal by applying it to the case of Mr.Eagleton. Finally we discuss future work and conclude.

2 Preliminaries

2.1 Abstract Argumentation

An abstract argumentation framework consists of a set of arguments AR and an attack relation att $\subseteq$ AR$\times$AR where (X,Y) $\in$ att means that the argument X attacks the argument Y.

A set S of arguments attacks an argument A if some argument in S attacks A. S is said to defend A if S attacks any argument attacking A. A set of arguments S is admissible iff S is conflict-free and defends all arguments attacking its arguments. Several semantics are introduced. Preferred extensions are maximal (wrt. set inclusion) admissible sets of arguments while complete extensions are admissible sets containing all arguments it defends. Grounded extension is the smallest (wrt. set inclusion) complete extension. A stable extension is a conflict-free set of arguments attacking every arguments outside it.

[4] It is interesting to note that had Eagleton not submitted the collected credits, he would not have been convicted in any court applying the proximity rule though he could still be convicted in an US court using the more liberal substantial-step rule provided that his intention to cheat could be proved.

2.2 Knowledge Base and Structured Argumentation Framework

A structured argumentation framework is based on a language $\mathcal{L}$, a non-empty set of ground atoms and their classical negations. A positive literal represents an atom and the negation of it is called a negative literal. A contradictory set of literals contains a pair of $f, \neg f$ where f is an atom. Atoms in $\mathcal{L}$ are classified into domain atoms and non-domain atoms. Domain atoms represent propositions of the concerned domains while non-domain atoms are of the form ab_d stating that the defeasible rule d can not be applied. A knowledge base [8, 9] is represented as a triple consisting of (RS,RD,BE)[5] where BE is a set of literals representing unchallenged observations and facts, RD and RS are sets of defeasible rules and strict rules of the form $f_1, ..., f_n \Rightarrow / \rightarrow f_0$ respectively. We denote hd(r)=f_0 and bd(r)=$\{f_1, ..., f_n\}$ for any rule r $\in$ RS $\cup$ RD.

An argument wrt a knowledge base (RS,RD,BE) is constructed by repeatedly applying the following steps:

1. For any $f \in$ BE, $[f]$ is an argument with conclusion f.
2. Let r be a rule in a form of $f_1, ..., f_n \rightarrow / \Rightarrow f_0$ in $RS \cup RD$ and $A_1, ..., A_n$ are arguments with conclusion $f_i, 1 \leq i \leq n$, respectively.
   A = $[A_1, ..., A_n \rightarrow / \Rightarrow f_0]$ (also denoted by A = $[A_1, ..., A_n, r]$) is also an argument with a conclusion f_0 where r is referred to as the last rule of A.

   The conclusion of the argument A is denoted by cnl(A). A is *defeasible* if it contains defeasible rules. A is a *basic defeasible argument* if its last rule is defeasible.

   An argument B is a subargument of A=$[A_1, ..., A_n, r]$) iff B=A or B is a subargument of some A_i.

Let A,B are arguments wrt. a knowledge base (RS,RD,BE). We say A **attacks** B if one of the following conditions is satisfied.

- A rebuts B (at B'), i.e. B' is a basic defeasible subargument of B and cnl(A) and cnl(B') are contradictory.
- A undercuts B (at B'), i.e. B' is a basic defeasible subargument of B and hd(A)= $ab_{last(B')}$.

More about the semantic of the knowledge base system can be seen in [8, 9].

2.3 Domain Representation Language

We assume a set $\mathcal{F}$ of fluents representing properties in the concerned domain and a set $\mathcal{A}$ of basic actions in it. Both $\mathcal{F}$ and $\mathcal{A}$ are finite and disjoint. For

[5] In [8] a knowledge base also contains a binary preference relation $\preceq$ between defeasible rules. But in this paper we only work with rules without preferences. Hence the preference relation is dropped.

convenience, we use f, f_i to denote fluents, and a, a_i to denote basic actions. We also use F,G to denote finite sets of fluents and their negations.

We view an act as a sequence of basic actions.

Executing a basic action can cause more than one fluents to hold. To represent causal effects of action, we introduce *dynamic causal laws* of the form *causes(a,G,F)*, a slight generalization to the one in [4, 11, 14, 15] stating that if action a is executed under pre-conditions F then fluents in G normally hold.

For convenience, we often identify a singleton set $\{f\}$ with its element f.

Rules specifying the effects of actions are described as follows.

$$d_1 \ : \ causes(a, G, F), holds(F, S), happen(a, S) \Rightarrow holds(G, res(a, S))$$

where *holds(F,S)* denotes that all fluents in F hold in the situation S.

The following rule states the obvious that if sets of fluents G, G' hold in situation S then $G \cup G'$ also holds.

$$r_h : \ holds(G, S), holds(G', S) \rightarrow holds(G \cup G', S)$$

Fluents normally do not change when an action is executed unless there are reasons for exceptions like they represent the effects caused the action execution. This property is specified by inertial rules below:

$$d_{i1} : \ inertial(G, a), holds(G, S), happen(a, S) \Rightarrow holds(G, res(a, S))$$

where *res(a,S)* represents the situation resulting from the execution of the basic action a in a situation S, and

$$\begin{aligned}
&d_{f,a} : \Rightarrow inertial(\{f\}, a) \\
&d_{i2} : \ causes(a, f, F) \Rightarrow ab_{d_{f,a}} \\
&d_{i3} : \ causes(a, \neg f, F) \Rightarrow ab_{d_{f,a}} \\
&d_{i4} : \ inertial(G, a), inertial(F, a) \Rightarrow inertial(G \cup F, a) \\
&d_{i5} : \ inertial(F, a_1), inertial(F, a_2 ... a_n) \Rightarrow inertial(F, a_1.a_2 ... a_n)
\end{aligned}$$

where the rule $d_{f,a}$ state that by default a fluent f is inertial wrt. a basic action 'a' where rules d_{i2}, d_{i3} represent possible exceptions. Rules d_{i4}, d_{i5} are self-explanatory.

The rules specifying the effects of acts consisting of sequences of basic actions are give below.

$$\begin{aligned}
&d_{c1} : \ causes(a, G, F), G' \subseteq G \rightarrow causes(a, G', F) \\
&d_{c2} : \ causes(a, G, F), F \subseteq F' \Rightarrow causes(a, G, F') \\
&d_{c3} : \ causes(a, G, F), causes(a, G', F) \rightarrow causes(a, G \cup G', F) \\
&d_{c4} : \ causes(a_1 ... a_n, G, F), F_1 \subseteq F, inertial(F_1, a_1 ... a_n), \\
&\qquad causes(a_{n+1}, G', G \cup F_1), G_1 \subseteq G, inertial(G_1, a_{n+1}) \\
&\qquad \Rightarrow causes(a_1 ... a_n a_{n+1}, G' \cup G_1, F)
\end{aligned}$$

2.4 Reactive and Non-reactive (Voluntary) Actions

We distinguish two kinds of actions: reactive actions and non-reactive ones.

Reactive actions are basic actions whose execution are normally triggered when the pre-conditions in their dynamic causal laws hold. For example an action 'receive money' will be triggered normally after a customer submits a withdrawal slip to a bank teller unless the action is interrupted by some exceptions.

In contrast, *non-reactive actions are voluntary actions* that may or may not be executed when the pre-conditions in their dynamic causal laws hold.

In an act we assume that non-reactive actions are carried out by the perpetrator of the crime while reactive actions are executed by the any other people.

3 Modeling Attempted Crime

As we discussed in Section 1, an attempt to commit an offence is a crime. An attempted crime requires both intention to commit an offence and acts that are proximate to the crime intended.

If a person has intention to commit an offence and already engaged in the last proximate act, he/she is guilty for an attempt to commit the offence according to the proximity rule[6].

We first model the two essential key points of attempted crime which are the last proximate act and intention as discussed in the following sections. Attempted crime is modeled then.

3.1 The Last Proximate Act

The last proximate act must be immediately connected with the substantive offence[7]. The proximity test requires the accused to engage to the last proximate act as explained in [16]:

> "the defendant have engaged in the last proximate act, that is, that he have done everything which he believes necessary to bring about the intended result."

In other words, a defendant is guilty if he has done every thing that he needs to do in order to achieve the alleged crime. The last proximate act usually be the act that lead to intended consequence naturally. For instance Mr.Eagleton had already claimed for his credits and waited for the payment. The court found

[6] which stated that "a person has engaged in the last proximate act that is, that he have done everything which he believes necessary to bring about the intended result "[16]

[7] (R v. Eagleton,1855) [1, 16] "The mere intention to commit a misdemeanour is not criminal. Some acts is required, and we do not think that all acts towards committing a misdemeanour are indictable. Acts remotely leading towards the commission of the offence are not to be considered as attempts to commit it, but acts immediately connected with it are..."

that he had engaged the last proximate act which immediately connected to the substantive crime. Therefore the last proximate act does not necessary to be a physical last act e.g. obtaining the payment which is a reactive action that automatically be executed by default.

Therefore, the last proximate act could be viewed as a sequence of basic actions where the first basic action is executed by the defendant voluntarily while other actions are triggered and carried out by other actors.

Formally we represent the last act as a sequence of basic actions $a_1...a_n$ where a_1 is non-reactive and the rest is reactive.

The following rule is our first try of formalizing the last act to commit a crime.

$$
\begin{aligned}
&offence(G), aware\text{-}offence(G,S), G = G_1 \cup F_1,\\
&causes(a_1...a_n, G_1, F), holds(F,S), non\text{-}reactive(a_1), reactive(a_2...a_n),\\
&F_1 \subseteq F, inertial(a_1...a_n, F_1)\\
&\Longrightarrow last\_act(a_1...a_n, G, S)
\end{aligned}
$$

The predicate offence(G) indicates that a criminal offence is committed if all the fluents in G hold in the same situation and reactive($a_2...a_n$) holds if each of the actions in the sequence is inactive. The definition of offence is domain specific and should be given. For example, in the Eagleton case, G consists of two fluents denoting distributing of underweight bread and obtaining money for it.

Intuitively, the rule states that an act $a_1 \ldots a_n$ represents a last-act to commit an offence represented by a set of fluent G, in a situation S if executing the act will cause a part G_1 of G to hold while the other part F_1 already holds before the act and still holds after the act due to its inertiality wrt the act itself.

There is an implicit but fundamental understanding in the above rule that the action of the defendant a_1 is relevant to the offence. To see this point consider the act consisting of the sequence of basic actions "go_on_holidays . obtain_payment". Suppose Eagleton has already submitted the credits. According to the rule above, this act would be qualified as the last act. Therefore, Mr.Eagleton would not be convicted that is absurd as go_on_holidays is legally irrelevant to the offence of the case.

The above rule should be reformulated as follows.

$$
\begin{aligned}
&d_1 : offence(G),\ aware\text{-}offence(G,S), G = G_1 \cup F_1,\\
&relevant\text{-}causes(a_1...a_n, G_1, F), holds(F,S),\ non\text{-}reactive(a_1),\ reactive(a_2...a_n),\\
&F_1 \subseteq F, inertial(a_1...a_n, F_1)\\
&\Longrightarrow last\_act(a_1...a_n, G, S)
\end{aligned}
$$

Let A= $a_1 \ldots a_n$. For simplicity, we denote the subsequence $a_2 \ldots a_n$ by $A \backslash a_1$.

$$
\begin{aligned}
&d_{rel(A,G,F)} : causes(A,G,F) \Rightarrow relevant\text{-}causes(A,G,F)\\
&d_{nrel} : causes(A \setminus a_1, G, F) \Rightarrow ab_{d_{rel(A,G,F)}}
\end{aligned}
$$

The rule $d_{rel(A,G,F)}$ states that if by executing the sequence of actions A under preconditions F causes a result G then that all actions in A are relevant actions which lead to G under F after executing A. However the second rule shows that if the first action of A is removed and by executing $A \setminus a_1$ can lead to G, the rule $d_{rel(A,G,F)}$ is not applicable.

For illustration of the two above rules in dealing with the relevant causes, consider the sequence of actions

A = go_on_holidays . obtain_payment where
G={Distributed_underweight_bread, Obtained_money}, and
F = {Distributed_underweight_bread, Aware_of_underweight_bread,
Submitted_credits}.

It is easy to see that causes(A,G,F) holds. But because causes(obtain_payment, G, F) also holds, we can conclude $ab_{d_{rel(A,G,F)}}$. Therefore any arguments using the last rule $d_{rel(A,G,F)}$ to conclude relevant-causes(A,G,F) is attacked and hence, can not be accepted.

3.2 Voluntary Character of Non-reactive Action and Intention

Attempt in criminal law requires the proof of an intent to commit an offence at the time the crime occurs. Intention is in a person's mind and it cannot be proven objectively. Therefore the courts provide rules for deriving intention from what the defendant has already done as stated in Commonwealth v. Niziolek (1980)[7] that

> "*Intent is a state of mind. The intention of a person...is to be ascertained by his acts and the inference is to be drawn from what is externally visible. Intent ordinarily cannot be proven directly because there is no way of reaching into and examining the operations of the human mind, but you may determine the defendant's intent from any statement or act done or act omitted and all the other circumstances which indicate his state of mind, provided you first find that any or all of such circumstances occurred. A person is presumed to intend the natural and probable consequences of his own acts.*"

Commonwealth vs. Bruce Blake (1990)[6] provides a clear guidance for drawing intentions from acts: "*The law presumes that a person intends the ordinary consequences of his voluntary acts.*"

Hence if a person has started a voluntary act, it is clear that he/she has an intent to do the act and its ordinary consequence. For instance, a person who has a boarding-pass and has passed the immigration has an intent to board the plane. Another illustration, as Mr.Eagleton had submitted his credits of distributing underweight bread loaves he had demonstrated clear intent to obtain the payment by false pretenses [8].

[8] ...by returning the tickets to the relieving officer he intended to represent that he had delivered the loaves mentioned in them of the weights stated. [19]

The proximity rule requires the defendant to engage in the last proximate act. A person who has engaged in actions of an act voluntarily intends the foreseen consequences of the act. Without an unexpected interruption the intended result will happen if the person completes the act. Hence, we formalize intention in the proximity rule as follows.

$d_2 : started(a_1...a_n, S), non\text{-}reactive(a_1), reactive(a_2...a_n) \Rightarrow intend(a_1...a_n, S)$

$d_3 : holds(F, S), causes(A, G, F), F_1 \subseteq F, inertial(A, F_1), intend(A, S) \Rightarrow intend(A, G \cup F_1, S)$

Rule d_2 states that if a person has already started an act A under a situation S and the first action a_1 is non-reactive (and hence voluntarily) while $a_2...a_n$ are reactive (and hence triggered by a_1) then the person has an intent to do A in the situation S. Rule d_3 states that if a person intends to do an act A in a situation S and the execution of A normally causes G to hold under preconditions F and F holds in the situation S then the person has an intent to commit the act A to achieve the result $G \cup F_1$ in the situation S where $F_1 \subseteq F$ is inertial wrt A.

Note that *in an act we assume that non-reactive actions are carried out by the perpetrator of the crime while reactive actions are executed by the any other people.*

3.3 Attempt in Proximity Rule

When a person has already engaged in the last proximate act purposely, he/she is guilty for an attempt to commit the offence according to the proximity rule. Hence, we represent the rule of attempt as follows.

$$d_4 : last\_act(A, G, S), started(A, S), intend(A, G, S), uncompleted(A, S) \\ \Longrightarrow attempt(G, S)$$

The meaning of rule d_4 is that if G is an offence (see d_1) and A is the last proximate act to successfully accomplish G in a situation S and a person has started A in S with an intention to achieve the offence G then the person attempts to commit the offence G in the situation S.

We specify the started and completed-predicates below.

$r_1 : happen(a_1, S) \rightarrow started(a_1...a_n, S)$
$r_2 : happen(a_1 \ldots a_n, S), happen(a_{n+1}, res(a_1 \ldots a_n, S)) \rightarrow happen(a_1...a_n a_{n+1}, S)$
$r_3 : \neg happen(a_1, S) \rightarrow uncompleted(a_1, S)$
$r_4 : happen(a_1...a_n, S), \neg happen(a_{n+1}, res(a_1...a_n, S)) \rightarrow uncompleted(a_1...a_{n+1}, S)$
$r_5 : uncompleted(a_1...a_k, S) \rightarrow uncompleted(a_1...a_k a_{k+1}, S)$

where $res(a_1 \ldots a_n, S) = res(a_n, res(\ldots res(a_1, S) \ldots)$.

The first rule states that if the first basic action a_1 happened in a situation S then the act $a_1 ... a_n$ has already started. The rule r_2 states that if the basic action $a_1 \ldots a_n$ happened in a situation S and a_{n+1} happened as a result after executing $a_1 \ldots a_n$ then $a_1 \ldots a_n a_{n+1}$ has happened in S.

The third rule describes that if the basic action a_1 has not happened in a situation S then it is uncompleted in S. The rule r_4 explains that if the act $a_1 ... a_n$ happened in a situation S but a_{n+1} has not happened in the situation resulting from executing $a_1 \ldots a_n$ in S then the act $a_1 ... a_{n+1}$ is uncompleted in S.

The rule r_5 defines that if the basic action $a_1 ... a_k$ is uncompleted in a situation S then the series of action $a_1 ... a_k a_{k+1}$ is also uncompleted in S.

4 Illustration

Example 2. R v Eagleton (1855) (Continuing)

Let us design the domain descriptions from the case.

- Mr.Eagleton had a contract to provide bread loaf to the poor. He baked bread weighted less than the contracted specification. Hence he was aware that the loaves are underweight, that is represented by two fluents.
  Have_ underweight_ bread, Aware_ of_ underweight_ bread
- Mr.Eagleton distributed underweight bread loaves to the needy people and got credits ticket in return, that is described by the fluent.
  Distributed_ underweight_ bread, Received_ credits
- After receiving the credit tickets, Mr.Eagleton submitted them to obtain money, that are specified by the following fluents.
  Submitted_ credits, Obtained_ money

The act that Mr.Eagleton was executing is represented by the sequence of actions

$$bake\_underweight\_bread \,.\, distribute\_underweight\_bread \,.\, submit\_credits \,.\, obtain\_payment$$

that is for simplicity abbreviated by

bub.dub.sc.op

The first three actions are non-reactive and whose executor is Mr.Eagleton while the last action is reactive.

In summary, the set of actions and fluents are as follows:

$\mathcal{A}$= $\{bub, dub, sc, op\}$

$\mathcal{F}$= $\{Have\_underweight\_bread, Aware\_of\_underweight\_bread,$
$Distributed\_underweight\_bread, Received\_credits, Submitted\_credits, Obtained\_money\}$

The effects of the actions are specified by:

– causes(bub, Aware_of_underweight_bread, [])[9]
causes(bub, Have_underweight_bread, [])

– causes(dub, Received_credits, Have_underweight_bread)
causes(dub, Distributed_underweight_bread, Have_underweight_bread)

– causes(sc, Submitted_credits, Received_credits)

– causes(op, Obtained_money, Submitted_credits)

Note that we follow the convention to identify a singleton set $\{e\}$ with its element e. For example, the dynamic causal law causes(dub, Received_credits, Have_underweight_bread) is identified with causes(dub, {Received_credits}, {Have_underweight_bread}).

We design a knowledge base K=(RS,RD,BE) as follows.

– Let $S_1 = \text{res}(\text{bub}, S_0)$, $S_2 = \text{res}(\text{dub}, S_1)$, $S_3 = \text{res}(\text{sc}, S_2)$, $S_4 = \text{res}(\text{op}, S_3)$

BE consists of the following facts

- offence({Distributed_underweight_bread, Obtained_money})
  stating that obtaining money for distributing underweight bread loaves is an offence.
- happen(bub,S_0), happen(dub,S_1), happen(sc,S_2)
  stating that three actions bake_underweight_bread, distribute_underweight_bread, submit_credits are already happened in the situation S_0, S_1, S_2 respectively.
- $\neg$ happen(op,S_3)
  stating that Mr.Eagleton has not executed the action obtain_payment in the situation S_3.

– RS consists of rules $r_1, ..., r_5, r_h$ specifying in Section 2 and 3 together with the following rule r_e

holds(Aware_of_underweight_bread, S),
offence({Distributed_underweight_bread,Obtained_money})
$\rightarrow$ aware-offence({Distributed_underweight_bread,Obtained_money}, S)

After distributing under-weight-breads, Eagleton may or may not have any under-weight-bread left. In other words, the fluent Have_underweight_bread is not inertial wrt action "dub". This property is represented by the following rule

[9] As the preconditions for the act bub are irrelevant for the case we ignore them, for simplicity. Therefore the set of preconditions of bub is empty.

$r_{d_i}: \rightarrow ab_{d_{hub,dub}}$

where $d_{hub,dub}$ is the rule

$d_{hub,dub}: \Rightarrow inertial(Have\_underweight\_bread, dub)$

– RD consists of rules $d_1, \ldots, d_4, d_{f,a}, d_{c1}, ..., d_{c4}, d_{i1}, ...d_{i5}, d_{rel(A,G,F),d_{nrel}}$ together with following instances of the rule d_1:

$d_5 : causes(bub, Aware\_of\_underweight\_bread, [\,]), holds(True, S_0), happen(bub, S_0)$
$\Longrightarrow holds(Aware\_of\_underweight\_bread, S_1)$

$d_6 : causes(bub, Have\_underweight\_bread, [\,]), holds(True, S_0), happen(bub, S_0)$
$\Longrightarrow holds(Have\_underweight\_bread, S_1)$

$d_7 : causes(dub, Received\_credits, Have\_underweight\_bread),$
$holds(Have\_underweight\_bread, S_1), happen(dub, S_1)$
$\Longrightarrow holds(Received\_credits, S_2)$

$d_8 : causes(dub, Distributed\_underweight\_bread, Have\_underweight\_bread),$
$holds(Have\_underweight\_bread, S_1), happen(dub, S_1)$
$\Longrightarrow holds(Distributed\_underweight\_bread, S_2)$

$d_9 : causes(sc, Submitted\_credits, Received\_credits),$
$holds(Received\_credits, S_2), happen(sc, S_2)$
$\Longrightarrow holds(Submitted\_credits, S_3)$

$d_{10} : causes(op, Obtained\_money, Submited\_credits),$
$holds(Submited\_credits, S_3), happen(op, S_3)$
$\Longrightarrow holds(Obtained\_money, S_4)$

The last act in Mr.Eagleton attempt to obtain money by false pretenses can be concluded by applying rule d_1 as follows:

– $a_1.a_2 = submit\_credits.obtain\_payment.$
– The offence of distributing under-weight-bread and obtaining money for it is represented by $G = \{Obtained\_money, Distributed\_underweight\_bread\}$.

– It is not difficult to see (by applying rule d_5, $d_{aub,dub}$, d_{c4}) that $causes(bub.dub, \{Aware\_of\_underweight\_bread\}, [])$ holds. Hence we can conclude using the strict rule r_e that
aware-offence({Obtained_money, Distributed_underweight_bread},S_2)
holds.

Therefore we can conclude

$$last\_act(a_1.a_2, \{Obtained\_money, Distributed\_underweight\_bread\}, S_2) \quad (1)$$

Because Mr.Eagleton has already submitted the credits represented by the facts happen(sc,S_2) in BE from rule r_1 we can conclude that he has already started the last act represented by the literal

$$started(a_1.a_2, S_2) \tag{2}$$

Further since Mr.Eagleton has not received the money represented by the fact $\neg$ happen(op,S_3), we can conclude using rule r_4 to conclude

$$uncompleted(a_1.a_2, S_2) \tag{3}$$

Since the action submit_credits is a non-reactive one, hence Mr.Eagleton has executed it voluntarily. Therefore from rule d_2 we can conclude that Mr.Eagleton intended to executed the last act represented by the literal

$$intend(a_1.a_2, S_2) \tag{4}$$

From $relevant\text{-}causes(a_1.a_2, \{Obtained\_money, Distributed\_underweight\_bread\}, F)$ where F={Received_credits, Distributed_underweight_bread} and F holds in S_2 we can conclude

$$intend(a_1.a_2, \{Distributed\_underweight\_bread, Obtained\_money\}, S_2) \tag{5}$$

From 1, 2, 3 and 5 we can conclude

$$attempt([Obtained\_money, Distributed\_underweight\_bread], S_2)$$

The illustration shows that according to the knowledge base K, Mr.Eagleton was guilty of an attempt to obtain money by false pretenses which is similar to the judgment of the Appeal court which stated that "the act of receiving credit for distribution of the loaves of bread was the last act necessary on behalf of defendant and makes him guilty of an attempt to obtain money by false pretenses "[19].

5 Discussion and conclusion

We provide a formalization of the proximity rule in the law of attempt to commit a crime. A key idea of this paper is the classification of actions into non-reactive and reactive actions where non-reactive actions are voluntary. This insight allows us to capture the legal interpretation of the last-proximate-act as well as the intention in proximity rule.

We illustrate our model using a classical case of attempting to obtain money by false pretenses in UK.

Kolkorn [12] has also discussed the proximity rule where last-acts are understood as the last part in a plan supposedly of the defendant but put forward by the prosecutor and the defendant has to defend himself by attacking this plan. In other words, the prosecutor could design any act as the last-act as long

as it is the last part in any plan supposedly of the defendant but designed by him/her. No proof is required to show that the last part of the established plan immediately connects to intended crime. This obviously does not capture the spirit of the last-proximate act. As in proximity rule, attempt is based solely on last-act and intent, one can commit a crime simply driven by an opportunistic decision at a moment without having a well-designed plan, we do not involve a plan in our modeling of attempt. Further Kolkorn [12] assumes that the defendant has intention to carry out the plan supposedly from him but put forward by the prosecutor unless he can counter by putting forward a counter-plan. This again does not follow the court's guidelines that a person intends the ordinary consequence of his actions.

There are clear legal guidelines for delineating the functions of district and appellate courts. According to [20] "*At a trial in a U.S. District Court, witnesses give testimony and a judge or jury decides who is guilty or not guilty...The appellate courts do not retry cases or hear new evidence. They do not hear witnesses testify. There is no jury. Appellate courts review the procedures and the decisions in the trial court to make sure that the proceedings were fair and that the proper law was applied correctly.*"

In short, the appellate courts deal only with the matter of law while the matter of facts has to be dealt with in trial courts. In Eagleton, the fact that he baked each loaf of bread weighting around 10% less than the contract was concluded in the trial court. The appellate court only deals with the law and with the qualitative fact that Eagleton had baked underweight bread.

In this paper, we model the interpretation of proximity rule by appellate courts and hence we do not deal with matter of facts and how qualitative facts could be drawn from quantitative evidences. To evaluate quantitative facts and evidence may require probabilities or other quantitative approaches to uncertainties. We plan to look at this problem in the future.

There are extensive amount of research in the literature on the dynamics of intention within an agent's mind [4, 5, 13, 18, 24]. This line of research is especially important for designing the mind of an agent. In this paper, we are not concerned with the role of intention in designing an agent mind. We focus on modelling legal guidelines for drawing intent of persons from their past actions.

There are other tests using in proving criminal attempt like the substantial step, which is mostly used in US courts. We plan to look at them in the follow-up works.

References

1. Allen, M.: Textbook on Criminal Law. OUP Oxford 11ed,(2011).
2. Andenaes, J.: General Part of the Criminal Law of Norway. Fred B Rothman & Co,(1965).
3. Anyangwe, C.: Criminal Law The General Part. Langaa RPCIG, (2015).
4. Baral, C. & Gelfond, M.: Reasoning About Intended Actions. In: Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, pp. 689–694. AAAI Press,(2005).

5. Blount, J., and Gelfond, M.: Reasoning about the intentions of agents. In: Artikis, A.; Craven, R.; Kesim CÂÿ icÂÿekli, N.; Sadighi, B.; and Stathis, K., eds., pp.147–171. Logic Programs, Norms and Action, volume 7360 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg,(2012).
6. Commonwealth vs. Bruce Blake. 409 Mass. 152, (1990), http://masscases.com/cases/sjc/409/409mass146.html, June 2016.
7. Commonwealth v. Niziolek, 380 Mass. 513 , 528, (1980), http://masscases.com/cases/sjc/380/380mass513.html, June 2016.
8. Dung, P. M.: An Axiomatic Analysis of Structured Argumentation with Priorities. Artif. Intell., Elsevier Science Publishers Ltd. 231, 107–150 (2016).
In: ECAI 2014 - 21st European Conference on Artificial Intelligence, pp. 267–272. Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014),(2014).
9. Dung, P. M.: Argumentation for Practical Reasoning: An Axiomatic Approach. In: International Conference on Principles and Practice of Multi-Agent Systems, pp. 20–39. Springer International Publishing,(2016).
10. Dung, P. M.: 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).
11. Dung, P. M.: Representing Actions in Logic Programming and Its Applications in Database Updates. In: Proceedings of the Tenth International Conference on Logic Programming, pp. 222–238. The MIT Press,(1993).
12. Kolkorn M.: Representing and Reasoning with Custom Law in Logic Programming, AIT Master Thesis,(2016).
13. Lorini, E. & Herzig, A.: A logic of intention and attempt. Syn- these. Springer. 163, 45–77 (2008).
14. Gelfond M., Baral C.: Reasoning Agents In Dynamic Domains. In: Workshop on Logic-Based Artificial Intelligence,(2000).
15. Gelfond M., Lifschitz V.: Representing Actions in Extended Logic Programming. In: Proc. of the JICSLP,(1992).
16. LaFave,Wayne R.: Criminal Law. 5th ed. West Academic Publishing,St. Paul, Minn(2010).
17. Massachusetts General Laws, https://malegislature.gov/Laws/GeneralLaws/PartIV/TitleI/Chapter274/Section6, June 2016.
18. Rao, A. S., Georgeff M. P.: Modelling rational agents within a BDI-architecture. In: Proc. of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers,(1991).
19. R v. Eagleton 6 Cox C.C. 559 (1855), http://www.casebriefs.com/blog/law/criminal-law/criminal-law-keyed-to-lafave/attempts/regina-v-eagleton/ ,http://swarb.co.uk/lisc/Crime18491899.php, August 2016.
20. The U.S. court role, http://www.uscourts.gov/about-federal-courts/court-role-and-structure, October 2016.
21. Thai Criminal Code Section 80, http://library.siam-legal.com/thai-law/criminal-code-attempt-sections-80-82/, June 2016.
22. UK Criminal Attempts Act 1981, http://www.legislation.gov.uk/ukpga/1981/47, June 2016.
23. Williams, G. L.: Textbook of criminal law. London : Stevens,(1978).
24. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley & Sons,(2002).
25. Vachanasvasti, K.: The explanation of Criminal Law 1. Pholsiam printing publishing(Thailand),542–554(2010).

Argument-based Logic Programming for Analogical Reasoning

Teeradaj Racharak[1,2], Satoshi Tojo[2], Nguyen Duy Hung[1]
, and Prachya Boonkwan[3]

[1] School of Information, Computer and Communication Technology,
Sirindhorn International Institute of Technology,
Thammasat University, Thailand
r.teeradaj@gmail.com, hung@siit.tu.ac.th
[2] School of Information Science,
Japan Advanced Institute of Science and Technology, Japan
racharak@jaist.ac.jp, tojo@jaist.ac.jp
[3] National Electronics and Computer Technology Center, Thailand
prachya.boonkwan@nectec.or.th

Abstract. Analogical reasoning can be understood as a kind of resemblance of one thing to another, thus assigning properties from one context to another. This kind of reasoning is used quite often by human beings, especially in unknown situations. In this paper, we present a formal and intuitive framework for analogical reasoning in an argument-based logic-programming-like language. A proof theory of our system is stated in the dialectical style, where a proof takes the form of dialogue between a proponent and an opponent of an argument.

Key words: Analogical Argumentation, Argumentation Schemes, Argument from Analogy, Argument-based Logic Programming

1 Introduction and Motivation

Analogical reasoning can be understood as a kind of resemblance of one thing to another, thus assigning properties from one context to another. This kind of reasoning is used quite often by human beings in real-life situations, especially when humans encounter an unseen situation. For example, in legal argumentation, attorney Gerry Spence reasons by analogy in the case of Silkwood v. Kerr-McGee Corporation (see Example 1) [1].

Example 1. (The Silkwood case) Karen Silkwood was a technician who had the job of grinding and polishing plutonium pins used to make fuel rods for nuclear reactors. She was active in union activities and had investigated health and safety issues at the plant. She had testified before an atomic energy commission that Kerr-McGee had violated safety regulations. Tests in 1974 showed that she had been exposed to dangerously high levels of plutonium radiation. High levels of radioactive contamination were found in her apartment. After she died

from radiation poisoning, her father brought an action against Kerr-McGee in which the Corporation was held to be at fault for her death on the basis of strict liability. Let us note that, in strict liability law, a person can be held accountable for the harmful consequences of some dangerous activity he was engaged in, without having to prove that he was aware of, intended the outcome, or even that he was negligent.

□

Spence's closing argument uses the analogy of the escaping lion, which had great rhetorical effect on the jury. According to his speech (p. 129 of [2]), he emphasized the statement *If the lion got away, Kerr-McGee has to pay.*

> Some guy brought an old lion on his ground, and he put it in a cage – and lions are dangerous – and through no negligence of his own through no fault of his own, the lion got away. Nobody knew how – like in the Silkwood case, *nobody knew how.* And, the lion went out and he ate up some people – and they sued the man. And they said, you know: *Pay. It was your lion and he got away.* And, the man says: *But I did everything in my power – I had a good cage – I had a lock on the door – I did everything that I could – I had security – I had trained people watching the lion – and it isn't my fault that it got away.* Why should we punish him? They said: *We have to punish him – we have to punish you – you have to pay.* You have to pay because it was your lion – unless the person who has hurt let the lion out himself.

Spence used the lion analogy during his closing argument, a part of the trial for summing up one's argument in the case. Spence used the example of the escaping lion to illustrate to the jury how strict liability works, showing them how cases of this kind had originated in English common law. Spence's argument (pp. 123-124 of [2]) in this famous case has said to be *as fine a closing argument as has ever been delivered in an American courtroom.*

Analogical reasoning is used quite often in everyday thinking. Practically, analogical reasoning should not be explicitly programmed and conclusions drawn by analogy must be debatable in the same way as ordinary domain information can be. Defeasible argumentation has been revealed itself to be a promising means for reasoning in this respect. Though many theoretical and practical developments build on Dung's abstract theory, *the use of analogy* for practical reasoning is still unexplored (cf. Section 4). Current frameworks (e.g. [15, 14, 17, 18]) may treat analogical reasoning by representing similarity in subtle ways; however, no satisfactory analysis on the interaction of analogical arguments is discussed. Thus, the nature of analogical reasoning is still not well-understood today. The goal of this paper is to study on this phenomena under the lens of structured argumentation. To the best of our knowledge, this is the first time the framework for analogical reasoning has been discussed (cf. Section 3). Preliminaries, related work and the conclusion are discussed in Section 2, Section 4 and Section 5, respectively.

2 Preliminaries

Argumentation Framework (AF) is another line of research that has its root in the study of Logic Programming and non-monotonic reasoning. The general idea here is that non-monotonic reasoning can be analyzed in terms of interactions between arguments for alternative conclusions. Non-monotonicity arises from the fact that arguments can be defeated by stronger counterarguments. On the one hand, the landmark paper [3] is entirely abstract and general-based framework. On the other hand, this paper works on *structured argumentation*.

Practically, structured argumentation has five elements, viz. **(1)** an underlying logical language; **(2)** notions of arguments; **(3)** notions of conflicts between arguments; **(4)** relations of attack among arguments; **(5)** and definitions of the status of arguments. These definitions of status of arguments are alternatively called *semantics* or *extensions*. They define what a set of arguments is accepted w.r.t. a rational reasoner's viewpoints.

2.1 Argumentation Schemes: Argument from Analogy

Argumentation schemes [4] are stereotypical non-deductive patterns of reasoning, consisting of a set of premises and a conclusion that is *presumed* to follow from the premises. Use of argumentation schemes is evaluated by a specific set of critical questions corresponding to each scheme. Precisely speaking, uttering an instance of scheme in a dialogue creates a presumption in favor of its conclusions and corresponding burden of proof for the other side in the dialogue to defeat this presumption by asking critical questions. If he/she gives a satisfactory answer to the question, the presumption is reinstated and the burden shifts back to the opponent of the argument to ask other critical questions, and so on.

Let us illustrate this with the argumentation scheme named *Argument from Analogy* given in [1] as follows:

1. A situation is described in C_1.
2. A is plausibly drawn as an acceptable conclusion in case C_1.
3. Generally, case C_1 is similar to case C_2.

$\therefore$ A is plausibly drawn as an acceptable conclusion in case C_2.

The following set of critical questions matches the scheme:

1. Are there respects in which C_1 and C_2 are different that would tend to undermine the similarity cited?;
2. Is A the right conclusion to be drawn in C_1?;
3. Is there some other case C_3 that is also similar to C_1, but in which some conclusion other than A should be drawn?.

Typically, there are three basic ways an argument can be attacked: **(1)** on one or more of premises; **(2)** on the conclusion; and **(3)** on the inferential link between premises and the conclusion. In this kind of scheme, an argument from a counter-analogy is used to attack the conclusion of the former argument from analogy as unacceptable, i.e. the second type of attacking.

Example 2. (Continuation of Example 1) In this case, the argument from analogy does all the work. To get a clue how Spence's argument can be represented as an Argument from Analogy, the scheme consists of three steps, i.e. from the base premise to the derived premise, from the derived premise to a similarity premise, and from there to the conclusion. Here, the base premise is *the lion case* and the derived premise is *the law held the owner of the lion liable for the harm*. A similar premise is *the lion case is similar to the Silkwood case*. Hence, the drawn conclusion is *the law holds Kerr-McGee liable for the harm*.

□

2.2 Concept Similarity Measure in Description Logics

The scheme Argument from Analogy (cf. Subsection 2.1) simply says that if two cases are similar and a particular conclusion is drawn in the first case, then the same conclusion may be drawn in the second case. That conditional represents the rule of inference behind the scheme. Unfortunately, the problem with this scheme is to define the notion of similarity.

In [5–7], we have developed a general framework in Description Logics (DLs) for concept similarity measure under an agent's preferences (in symbols $\overset{\pi}{\sim}$). Basically, π is a set of preferential aspects (called *preference profile*) and $\overset{\pi}{\sim}$ is a function mapping (under π) from a concept pair to a unit interval (i.e. $0 \leq x \leq 1$ for any real numbers x).

Definition 1 (Preference Profile [6]). *A* preference profile*, in symbol π, is a quintuple $\langle \mathfrak{i}^{\mathfrak{c}}, \mathfrak{i}^{\mathfrak{r}}, \mathfrak{s}^{\mathfrak{c}}, \mathfrak{s}^{\mathfrak{r}}, \mathfrak{d} \rangle$ where $\mathfrak{i}^{\mathfrak{c}}$ is called* primitive concept importance*, $\mathfrak{i}^{\mathfrak{r}}$ is called* role importance*, $\mathfrak{s}^{\mathfrak{c}}$ is called* primitive concepts similarity*, $\mathfrak{s}^{\mathfrak{r}}$ is called* primitive roles similarity*, and $\mathfrak{d}$ is called* role discount factor*.*

In DLs, we assume countably infinite sets CN of concept names and RN of role names that are fixed and disjoint. The set of concept descriptions, or simply concepts, for a specific DL $\mathcal{L}$ is denoted by $\mathsf{Con}(\mathcal{L})$. The set $\mathsf{Con}(\mathcal{L})$ is inductively defined on CN and RN with the use of concept constructors. In DLs, a knowledge base $\mathcal{K}$ is usually defined as $\langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a terminological component or TBox and $\mathcal{A}$ is an assertion component or ABox. However, some practical ontologies may exclude $\mathcal{A}$ from $\mathcal{K}$.

Definition 2 ([7]). *Given a preference profile π, two concepts $C, D \in \mathsf{Con}(\mathcal{L})$, and a TBox $\mathcal{T}$, a* concept similarity measure under preference profile *w.r.t. a TBox $\mathcal{T}$ is a function $\overset{\pi}{\sim}_{\mathcal{T}} : \mathsf{Con}(\mathcal{L}) \times \mathsf{Con}(\mathcal{L}) \to [0, 1]$, such that $C \overset{\pi}{\sim}_{\mathcal{T}} D = 1$ iff $C \equiv_{\mathcal{T}} D$ (*total similarity*) and $C \overset{\pi}{\sim}_{\mathcal{T}} D = 0$ indicates* total dissimilarity *between C and D.*

3 The Formal System: Analogical Reasoning

3.1 The Language

The object language of our system conforms to the familiar logic-programming-like style. That is, it contains a two-place one-direction connective that forms

rules out of literals; plus a newly introduced (one-direction) similarity connective $\overset{x}{\leftarrow\!\!\!\sim}$ that forms similarity premises. There are two types of literals. A *strong* literal is an atomic first-order formula A or such a formula preceded by a strong negation symbol (in symbols, $\neg A$). For any ground literals A, we say that A and $\neg A$ are complement and, in the meta-language, we denote by $\overline{A}$ a complement of A. A *weak* literal is a literal of *not* A, where A is a strong literal and *not* denotes *negation-as-failure*. Informally, *not* A reads as *there is no evidence that* A *is the case* whereas $\neg A$ reads as A *is definitely not the case*. In what follow, we use standard typographic conventions of Logic Programming.

Definition 3 (Strict Rule). *A strict rule is an expression of the form* $L_0 \leftarrow L_1, \ldots, L_n$ *where* $n \geq 0$ *and* L_i $(0 \leq i \leq n)$ *is a* strong *literal. If* $n = 0$*, it is referred to as a* fact*.*

Definition 4 (Defeasible Rule). *A defeasible rule is an expression of the form* $L_0 \Leftarrow L_1, \ldots, L_n$ *where* $n \geq 0$*,* L_0 *is a* strong *literal, and* L_i $(1 \leq i \leq n)$ *is a literal. If* $n = 0$*, it is referred to as a* presumption*.*

Syntactically, strict rules and defeasible rules differ only one aspect, i.e. only defeasible rules may contain weak literals in the body. Pragmatically, defeasible rules are used to represent tentative information which will be used if no one could disprove it whereas strict rules are used to represent strict information. For example, *fly*$(X) \Leftarrow$ *bird*(X) expresses *usually, a bird can fly* whereas *bird*$(X) \leftarrow$ *penguin*(X) expresses *all penguins are birds.*

Definition 5 (Similarity Rule). *A similarity rule is an expression of the form* $L_0 \overset{x}{\leftarrow\!\!\!\sim} L_1$ *where* L_0, L_1 *are first-order predicates and* $0 < x \leq 1$[1] *for any real number* x*.*

Our motivation is to form similarity premises by using similarity rules. The meaning of $\overset{x}{\leftarrow\!\!\!\sim}$ makes use of the notion of *similarity distance*; henceforth, we can employ the notion of $\overset{\pi}{\sim}$ (cf. Subsection 2.2). To the best of our knowledge, the connective $\overset{x}{\leftarrow\!\!\!\sim}$ is first introduced for analogical reasoning in this work. More precisely, the connective captures a one-direction relation between two predicates. The higher the value x is denoted, the more likely (one-direction) similarity of them may hold. Intuitively, the value 1 can be interpreted as *total similarity.* For example, *plutonium_plant* $\overset{1}{\leftarrow\!\!\!\sim}$ *lion* expresses *lions are totally similar to plutonium plants.*

Definition 6 (Logic Program). *A logic program* $\mathcal{P}$ *is a triple* $\langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ *where* SR *is a finite set of of strict rules,* DR *is a finite set of defeasible rules, and* SIM *is a finite set of similarity rules.*

In this work, we assume that every rule in a program $\mathcal{P}$ are ground. Nevertheless, our examples may use the usual convention, i.e. *schematic rules with variables*, in Logic Programming.

[1] When $x = 1$, we may remove it.

Definition 7 (Derivation). *Let $\mathcal{P}$ be a logic program and L be a ground literal. A derivation for L from $\mathcal{P}$ with an* analogy degree *w, in symbols $\mathcal{P} \overset{w}{\vdash} L$[1], is a finite sequence $L_1, \ldots, L_n = L$ of ground literals satisfying the following conditions:*

1. *L_1 is a fact or a presumption;*
2. *L_i $(1 < i \leq n)$ is a fact or a presumption; or is obtained by a rule of* strict modus ponens *(MP_s) or a rule of* defeasible modus ponens *(MP_d), both possibly with a rule of* argument from strict analogy *(AA_s) or a rule of* argument from defeasible analogy *(AA_d), from a subset of ground literal L_j $(j, k, m, n < i)$ and (strict or defeasible) rules R_i in $\mathcal{P}$*

$$\frac{L_1, \ldots, L_j \qquad L_i \leftarrow L_1, \ldots, L_j}{L_i}\ MP_{\mathrm{s}}$$

$$\frac{L_1, \ldots, L_j \qquad L_i \Leftarrow L_1, \ldots, L_j, not\ L_k, \ldots, not\ L_m}{L_i}\ MP_{\mathrm{d}}$$

$$\frac{A_1 \overset{x_1}{\leftleftarrows} L_1\ \ldots\ A_k \overset{x_k}{\leftleftarrows} L_k \qquad L_i \leftarrow L_1, \ldots, L_k, \ldots, L_j}{L_i \Leftarrow A_1, \ldots, A_k, \ldots, L_j}\ \mathrm{AA}_s$$

$$\frac{A_1 \overset{x_1}{\leftleftarrows} L_1\ \ldots\ A_k \overset{x_k}{\leftleftarrows} L_k \qquad L_i \Leftarrow L_1, \ldots, L_k, \ldots, L_j, not\ L_m, \ldots, not\ L_n}{L_i \Leftarrow A_1, \ldots, A_k, \ldots, L_j, not\ L_m, \ldots, not\ L_n}\ \mathrm{AA}_d;$$

3. *$w = \bigotimes_{A_i \overset{x_i}{\leftleftarrows} L_i} \{x_i\}$, where $\bigotimes$ is a triangular norm (t-norm) and $A_i \overset{x_i}{\leftleftarrows} L_i$ is used to derive L. Otherwise, we set $w = 1$ as the default value.*

Basically, $\mathcal{P} \overset{1}{\vdash} L$ means that L may be derived without any use of analogies. Condition (2) of Definition 7 assumes an inference rule for conjoining strong literals. Since this work employs the notion of t-norm, we include its definition here for self-containment. A function $\otimes : [0,1]^2 \to [0,1]$ is called a t-norm iff it fulfills the following properties for all $x, y, z, w \in [0,1]$: **(1)** $x \otimes y = y \otimes x$ (commutativity); **(2)** $x \leq z$ and $y \leq w \Rightarrow x \otimes y \leq z \otimes w$ (monotonicity); **(3)** $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ (associativity); **(4)** $x \otimes 1 = z$ (identity). A t-norm is called *bounded* iff $x \otimes y = 0 \Rightarrow x = 0$ or $y = 0$. There are several reasons for the use of a t-norm. Firstly, it is the generalization of the conjunction in propositional logic. Secondly, the operator *min* (i.e. $x \otimes y = min\{x, y\}$) is an instance of a bounded t-norm. This reflects an intuition on the use of analogical reasoning that the strength of a consequence depends upon the use of similarities. Lastly, 1 acts as the neutral element for t-norms. Some other instances of the operator include *product* (i.e. $x \otimes y = x \cdot y$) and *hamacher product* (i.e. $x \otimes y = 0$ if $x = y = 0$ or $\frac{x \cdot y}{x + y - x \cdot y}$ if otherwise).

[1] When $w = 1$, we may remove it.

Example 3. (Continuation of Example 1) We translate the Silkwood case into our logic program $\mathcal{P}$ as follows. The literal *exception(X)* means *X is an exception to inactivate the goal*. To avoid confusion, we separate each rule by a semicolon. That is, we have a strict rule:

$\mathsf{SR} = \{$*danger(X)* $\leftarrow$ *lion(X)*; *plutonium_plant(kerr_mcgee)*, $defendant(kerr\_mcgee)\}$; three defeasible rules:

$\mathsf{DR} = \{$*liable(X)* $\Leftarrow defendant(X), danger(X),$ *not exception(X)*$\}$;
and a similarity rule:

$\mathsf{SIM} = \{$*plutonium_plant* $\overset{0.8}{\leftleftarrows}$ *lion*$\}$[1].

Assume we define $\bigotimes$ as the *min* operator, we have $\mathcal{P} \overset{0.8}{\vdash}$ *liable(kerr_mcgee)*. □

3.2 Structured Argument

We are ready to extend the notion of derivation (Definition 7) for constructing arguments. Intuitively, an argument structure is a minimal consistent set of defeasible rules and similarity rules, obtained from a derivation for a literal Q. This representation is adapted from the concept of Defeasible Logic Programming [8], where inconsistency through strict rules can be discovered (see a small discussion inside Section 4).

Definition 8 (Argument). *Let $\mathcal{P} = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ be a logic program and Q be a ground literal. An argument $\mathcal{A}$ for Q has a structure $\langle \mathsf{D}, \mathsf{S}, Q, w \rangle$ where $\mathsf{D} \subseteq \mathsf{DR}$ and $\mathsf{S} \subseteq \mathsf{SIM}$ such that:*

1. *There exists a derivation for Q from $\langle \mathsf{SR}, \mathsf{D}, \mathsf{S} \rangle$, i.e. $\langle \mathsf{SR}, \mathsf{D}, \mathsf{S} \rangle \overset{w}{\vdash} Q$;*
2. *$\langle \mathsf{SR}, \mathsf{D}, \mathsf{S} \rangle$ is consistent, i.e. $\langle \mathsf{SR}, \mathsf{D}, \mathsf{S} \rangle \overset{w}{\nvdash} A, \overline{A}$ for some ground literal A;*
3. *$\mathcal{A}$ is minimal, i.e. there is no $\mathsf{D}' \cup \mathsf{S}' \subseteq \mathsf{D} \cup \mathsf{S}$ such that $\langle \mathsf{D}', \mathsf{S}', Q, w \rangle$.*

By distinguishing a set of defeasible rules of DR and a set of similarity rules of SIM in an argument's structure, we can clearly identify analogical arguments (or arguments from analogy) apart from standard arguments, i.e. for any arguments $\mathcal{A} = \langle \mathsf{D}, \mathsf{S}, Q, w \rangle$, $\mathcal{A}$ is called an *argument from analogy* (or *analogical argument*) if $\mathsf{S} \neq \emptyset$ and is called a (*standard*) argument if otherwise. For any arguments $\mathcal{A}$, $\mathcal{A}$ is called a *strict argument* if $\mathsf{D} = \mathsf{S} = \emptyset$ and is called a *defeasible argument* if otherwise. For example, $\mathcal{A}_1 = \langle \emptyset, \mathsf{S}_1, Q_1, 0.8 \rangle$, where $\mathsf{S}_1 = \{$*plutonium_plant* $\overset{0.8}{\leftleftarrows}$ *lion*$\}$ and Q_1 = *danger(kerr_mcgee)*, is an argument from analogy. On the other hand, $\mathcal{A}_2 = \langle \mathsf{D}_1, \emptyset,$ *federal_law(kerr_mcgee)*$, 1 \rangle$, where $\mathsf{D}_1 = \{$*federal_law(X)* $\Leftarrow$ *plutonium_plant(X)*$\}$, is a defeasible argument. Observe that an argument from analogy is also a defeasible argument (such as $\mathcal{A}_1$). Condition (1) of Definition 8 conforms to our intuition that an argument $\mathcal{A}$ is a derivation for a literal Q. Condition (2) of Definition 8 spells out that an argument does not contain contradiction whereas Condition (3) says that an argument does not contain irrelevant information to derive Q.

[1] We may employ the notion of $\overset{\pi}{\sim}$ to yield the number 0.8.

Definition 9. *Let* $\mathcal{P} = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ *be a logic program and* $\mathcal{A}_1 = \langle \mathsf{D}_1, \mathsf{S}_1, Q_1, w_1 \rangle$, $\mathcal{A}_2 = \langle \mathsf{D}_2, \mathsf{S}_2, Q_2, w_2 \rangle$ *be arguments. Then, we say that* $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *iff one of the following conditions hold:*

1. $\langle \mathsf{SR}, \mathsf{D}_1, \mathsf{S}_1 \rangle \overset{w_1'}{\vdash} A$ *and* $\langle \mathsf{SR}, \mathsf{D}_2, \mathsf{S}_2 \rangle \overset{w_2'}{\vdash} \overline{A}$ *for some ground literal A;*
2. $\langle \mathsf{SR}, \mathsf{D}_1, \mathsf{S}_1 \rangle \overset{w_1'}{\vdash} A$ *and there exists* $r \in \mathsf{D}_2$ *which contains* not A *in the body;*

Definition 9 does not include ways of checking which arguments are better. It only says which arguments are in conflict. The concept of *attack* is important because it is a standard requirement of defeasible argumentation systems. We illustrate this by the following example.

Example 4. (Continuation of Example 3) Let us enrich the example that $\mathsf{SIM} = \{plutonium\_plant \overset{0.8}{\leftrightarrows} lion;\ plutonium\_plant \overset{0.4}{\leftrightarrows} wind\_turbine\}$.
and $\mathsf{DR} = \{liable(X) \Leftarrow defendant(X), danger(X), not\ exception(X);$
$federal\_law(X) \Leftarrow plutonium\_plant(X);\ \neg liable(X) \Leftarrow federal\_law(X)\}$;

We have $\mathcal{A}_1 = \langle \mathsf{D}_1, \mathsf{S}_1, liable(kerr\_mcgee), 0.8 \rangle$, where
$\mathsf{D}_1 = \{liable(X) \Leftarrow defendant(X), danger(X), not\ exception(X)\}$ and
$\mathsf{S}_1 = \{plutonium\_plant \overset{0.8}{\leftrightarrows} lion\}$, as an argument from analogy.

We also have $\mathcal{A}_2 = \langle \mathsf{D}_2, \emptyset, \neg liable(kerr\_mcgee), 1 \rangle$, where $\mathsf{D}_2 = \{federal\_law(X) \Leftarrow plutonium\_plant(X);\ \neg liable(X) \Leftarrow federal\_law(X)\}$, as a defeasible argument.

We also have $\mathcal{A}_3 = \langle \mathsf{D}_3, \mathsf{S}_3, \neg liable(kerr\_mcgee), 0.4 \rangle$, where
$\mathsf{D}_3 = \{federal\_law(X) \Leftarrow plutonium\_plant(X);\ \neg liable(X) \Leftarrow federal\_law(X)\}$ and
$\mathsf{S}_3 = \{plutonium\_plant \overset{0.4}{\leftrightarrows} wind\_turbine\}$, as another argument from analogy.

Then, we have that $\mathcal{A}_1$ attacks $\mathcal{A}_2$ and $\mathcal{A}_2$ also attack $\mathcal{A}_1$ by Condition (1). In addition, $\mathcal{A}_1$ attacks $\mathcal{A}_3$ and $\mathcal{A}_3$ also attacks $\mathcal{A}_1$ by Condition (1).

□

The argument $\mathcal{A}_1$ attacks $\mathcal{A}_2$ since it satisfies the first condition, i.e. $\langle \mathsf{SR}, \mathsf{D}_1, \mathsf{S}_1 \rangle \overset{w_1'}{\vdash} liable(kerr\_mcgee)$ and $\langle \mathsf{SR}, \mathsf{D}_2, \mathsf{S}_2 \rangle \overset{w_2'}{\vdash} \neg liable(kerr\_mcgee)$. This way of attack is called *rebuttal*. Basically, a rebuttal attacks an argument by drawing the complement of a derived literal. On the other hand, the second condition is called *undercut*. An undercut attacks by showing an exceptional situation without drawing the complement of a literal. You may observe that the definition also allows to indirectly attack an argument by either drawing the complement of an intermediate derived literal (i.e. *indirect rebuttal*) or showing an intermediate exceptional situation (i.e. *indirect undercut*).

Formalizing the scheme may need more subtle ways of comparing arguments. This is because Argument from Analogy imposes some specialties for adjudicating between conflicting arguments. We discuss about these in order. Firstly, if

an argument rebuts another argument in which both do not use similarity rules, we need some preference criteria to compare which one is better. Such criteria can be defined as a relation $>\subseteq \mathsf{DR} \times \mathsf{DR}$ in a standard way, i.e. $r_1 > r_2$ means r_1 is preferred over r_2 for any $r_1, r_2 \in \mathsf{DR}$. Then, we say an argument $\mathcal{A}_1 = \langle \mathsf{S}_1, \mathsf{D}_1, Q_1, w_1 \rangle$ is better than $\mathcal{A}_2 = \langle \mathsf{S}_2, \mathsf{D}_2, Q_2, w_2 \rangle$ (denoted by $\mathcal{A}_1 \succ \mathcal{A}_2$) if **(1)** $\exists r_1 \in \mathsf{D}_1 . \exists r_2 \in \mathsf{D}_2 : r_1 > r_2$; and **(2)** $\forall r_1 \in \mathsf{D}_1 . \forall r_2 \in \mathsf{D}_2 : r_2 \not> r_1$. We assume that preference criteria are given in this work. Secondly, if an argument rebuts another argument in which both contain similarity rules, we need to compare their analogy degree w. In this case, we respect more on an argument containing the higher w. The third case is more subtlety. If an argument from analogy rebuts a defeasible argument, we reason this based on the scheme Argument from Analogy, i.e. an argument from analogy is always preferred. Like the closing argument of Spence (cf. Example 1), he emphasizes the statement *If the lion got away, Kerr-McGee has to pay.* Uttering an instance of Argument from Analogy like this in a dialogue creates a presumption in favor of the conclusion and a corresponding proof for the other side in the dialogue to defeat this presumption by asking critical questions. This is also similar to [9, 10] who bases the set of defeasible inference rules on general cognitive and epistemological principles, such as principles of perception, memory and so on. The forth case is also special. Here, we prevent defeasible arguments to attack an argument from analogy. As indicated earlier (cf. Subsection 2.1), reasoning by Argument from Analogy use arguments from counter-analogy to attack arguments from analogy. For the fifth case, we impose that an argument from analogy may undercut another argument from analogy if the undercutter has the higher analogy degree. Otherwise, it proceeds undercutting without special conditions. This is consistent with real use of analogy, i.e. analogical arguments may be undercut if those arguments contain fault assumptions. For instance, Spence's arguments also include the following statement: *You have to pay because it was your lion – unless the person who was hurt let the lion out himself.* We formulate these phenomena in the following definition.

Definition 10. *Let* $\mathcal{A}_1 = \langle \mathsf{D}_1, \mathsf{S}_1, Q_1, w_1 \rangle$ *and* $\mathcal{A}_2 = \langle \mathsf{D}_2, \mathsf{S}_2, Q_2, w_2 \rangle$ *be two arguments. Then,* $\mathcal{A}_1$ *defeats* $\mathcal{A}_2$ *iff one of the following holds:*

1. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (1) of Definition 9,* ($\mathsf{S}_1, \mathsf{S}_2 = \emptyset$)*,* $\mathcal{A}_2 \not\succ \mathcal{A}_1$*, and* $\mathcal{A}_2$ *does not attack* $\mathcal{A}_1$ *under Condition (2) of Definition 9;*
2. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (1) of Definition 9,* ($\mathsf{S}_1, \mathsf{S}_2 \neq \emptyset$)*,* $w_1 \geq w_2$*, and* $\mathcal{A}_2$ *does not attack* $\mathcal{A}_1$ *under Condition (2) of Definition 9;*
3. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (1) of Definition 9,* $\mathsf{S}_1 \neq \emptyset$*,* $\mathsf{S}_2 = \emptyset$*, and* $\mathcal{A}_2$ *does not attack* $\mathcal{A}_1$ *under Condition (2) of Definition 9;*
4. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (1) of Definition 9,* ($\mathsf{D}_1, \mathsf{S}_1 = \emptyset$) *and* $\mathsf{S}_2 \neq \emptyset$*;*
5. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (2) of Definition 9,* ($\mathsf{S}_1, \mathsf{S}_2 \neq \emptyset$) *and* $w_1 \geq w_2$*;*
6. $\mathcal{A}_1$ *attacks* $\mathcal{A}_2$ *under Condition (2) of Definition 9 and* ($\mathsf{S}_1 = \emptyset$ *or* $\mathsf{S}_2 = \emptyset$)*.*

We say that $\mathcal{A}_1$ *strictly defeats* $\mathcal{A}_2$ *iff* $\mathcal{A}_1$ *defeats* $\mathcal{A}_2$ *and* $\mathcal{A}_2$ *does not defeat* $\mathcal{A}_1$*.*

We note that many researchers in the area of argumentation with priority have defined many ways to compare arguments. Addressing this issue is outside the scope and is irrelevant to Condition 1 of Definition 10. Hence, we make the treatment as simple as defined above, i.e. $\mathcal{A}_1 \succ \mathcal{A}_2$ if **(1)** $\exists r_1 \in \mathsf{D}_1.\exists r_2 \in \mathsf{D}_2 : r_1 > r_2$; and **(2)** $\forall r_1 \in \mathsf{D}_1.\forall r_2 \in \mathsf{D}_2 : r_2 \not> r_1$, and leave this as a future task.

Example 5. (Continuation of Example 4) We have that $\mathcal{A}_1$ *strictly* defeats $\mathcal{A}_2$ and also *strictly* defeats $\mathcal{A}_3$.

□

Theorem 1. *There does not exist an argument which attacks an argument* $\mathcal{A} = \langle \emptyset, \emptyset, Q, w \rangle$*, where* Q *is a ground literal and* $w \in (0, 1]$*.*

Proof. *Let* $\mathcal{P} = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ *be a logic program and suppose that there exists an argument* $\mathcal{B} = \langle \mathsf{D}_b, \mathsf{S}_b, Q_b, w_b \rangle$ *which attacks* $\mathcal{A}$*. By Definition 9, we show contradictory by cases:*

- *(Rebuttal) We have* $\langle \mathsf{SR}, \mathsf{D}_b, \mathsf{S}_b \rangle \overset{w_b}{\vdash} A$ *and* $\langle \mathsf{SR}, \emptyset, \emptyset \rangle \overset{w_a}{\vdash} \overline{A}$ *for some ground literal* A*. This means* $\langle \mathsf{SR}, \mathsf{D}_b, \mathsf{S}_b \rangle$ *is inconsistent and* $\mathcal{B}$ *is not an argument.*
- *(Undercut) We have* $\langle \mathsf{SR}, \mathsf{D}_b, \mathsf{S}_b \rangle \overset{w_b}{\vdash} A$ *and there exists* $r \in \emptyset$ *which contains* not A *in the body. This case is trivial.*

❑

Corollary 1. *Strict arguments always* strictly defeat *defeasible arguments, including arguments from analogy.*

Proof. *This immediately follows from Theorem 1 and Definition 10.* ❑

Corollary 1 exhibits that any conclusions drawn by analogical reasoning cannot strictly defeat conclusions drawn by the use of only strict rules. This shows that our system conforms to how the legal (and similar) uses analogy in reasoning. Spence's closing argument works out because what he reasons does not conflict with the state law.

3.3 Justification through Dialectical Analysis

In this work, we base our semantics on the grounded semantics of Dung's framework [3]. A sound and complete calculus under the grounded semantics has a form of the dialectical style between a proponent (P) and an opponent (O) of an argument. A proponent starts with an argument to be justified and then the players take turn. An opponent must defeat or strictly defeat a proponent's last argument while a proponent must strictly defeat opponent's last argument. Moreover, proponent is not allowed to repeat his arguments.

Definition 11 ([11]). *A dialogue is a finite nonempty sequence of moves* $move_i = \langle Player_i, \mathcal{A}_i \rangle$ *where* $i > 0$ *such that:*

1. $(Player_i = P \Leftrightarrow i \text{ is odd})$ *and* $(Player_i = O \Leftrightarrow i \text{ is even})$*;*

2. $Player_i = Player_j = P$ *and* $i \neq j \Rightarrow \mathcal{A}_i \neq \mathcal{A}_j$*;*
3. $Player_i = P$ $(i > 1) \Rightarrow \mathcal{A}_i$ *strictly defeats* $\mathcal{A}_{i-1}$*;*
4. $Player_i = O \Rightarrow \mathcal{A}_i$ *defeats* $\mathcal{A}_{i-1}$*.*

Definition 12 ([11]). *A dialogue tree is a finite tree of moves such that:*

1. *Each branch is a dialogue;*
2. *If* $Player_i = P$*, then children of* $move_i$ *are all defeaters of* $\mathcal{A}_i$*.*

A player wins a dialogue if there are no moves for another player. Furthermore, a player wins a dialogue tree iff that player wins all branches of the tree.

Definition 13. *An argument* $\mathcal{A} = \langle \mathsf{D}, \mathsf{S}, Q, w \rangle$ *is a* justified argument *iff there exists a dialogue tree which* $\mathcal{A}$ *appears at the root and is won by the proponent. If* $\mathcal{A}$ *is justified, then* Q *is called a* justified conclusion *of* $\mathcal{A}$ *and the degree of proof of* Q*, denoted by* $|Q|$*, is defined as follows:*

$$|Q| = \bigoplus_{\mathcal{A}_i = \langle \mathsf{D}_i, \mathsf{S}_i, Q_i, w_i \rangle} \{w_i\}, \textit{ for all } \mathcal{A}_i \textit{ of each } P \textit{ along the tree and}$$

an accumulator $\bigoplus : [0,1]^n \rightarrow [0,1]$ *holds the following properties where* n *is the cardinality of the set* $\{w_i\}$*:*

- *Identity closed:* $\forall a, \ldots, z \in \{w_i\} : \bigoplus\{a, \ldots, z\} = 1 \Leftrightarrow a = \cdots = z = 1$*;*
- *Monotonicity:* $\forall a, \ldots, z, a', \ldots, z' \in \{w_i\} : a \leq a' \wedge \cdots \wedge z \leq z' \Rightarrow \bigoplus\{a, \ldots, z\} \leq \bigoplus\{a', \ldots, z'\}$*.*

Theorem 2. *If an argument* $\mathcal{A} = \langle \mathsf{D}, \mathsf{S}, Q, w \rangle$ *is justified and its dialogue tree does not use arguments from analogy, then* $|Q| = 1$*.*

Let us illustrate this based on our motivative example. In the following, we define $\bigotimes$ as the *min* operator and $\bigoplus$ as the *arithmetic mean*.

Example 6. (Continuation of Example 5) Let us note here that We have $\mathcal{A}_1 = \langle \mathsf{D}_1, \mathsf{S}_1, \textit{liable(kerr\_mcgee)}, 0.8 \rangle$, where
$\mathsf{D}_1 = \{\textit{liable(X)} \Leftarrow \textit{defendant}(X), \textit{danger}(X), \textit{not exception(X)}\}$ and
$\mathsf{S}_1 = \{\textit{plutonium\_plant} \overset{0.8}{\leftleftarrows} \textit{lion}\}$, as an argument from analogy.
We also have $\mathcal{A}_2 = \langle \mathsf{D}_2, \emptyset, \neg \textit{liable(kerr\_mcgee)}, 1 \rangle$, where $\mathsf{D}_2 = \{\textit{federal\_law(X)} \Leftarrow \textit{plutonium\_plant(X)}; \neg \textit{liable(X)} \Leftarrow \textit{federal\_law(X)}\}$, as a defeasible argument.
We also have $\mathcal{A}_3 = \langle \mathsf{D}_3, \mathsf{S}_3, \neg \textit{liable(kerr\_mcgee)}, 0.4 \rangle$, where
$\mathsf{D}_3 = \{\textit{federal\_law(X)} \Leftarrow \textit{plutonium\_plant(X)}; \neg \textit{liable(X)} \Leftarrow \textit{federal\_law(X)}\}$ and
$\mathsf{S}_3 = \{\textit{plutonium\_plant} \overset{0.4}{\leftleftarrows} \textit{wind\_turbine}\}$.

To show that *liable(kerr_mcgee)* is logically entailed, we show a dialectical tree as follows. Also, $|\textit{liable(kerr\_mcgee)}| = \frac{0.8}{1} = 0.8$.

$$\langle P, \mathcal{A}_1 \rangle$$

□

Ones may be curious if our system is easily influenced in a sense that all similarity rules are always relevant to play in the game. Of course, they are not always relevant because we would not like to draw conclusions from two predicates which have less similarity. In other words, the strength of analogical arguments come from the similarity of two predicates used to derive conclusions. Hence, our input information will also be a *relevant degree* s where $s \in (0, 1]$. We capture this intuitive idea in the following definition.

Definition 14 (Analogical Reasoner). *An analogical reasoner* $\mathcal{R}$ *is a pair* $\langle \mathcal{P}, s \rangle$ *where* $\mathcal{P} = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ *is a logic program and* $s \in (0, 1]$ *is a relevant degree. An argument* $\mathcal{A}$ *is* justified by $\mathcal{P}$ under s *iff* $\mathcal{A}$ *is a justified argument of* $\mathcal{P}'$ *where* $\mathcal{P}' = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM}' \rangle$ *and* $\mathsf{SIM}' = \{L_0 \overset{x}{\Leftarrow} L_1 \subseteq \mathsf{SIM} : x \geq s\}$.

4 Related Work

After surveying the literature on argument from analogy in many fields, such as logic, law, philosophy of science, and computer science, it appears to us that there are two different forms of Argument from Analogy. The first form (cf. Subsection 2.1), in which our work is based, is the most widely accepted version whereas the second one compares factors of two cases (such as [12, 13]), which may regard as an instance of the first form. As the second one makes no reference to the notion of similarity, it becomes simpler to use, such as in standard case-based reasoning. The method of evaluating an analogical argument in case-based reasoning (CBR) uses respects (called dimensions and factors) in which two cases are similar or different. In CBR, the decision in the best precedent case is then taken as the decision into the current case. A dimension is a relevant aspect of the case whereas a factor is an argument favoring one side or the other in relation to the issue being disputed. The HYPO system [14] uses dimensions. CATO [15] is a simpler CBR system that uses factors. Systems which employ factors use pro factors to represent similarities for supporting an argument whereas con factors representing dissimilarity to undermine the argument. Factors may be weighted. To do this numerically, these systems have to attach a positive number or negative number to each factor. There is another approach which is seen as dialectical.

As an example of the dialectical approach, HYPO evaluates analogical arguments in a three-step method called *three-ply argumentation*, which can be considered as a series of moves in a formal dialogue. In the first move, the proponent puts forward an analogical argument by finding a comparable past case in which the outcome closely matches that of the proponent's thesis. In the second move, the respondent can reply to the original argument in one of the following ways, i.e. by **(1)** finding a past case that matches the current case having the opposite outcome; or, **(2)** pointing to factors presented in the new case (but not in the previous case). In the third move, the proponent may reply in one of the following ways: by **(1)** pointing out factors; or **(2)** citing other cases to show that the respondent's attack does not really rebut the original argument. Unfortunately, the system is semi-formalized. Later, these systems are reanalyzed and

reformalized in terms of the formal argumentation framework called ASPIC+ [16] for legal case-based reasoning, which also captures argumentation schemes.

In contrast with these systems, our approach formalizes the first form of Argument from Analogy. Instead of defining similarity via notions of respects, we explicitly state similarity of two cases via similarity rules. Doing so creates a more general framework for analogical reasoning since it enables integration with external systems for taking advantage of existing similarity measures. For example, it is possible to define cases in terms of DLs' terminology and employ the notion $\overset{\pi}{\sim}$ (cf. Subsection 2.2) to yield similarity degree of two cases.

Another important difference between this work and ASPIC+ is arguments' structure whose structure of arguments is represented by a proof tree. Consider a program $\mathcal{P} = \langle \mathsf{SR}, \mathsf{DR}, \mathsf{SIM} \rangle$ where $\mathsf{SR} = \{a \leftarrow b; \neg a \leftarrow b\}$, $\mathsf{DR} = \{b \Leftarrow\}$, and $\mathsf{SIM} = \emptyset$. In approaches which constructs an argument from a proof tree, b can be acceptable. However, the literal b together with $a \leftarrow b$ and $\neg a \leftarrow b$ can derive inconsistency. Inspired by Defeasible Logic Programming [8], this work can discover this kind of inconsistency through strict rules.

Lastly, we discuss on some systems which adopt Argument from Analogy in practices. As an application of Defeasible Logic Programming, [17] finds relationships from the database and build analogies; after that, implements a reasoner on top of a Defeasible Logic Programming interpreter. On the other hand, [18] employs the second form of Argument from Analogy to predict new assertions which cannot logically be entailed by standard DLs reasoner.

5 Conclusion

In this paper, we make contributions to the logical study of analogical reasoning under the lens of structured argumentation. The key idea behinds our approach lies in the scheme Argument from Analogy proposed by Walton [1]. As a result, this work provides the possibility of representing information in the form of strict, defeasible, and similarity rules in a declarative manner. To the best of our knowledge, this is the first proposal of analogical reasoning framework in the field of argumentation and Logic Programming. The proposed framework enables the integration with external similarity measures; and, is also flexible to be tuned by users. That is, instead of fixing specific operators for aggregating the degree, we identify required properties for them. Hence, different instances of our generic operators, viz. $\bigotimes$ and $\bigoplus$, can be used, depending on target cases. Finally, we note that our focuses in this paper is on the formalization of Argument from Analogy. The structure of arguments proposed in this paper is adapted from Defeasible Logic Programming to increase the potential of its implementation. Currently, we are under an implementation aiming to integrate the proposed framework with our previous work (cf. Subsection 2.2) in DLs. The notion $\overset{\pi}{\sim}$ will be used to determine similarity of two predicates.

As we adapt the concept of Defeasible Logic Programming for constructing arguments, it would also be interesting for us to study the phenomena in other

formalisms of logic program, such as Answer Set Programming, and extended frameworks of the abstract argumentation. We leave this for our future work.

Acknowledgments

This research is part of the JAIST-NECTEC-SIIT dual doctoral degree program; and is partly supported by CILS of Thammasat University and the NRU project of Thailand Office of Higher Education Commission

References

1. Walton, D.: Argumentation schemes for argument from analogy. Systematic Approaches to Argument by Analogy (2014) 23–40
2. Lief, M.S., Caldwell, H.M., Bycel, B.: Ladies And Gentlemen Of The Jury: Greatest Closing Arguments In Modern Law. Scribner (2000)
3. Aleven, V.: Teaching case-based argumentation through a model and examples (1997)
4. Ashley, K. In: Case-based reasoning. Springer Netherlands, Dordrecht (2006) 23–60
5. Paola D. Budán, Federico Rosenzvaig, M.C.D.B., Simari, G.R.: Recommender system based on argumentation by analogy. In: Proceedings of Congreso Argentino de Ciencias de la Computación. (2014)
6. d'Amato, C., Fanizzi, N., Esposito, F. In: Analogical Reasoning in Description Logics. Springer Berlin Heidelberg, Berlin, Heidelberg (2008) 330–347
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2) (1995) 321–358
8. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press (2008)
9. Racharak, T., Suntisrivaraporn, B.: Similarity measures for $\mathcal{FL}_0$ concept descriptions from an automata-theoretic point of view. In: Information and Communication Technology for Embedded Systems (IC-ICTES), 2015 6th International Conference of. (March 2015) 1–6
10. Racharak, T., Suntisrivaraporn, B., Tojo, S. In: Identifying an Agent's Preferences Toward Similarity Measures in Description Logics. Springer International Publishing, Cham (2016) 201–208
11. Racharak, T., Suntisrivaraporn, B., Tojo, S.: sim^{π}: A concept similarity measure under an agent's preferences in description logic $\mathcal{ELH}$. In: Proceedings of the 8th International Conference on Agents and Artificial Intelligence. (2016) 480–487
12. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory Pract. Log. Program. **4**(2) (January 2004) 95–138
13. Pollock, J.L.: Defeasible reasoning. Cognitive Science **11** (1987) 481–518
14. Pollock, J.L.: Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press, Cambridge, MA, USA (1995)
15. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-classical Logics **7** (1997) 25–75
16. Hurley, P.J.: A concise introduction to logic. Belmont: Wadsworth (2003)
17. Copi, I.M., Cohen, C.C.: Introduction to logic. New York: Macmillan (1990)
18. Prakken, H., Wyner, A.Z., Bench-Capon, T.J.M., Atkinson, K.: A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. J. Log. Comput. **25**(5) (2015) 1141–1166

Dischargeable Obligations in ALP: A case study from the Japanese Civil Code

Marco Alberti[1], Marco Gavanelli[2], Evelina Lamma[2], Fabrizio Riguzzi[1], and Riccardo Zese[2]

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

Abstract. Abductive Logic Programming (ALP) has been proven very effective for formalizing societies of agents, commitments and norms. In [5] and [4], the most common deontic operators (obligation, prohibition, permission) were mapped into the abductive expectations of an ALP framework, where operators are also enriched with quantification over time, by means of ALP and Constraint Logic Programming (CLP).
In our previous work presented at JURISIN2015 workshop, we have shown that ALP is a suitable framework for representing both norms and ontologies, taking advantage of the ALP representation for Datalog$^{\pm}$ ontologies introduced in [19, 20]. Normative reasoning and ontological query answering were then obtained by applying the same abductive proof procedure, smoothly achieving their integration. In particular, we considered the ALP framework named $\mathcal{S}$CIFF and derived from the IFF abductive framework, able to deal with existentially (and universally) quantified variables in rule heads and CLP constraints.
In this work, we introduce a defeasible flavour in this framework, in order to possibly discharge obligations in some scenarios. Abductive expectations can be also qualified as dischargeable, in the new, extended syntax. Both declarative and operational semantics are improved accordingly, and proof of soundness is given under syntax allowedness conditions.
We show how the extended framework, named $\mathcal{S}$CIFF$^{\mathcal{D}}$, can be exploited to model and reason upon a fragment of the Japanese Civil Code, and how it is a better tool for legal modeling, and its operational support suitable for legal reasoning. We consider a case study concerning manifestations of intention and their rescission (Section II of the Japanese Civil Code).

1 Introduction

A normative system is a set of norms encoded in a formal language, together with mechanisms to reason about, apply, and modify them. Since norms preserve the autonomy of the interacting parties (which ultimately decide whether or not to follow the norms), normative systems are an appropriate tool to regulate

interaction in multi-agent systems [10]. Usually, norm definitions build upon some notion of obligation, permission and prohibition, in the tradition of Deontic Logic [34].

When formalizing norms, a natural approach is to encode them as implications; semantically, implications naturally represent conditional norms, where the antecedent is read as a property of a state of affairs and the consequent as its deontic consequence, and operationally, rule-based systems offer support for reasoning and drawing conclusions from norms and a description of the system they regulate. Applications of computational logic to formalize norms include logic programming for the British Nationality Act [32]; argument-based extended logic programming with defeasible priorities [29], defeasible logic [22]. Satoh et al.'s PROLEG is a Prolog implementation of the Presupposed Ultimate Fact Theory of the Japanese Civil Code [31]. The contract cancellation under the Japanese law was also formalized in computational logics [12].

As mentioned above, normative systems have been applied in multi-agent systems [10]. Among the organizational models, [13, 15, 14] exploit Deontic Logic to specify the society norms and rules. Significant portions of EU research projects were devoted to formalizing norms for multiagent systems; namely, the ALFEBITE project [8] was focused on the formalization of an open society of agents using Deontic Logic, and in the IMPACT project [9, 17] an agent's obligations, permissions and prohibitions are specified by corresponding deontic operators.

The EU IST Project SOCS proposed different Abductive Logic Programming (ALP) languages and proof procedures to specify and implement both individual agents [11] and their interaction [2]; both approaches have later been applied to model and reason about norms with deontic flavours [5, 30].

ALP has been proved a powerful tool for knowledge representation and reasoning [26], taking advantage of ALP operational support as (static or dynamic) verification tool. ALP languages are usually equipped with a declarative (model-theoretic) semantics, and an operational semantics given in terms of a proof-procedure. Fung and Kowalski proposed the IFF abductive proof-procedure [18] to deal with forward rules, and with non-ground abducibles. It has been later extended [3], and the resulting proof procedure, named $\mathcal{S}$CIFF, can deal with both existentially and universally quantified variables in rule heads and CLP constraints [23]. The resulting system was used for modeling and implementing several knowledge representation frameworks, such as deontic logic [5], where the deontic notions of obligation and permission are mapped into special $\mathcal{S}$CIFF abducible predicates, normative systems [4], interaction protocols for multi-agent systems [6], Web services choreographies [1], etc.

Several efforts have been devoted to extend logic programming approaches with Description Logics, in an attempt to exploit legal ontologies for representing, processing and retrieving legal information, see for instance, [7], [33], [27, 16]).

In our JURISIN2015 work [21], we showed that ALP, and $\mathcal{S}$CIFF in particular, is a suitable framework for representing and integrating both norms and Datalog$^{\pm}$ ontologies. Normative reasoning and ontological query answering

were obtained by directly applying the $\mathcal{S}$CIFF abductive proof procedure. The main advantage was that this integration was achieved within a single language, grounded on abduction in Computational Logic.

In this work, we present $\mathcal{S}$CIFF$^{\mathcal{D}}$, an extension of the $\mathcal{S}$CIFF framework, and its declarative semantics and operational support, with a mechanism for discharging obligations: intuitively, rather than removing an abductive expectation representing obligation with a sort of contraction, we mark it as discharged, i.e., the lack of a fulfilling act is not a violation of the norms. This extension introduces a defeasible flavour in the framework. Both declarative and operational semantics are extended accordingly, and proof of soundness is given under syntax allowedness conditions.

As a running example throughout the paper, we consider a case study concerning manifestation of intention (Section II of the Japanese Civil Code), and the $\mathcal{S}$CIFF$^{\mathcal{D}}$ framework is exploited to model and reason upon this case study.

The paper is organized as follows. We first recall the $\mathcal{S}$CIFF language in Section 2, also mentioning its declarative semantics and its underlying proof procedure. Then, in Section 3, we introduce the $\mathcal{S}$CIFF$^{\mathcal{D}}$ syntax, with a novel abducible, needed for discharging obligations (namely, expectations). The case study from Section II of the Japanese Civil Code is discussed in Section 4. Section 5 and Section 6 extend the declarative and operational semantics accordingly, and state and prove soundness and completeness for the extended framework. In Section 7 we conclude the paper, and outline future work.

2 Preliminaries

In this section, we provide an informal description of the $\mathcal{S}$CIFF language; for formal definitions, we refer the reader to [5].

The $\mathcal{S}$CIFF language. A $\mathcal{S}$CIFF program is a pair $\langle KB_S, \mathcal{IC}_S \rangle$, where KB_S is a set of logic programming clauses (used to express domain specific knowledge) and $\mathcal{IC}_S$ is a set of implications called Integrity Constraints, which implicitly define the expected behaviour of the interacting parties.

In particular, such behaviour is described by means of events (actual behaviour) and expectations (expected behaviour):

- events are atoms of the form $\mathbf{H}(\mathit{Content}, \mathit{Time})$
- expectations are abducible atoms of the following possible forms:
  - $\mathbf{E}(\mathit{Content}, \mathit{Time})$: positive expectations, with a deontic reading of obligation [5];
  - $\mathbf{EN}(\mathit{Content}, \mathit{Time})$: negative expectations, read as prohibition
  - $\neg\mathbf{E}(\mathit{Content}, \mathit{Time})$: negation of positive expectation, or explicit absence of obligation
  - $\neg\mathbf{EN}(\mathit{Content}, \mathit{Time})$: negation of negative expectation, or explicit permission

where *Content* is a logic term that describes the content and *Time* is a variable or a term representing the time of the event. CLP constraints can be imposed over variables; in the case of time variables, they represent time constraints, such as deadlines.

Each Integrity Constraint (IC) in $\mathcal{IC}_S$ has the form $Body \rightarrow Head$, where $Body$ is a conjunction of atoms defined in KB_S, events and expectations, while $Head$ is a disjunction of conjunctions of expectations.

Declarative semantics. The abductive semantics of the $\mathcal{S}$CIFF language defines, given a set **HAP** of **H** atoms called history and representing the actual behaviour, an abductive answer, i.e., a ground set **EXP** of expectations that

- together with the history and KB_S, entails $\mathcal{IC}_S$:

$$KB_S \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{IC}_S \tag{1}$$

- is consistent with respect to explicit negation; for given *Content* and *Time*, it cannot include $\{\mathbf{E}(Content, Time), \neg\mathbf{E}(Content, Time)\}$ or $\{\mathbf{EN}(Content, Time), \neg\mathbf{EN}(Content, Time)\}$:

$$\begin{array}{c}\{\mathbf{E}(Content, Time), \neg\mathbf{E}(Content, Time)\} \nsubseteq \mathbf{EXP} \wedge \\ \wedge \{\mathbf{EN}(Content, Time), \neg\mathbf{EN}(Content, Time)\} \nsubseteq \mathbf{EXP}\end{array} \tag{2}$$

- is consistent with respect to the meaning of expectations (for given *Content* and *Time*, it cannot include $\{\mathbf{E}(Content, Time), \mathbf{EN}(Content, Time)\}$)

$$\{\mathbf{E}(Content, Time), \mathbf{EN}(Content, Time)\} \nsubseteq \mathbf{EXP} \tag{3}$$

- is fulfilled by the history, i.e.

$$\text{if } \mathbf{E}(Content, Time) \in \mathbf{EXP} \text{ then } \mathbf{H}(Content, Time) \in \mathbf{HAP} \text{ and} \tag{4}$$

$$\text{if } \mathbf{EN}(Content, Time) \in \mathbf{EXP} \text{ then } \mathbf{H}(Content, Time) \notin \mathbf{HAP} \tag{5}$$

Operational Semantics. Operationally, the $\mathcal{S}$CIFF abductive proof procedure will find an abductive answer if one exists, or detect that no one exists (see [3] for soundness and completeness statements), meaning that the history violates the $\mathcal{S}$CIFF program. We call the two cases success and failure, respectively. The $\mathcal{S}$CIFF proof-procedure is defined through a set of transitions, each rewriting one node of a proof tree into one or more nodes. The basic transitions of $\mathcal{S}$CIFF are inherited from the IFF [18], and they account for the core of abductive reasoning. Other transitions deal with CLP constraints, and are inherited from the CLP transitions [24]. Due to the lack of space, we cannot describe in detail all the transitions; we sketch those dealing with the concept of expectation, that are most relevant for the rest of the paper.

In order to deal with the concept of expectation, in each node of the proof tree, the set of abduced expectations **EXP** is partitioned into two sets: the fulfilled (**FULF**), and pending (**PEND**) expectations.

Transition *Fulfillment* **E** deals with the fulfillment of **E** expectations: if an expectation $\mathbf{E}(E, T_E) \in \mathbf{PEND}$ and the event $\mathbf{H}(H, T_H)$ is in the current history **HAP**, two nodes are generated: one in which $E = H$, $T_E = T_H$ and $\mathbf{E}(E, T_E)$ is moved to the set **FULF**, the other in which $E \neq H$ or $T_E \neq T_H$ (where $\neq$ stands for the constraint of disunification).

Transition *Violation* **EN** deals with the violation of **EN** expectations: if an expectation $\mathbf{EN}(E, T_E) \in \mathbf{PEND}$ and the event $\mathbf{H}(H, T_H) \in \mathbf{HAP}$, the constraint $E \neq H \vee T_E \neq T_H$ is imposed, possibly leading to failure.

When there are no more relevant events, *history closure* is applied; in this case, all remaining **E** expectations in the set **PEND** are considered as violated and a failure occurs.

3 $\mathcal{S}$CIFF$^{\mathcal{D}}$ language

In legal reasoning, expectations could be discharged not only because they become fulfilled by matching the actual behaviour of the agent, but also for other reasons. For example, in case a contract is declared null, the agents are no longer expected to perform the actions required in the contract.

We introduce an extension of the $\mathcal{S}$CIFF language to deal with expectations that do not hold any longer. In the proposed extension, the user might declare that an expectation is discharged by using the functor **D**; the atom

$$\mathbf{D}(\mathbf{E}(X, T))$$

means that the expectation $\mathbf{E}(X, T)$ is no longer required to be fulfilled, as it has been discharged.

The integrity constraints can have **D** atoms, which can be abduced; the occurrence of **D** atoms is subject to the allowedness conditions stated in Def. 1 below.

Example 1. For example, the user might write an IC saying that if a contract with identifier Id_C is null (represented in this example with an abducible **NULL***, carrying the identifier of the contract and that of the reason for nullification), all expectations requiring the performance of an action in the context of that contract are discharged:*

$$\begin{aligned}\mathbf{NULL}(Id_C, Id_{null}) \wedge \mathbf{E}(do(Agent, Action, Id_C), T_{do}) \\ \rightarrow \mathbf{D}(\mathbf{E}(do(Agent, Action, Id_C), T_{do})).\end{aligned} \tag{6}$$

We can express that a contract that is explicitly permitted to be rescinded can be nullified, by performing the rescission.

$$\begin{aligned}\neg\mathbf{EN}(rescind(A, I, F, Id_I, Id_R), T_r) \wedge \mathbf{H}(rescind(A, I, F, Id_I, Id_R), T_r) \\ \rightarrow \mathbf{NULL}(Id_I, Id_R)\end{aligned} \tag{7}$$

The combined effect of ICs (6) and (7) is that rescission is only effective when the circumstances grant an explicit permission.

In order to have an efficient treatment of the **D** atoms, we restrict the syntax of ICs containing **D** atoms as follows.

Definition 1. **D**-*allowedness. An IC containing a* **D** *atom is* **D**-allowed *if the following conditions hold:*

- *there is only one* **D** *atom, it occurs in the head, and the head contains only that atom*
- *the expectation in the* **D** *atom occurs identical in the body of the IC.*

A KB_S *is* **D**-*allowed if none of its clauses contains* **D** *atoms.*

Note that IC (6) is **D**-allowed; intuitively, the given notion of **D**-allowedness lets one define ICs that select one expectation and make it discharged, subject to a number of conditions that can occur in the body.

This synctactic restriction reduces the branching in the operational semantics (see Section 6), because it makes a class of case analysis transitions, which would otherwise be required to match **E** and **D**(**E**) atoms, unnecessary.

Next section is devoted to discuss a running example concerning manifestation of intention (Section II of the Japanese Civil Code), modeled in the extended $\mathcal{S}$CIFF framework. Sections presenting the declarative and operational semantics of the extended framework and proof of soundness then follow.

4 Case study

As a running example, we take article 96 ("Fraud or duress") of the Japanese civil code (see, for example, [25]).

We describe the content of events and expectations by means of the following terms:

- $intention(A, B, I, Id_I)$: person A utters to person B a manifestation of intention, with identifier Id_I for the action I;
- $do(A, B)$: person A performs act B;
- $induce(A, B)$: act A induces act B;
- $rescind(A, B, I, F, Id_I, Id_R)$: person A rescinds, with identifier Id_R, his or her intention uttered to B and identified by Id_I to perform action I, due to fraud or duress F;
- $know(A, F)$: person A becomes aware of fact F;
- $assertAgainst(A, B, Id_R)$: person A asserts the legal act identified by Id_R against person B.

Note that legally relevant acts (namely *intention*, *rescind*, *assertAgainst*) have identifiers.

A manifestation of intention should, in general, be followed by the performance of the act.

$$\mathbf{H}(intention(A, B, I, IdI), T_1) \rightarrow \mathbf{E}(do(A, I), T_2) \wedge T_2 > T_1 \tag{8}$$

1. Manifestation of intention which is induced by any fraud or duress may be rescinded.

$$\begin{aligned}&\mathbf{H}(intention(A,B,I,Id_I),T_1)\\&\wedge\mathbf{H}(do(B,F),T_3)\\&\wedge\mathbf{H}(induce(F,I),T_2)\\&\wedge fraudOrDuress(F)\\&\wedge T_3<T_2\wedge T_2<T_1\wedge T_1<T_4\\\rightarrow&\neg\mathbf{EN}(rescind(A,B,I,F,Id_I,Id_R),T_4)\end{aligned}\tag{9}$$

2. In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.

$$\begin{aligned}&\mathbf{H}(intention(A,B,I,Id_I),T_1)\\&\wedge\mathbf{H}(do(C,F),T_3)\wedge C\neq B\\&\wedge\mathbf{H}(know(B,F),T_5)\\&\wedge\mathbf{H}(induce(F,I),T_2)\\&\wedge fraudOrDuress(F)\\&\wedge T_2<T_1\wedge T_3\leq T_5\wedge T_5<T_1\\\rightarrow&\neg\mathbf{EN}(rescind(A,B,I,F,Id_I,Id_R),T_4)\end{aligned}\tag{10}$$

3. The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.

$$\begin{aligned}&\mathbf{H}(rescind(A,B,I,F,Id_I,Id_R),T_1)\\&\wedge\mathbf{not}\ \mathbf{H}(know(C,F),T_2)\\\rightarrow&\mathbf{EN}(assertAgainst(A,C,Id_R),T_3)\wedge T_1<T_3\end{aligned}\tag{11}$$

Example 2. Let us assume that work on an acquired good G is to be paid by the good's owner O, unless O rescinds the good's purchase and asserts the rescission against the performer of the work M (which, due to integrity constraint (11), is only allowed if the performer was aware of the rescission's cause, such as a fraud). We can express this norm by means of the following integrity constraint:

$$\begin{aligned}&\mathbf{H}(work(M,G,O,W),T_1)\\\rightarrow&\mathbf{E}(pay(O,M,W),T_2)\wedge T_1<T_2\\\vee(&\mathbf{E}(rescind(O,B,buy(G),F,Id_I,Id_R),T_3)\\&\wedge\mathbf{E}(assertAgainst(O,M,Id_R),T_4)\\&\wedge T_1<T_3\wedge T_3<T_4)\end{aligned}\tag{12}$$

The KB defines what constitutes fraud or duress; namely, fixing a car's mileage.

$$fraudOrDuress(fixMileage(C))\tag{13}$$

In the scenario defined by the following history,

$$\begin{aligned}
&\mathbf{H}(do(bob, fixMileage(car)), 1)\\
&\mathbf{H}(induce(fixMileage(car), buy(car)), 2)\\
&\mathbf{H}(intention(alice, bob, buy(car), i1), 3)\\
&\mathbf{H}(work(mechanic, car, alice, lpgSystem), 4)\\
&\mathbf{H}(rescind(alice, bob, buy(car), fixMileage(car), i1, i2), 5)
\end{aligned} \tag{14}$$

Alice's rescission is explicitly permitted by IC (9), because her manifestation of intention was induced by Bob's fraudulent act of fixing the car's mileage; thanks to ICs (7) and (6), the expectation

$$\mathbf{E}(do(alice, buy(car)), T)$$

raised because of IC (8) is fulfilled even though no matching event occurs in the history.

However, since the mechanic was not aware of Bob's fraud, IC (11) prevents Alice from asserting the rescission against him, so the second disjunct in IC (12) cannot hold, and Alice still has to pay him for installing the LPG system ($\mathbf{E}(pay(alice, mechanic, lpgSystem, T_2))$*).*

Let us consider the history that contains all the events in formula (14), plus the following:

$$\mathbf{H}(know(mechanic, fixMileage(car)), 1)$$

i.e., the mechanic is now aware of the car's mileage being fixed; in this case alice would not have been prohibited from asserting the rescission against the mechanic by integrity constraint (11). By performing the following action

$$\mathbf{H}(assertAgainst(alice, mechanic, i2), 6)$$

the second disjunct in the head of integrity constraint (12) would be satisfied and alice would not be obligated to pay the mechanic for his work.

5 Declarative Semantics

We now show how to deal with the discharge of expectations in the context of ALP. We first give an intuitive definition, then show its pitfalls and finally provide a correct definition.

In order to accept histories in which expectations could not have matching events, we need to extend the definition of fulfillment of expectations given in equations (4) and (5); intuitively, a positive expectation is fulfilled if either there is a matching event or if the expectation has been discharged:

$$\begin{aligned}
\text{if}\ \ &\mathbf{E}(Content, Time) \in \mathbf{EXP}\\
\text{then}\ \ &\mathbf{H}(Content, Time) \in \mathbf{HAP} \vee \mathbf{D}(\mathbf{E}(Content, Time)) \in \mathbf{EXP}
\end{aligned} \tag{15}$$

and symmetrically for a negative expectation:

$$\begin{array}{rl} \text{if} & \mathbf{EN}(\mathit{Content}, \mathit{Time}) \in \mathbf{EXP} \\ \text{then} & \mathbf{H}(\mathit{Content}, \mathit{Time}) \notin \mathbf{HAP} \vee \mathbf{D}(\mathbf{EN}(\mathit{Content}, \mathit{Time})) \in \mathbf{EXP} \end{array} \quad (16)$$

However, in this way, there would exist abductive answers with **D** literals even if there is no explicit rule introducing them. For example, in the history of formula (14) an abductive answer would be[3]

$$\begin{array}{l} \neg\mathbf{EN}(\mathit{rescind}(\mathit{alice}, \mathit{bob}, \mathit{buy}(\mathit{car}), \mathit{fixMileage}(\mathit{car}), i1, \_), T2), \\ \mathbf{D}(\mathbf{EN}(\mathit{assertAgainst}(\mathit{alice}, \_, i2), \_)) \\ \mathbf{D}(\mathbf{E}(\mathit{do}(\mathit{alice}, \mathit{buy}(\mathit{car}), i1), \_)) \\ \mathbf{D}(\mathbf{E}(\mathit{pay}(\mathit{alice}, \mathit{mechanic}, \mathit{lpgsystem}), \_)) \\ \mathbf{NULL}(i1, i2) \end{array}$$

since it satisfies equations (1) (which takes into account knowledge base and integrity constraints), (2), (3), (15) and (16). Note that Alice is no longer required to pay the mechanic, because although no IC introduces explicitly the dischargement of the expectation that she should pay, the abductive semantics would accept the introduction of the literal

$$\mathbf{D}(\mathbf{E}(\mathit{pay}(\mathit{alice}, \mathit{mechanic}, \mathit{lpgsystem}), \_)).$$

We propose the following semantics.

Definition 2. *Abductive answer.*
If there is a set **EXP** *such that*

1. *equations (1), (2), and (3) are satisfied*
2. *is minimal with respect to set inclusion within the sets satisfying point 1, considering only* **D** *atoms; more precisely: there is no set* $\mathbf{EXP}'$ *satisfying point 1 and such that* $\mathbf{EXP}' \subset \mathbf{EXP}$ *and* $\mathbf{EXP} = \mathbf{EXP}' \cup F$*, where* F *contains only* **D** *atoms*
3. *it satisfies equations (15) and (16)*

then the set **EXP** *is an abductive answer, and we write* $\langle KB_S, \mathcal{IC}_S \rangle \models_{\mathbf{EXP}} true$.

6 Operational Semantics

Operationally, the proof procedure is extended with a new transition *Dischargement.*

If **EXP** contains two atoms $\mathbf{E}(x, T) \in \mathbf{PEND}$ and $\mathbf{D}(\mathbf{E}(x, T))$, then the atom $\mathbf{E}(x, T)$ is moved to the set **FULF** of fulfilled expectations.

Similarly, if **EXP** contains two atoms $\mathbf{EN}(x, T) \in \mathbf{PEND}$ and $\mathbf{D}(\mathbf{EN}(x, T))$, then the atom $\mathbf{EN}(x, T)$ is moved to the set **FULF** of fulfilled expectations.

[3] For brevity, we omit an expectation $\mathbf{E}(x)$ if there is already its discharged version $\mathbf{D}(\mathbf{E}(x))$.

Another modification is that transition *violation* **EN** is postponed after all other transitions (including the closure of the history). In fact, if we have $\mathbf{H}(x, 1)$, $\mathbf{EN}(x, T)$ and $\mathbf{D}(x, T)$, if the proof-procedure applied first *violation* **EN**, a failure would occur. Instead, if *Dischargement* is applied first, the expectation $\mathbf{EN}(x, T)$ is moved to the **FULF** set and transition *violation* **EN** is no longer applicable.

In case, at the end of a derivation, there are still expectations that are not fulfilled, the derivation is a failure derivation (and backtracking might occur to explore another branch, if available).

If a computation terminates with success, we write $\langle KB_S, \mathcal{IC}_S \rangle \vdash_{\mathbf{EXP}} true$.

We are now ready to give the soundness statements; these statements rely on the soundness and completeness theorems of the $\mathcal{S}$CIFF proof-procedure, so they hold in the same cases, in particular for the allowedness conditions reported in [3].

Theorem 1. *Soundness of success.*
If $\langle KB_S, \mathcal{IC}_S \rangle \vdash_{\mathbf{EXP}} true$ *then* $\langle KB_S, \mathcal{IC}_S \rangle \models_{\mathbf{EXP}} true$

Proof. If no **D** atoms occur in $\mathcal{IC}_S$, the proof procedure coincides with $\mathcal{S}$CIFF, that is sound [3].

In the case with **D**, the proof-procedure might report success in cases in which $\mathcal{S}$CIFF reports failure due to the extended notion of fulfillment (eq. 15 and 16). In such a case, the **D** atom must have been generated, and the only way to generate it is through an IC having such atom in the head (see Definition 1). The proof-procedure generates the atom only if the body of the IC is true. If the body is true, it means (from the soundness of $\mathcal{S}$CIFF) that it is true also in the declarative semantics, so also declaratively the **D** atom must be true. In such a case, eq. 15 (or 16) is satisfied, meaning that the success was sound. □

Theorem 2. *Soundness of failure.*
If $\langle KB_S, \mathcal{IC}_S \rangle \models_{\mathbf{EXP}} true$,
then $\exists \mathbf{EXP'} \subseteq \mathbf{EXP}$ *such that* $\langle KB_S, \mathcal{IC}_S \rangle \vdash'_{\mathbf{EXP}} true$

Proof. If there are no **D** atoms in $\mathcal{IC}_S$, the proof procedure coincides with $\mathcal{S}$CIFF, so the completeness theorem of $\mathcal{S}$CIFF holds [3].

In the case with **D**, the declarative semantics has as abductive answers some sets that would not have been provided by $\mathcal{S}$CIFF, and in which expectations are not fulfilled by actual events, but discharged through an abduced **D** atom.

Consider such an abductive answer **EXP**. We prove by contradiction that each **D** atom in **EXP** occurs in the head of an IC whose body is true. In fact, if a **D** atom in **EXP** was not in the head of an IC whose body is true, then the set **EXP′** obtained by removing the **D** atom from **EXP** would satisfy equation (1). On the other hand, since **EXP** satisfies equations (2) and (3) and those equations do not involve **D** atoms, also **EXP′** satisfies those equations. This means that **EXP** does not satisfy condition 2 of Definition 2, which means that **EXP** was not an abductive answer: contradiction.

Since each **D** atom occurs in the head of an IC whose body is true, the proof-procedure propagates such IC and abduces that atom. This means that

the corresponding expectation becomes discharged, and hence it does not cause failure. □

7 Conclusions and future work

In this article, we continue our line of research that applies abductive logic programming to the formalization of normative systems.

We introduced the $\mathcal{S}$CIFF$^{\mathcal{D}}$ language, which extends the $\mathcal{S}$CIFF abductive framework with the notion of dischargeable obligation, which can occurr in the head of forward rules (named ICs), fired under specific conditions mentioned in the body. The $\mathcal{S}$CIFF$^{\mathcal{D}}$ declarative semantics and its operational counterpart for verification accordingly extend $\mathcal{S}$CIFF's, and soundness is proved under syntactic conditions over these (discharging) constraints.

To experiment the framework we considered a case study requiring the notion of discharging of an obligation. In particular, we considered the articles in the Japanese Civil Code that deal with the rescission of manifestation of intentions. We also show - informally - how is the result of running the operational support upon this example in some given, simple scenarios.

In the future, we will consider lifting the **D**-allowedness restriction, allowing **D** atoms in the knowledge base and in more general integrity constraints; a preliminary assessment suggests that this extension is possible, but may be expensive in terms of computational complexity.

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Montali, M.: An abductive framework for a-priori verification of web services. In: Maher, M. (ed.) Proceedings of the Eighth Symposium on Principles and Practice of Declarative Programming. pp. 39–50. ACM Press, New York, USA (Jul 2006)
2. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based software tool. Applied Artificial Intelligence 20(2-4), 133–157 (Feb-Apr 2006)
3. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the *S*CIFF framework. ACM T. Comput. Log. 9(4) (2008)
4. Alberti, M., Gavanelli, M., Lamma, E.: Deon+ : Abduction and constraints for normative reasoning. In: Artikis, A., Craven, R., Cicekli, N.K., Sadighi, B., Stathis, K. (eds.) Logic Programs, Norms and Action - Essays in Honor of Marek J. Sergot on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 7360, pp. 308–328. Springer (2012)
5. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Sartor, G., Torroni, P.: Mapping deontic operators to abductive expectations. Computational and Mathematical Organization Theory 12(2–3), 205 – 225 (Oct 2006)
6. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and verification of agent interactions using social integrity constraints. Electronic Notes in Theoretical Computer Science 85(2) (2003)

7. Alberti, M., Gomes, A.S., Gonçalves, R., Leite, J., Slota, M.: Normative systems represented as hybrid knowledge bases. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17-18, 2011. Proceedings. Lecture Notes in Artificial Intelligence, vol. 6814, pp. 330–346. Springer Verlag (2011)
8. ALFEBIITE: A Logical Framework for Ethical Behaviour between Infohabitants in the Information Trading Economy of the universal information ecosystem. IST-1999-10298 (1999)
9. Arisha, K.A., Ozcan, F., Ross, R., Subrahmanian, V.S., Eiter, T., Kraus, S.: IMPACT: a Platform for Collaborating Agents. IEEE Intelligent Systems 14(2) (1999)
10. Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computational and Mathematical Organization Theory 12, 71–79 (2006)
11. Bracciali, A., Demetriou, N., Endriss, U., Kakas, A.C., Lu, W., Mancarella, P., Sadri, F., Stathis, K., Terreni, G., Toni, F.: The KGP model of agency for global computing: Computational model and prototype implementation. In: Priami, C., Quaglia, P. (eds.) Global Computing. LNCS, vol. 3267. Springer (2004)
12. De Vos, M., Padget, J., Satoh, K.: Legal modelling and reasoning using institutions. In: Onada et al. [28], pp. 129–140
13. Dignum, V., Meyer, J.J., Dignum, F., Weigand, H.: Formal specification of interaction in agent societies. In: Proceedings of the Second Goddard Workshop on Formal Approaches to Agent-Based Systems (FAABS), Maryland (Oct 2002)
14. Dignum, V., Meyer, J.J., Weigand, H.: Towards an organizational model for agent societies using contracts. In: Castelfranchi, C., Lewis Johnson, W. (eds.) Proceedings of the First International Joint Conference on Autonomous Ag ents and Multiagent Systems (AAMAS-2002), Part II. pp. 694–695. ACM Press, Bologna, Italy (Jul 15–19 2002)
15. Dignum, V., Meyer, J.J., Weigand, H., Dignum, F.: An organizational-oriented model for agent societies. In: Proc. International Workshop on Regulated Agent-Based Social Systems: Theories and Applications. AAMAS'02, Bologna (2002)
16. Du, J., Wang, K., Shen, Y.D.: A tractable approach to ABox abduction over description logic ontologies. In: Brodley, C.E., Stone, P. (eds.) Proc. AAAI 2014. AAAI Press (2014), http://www.aaai.org/Library/AAAI/aaai14contents.php
17. Eiter, T., Subrahmanian, V., Pick, G.: Heterogeneous active agents, I: Semantics. Artificial Intelligence 108(1-2), 179–255 (March 1999)
18. Fung, T.H., Kowalski, R.A.: The IFF proof procedure for abductive logic programming. J. Logic Program. 33(2), 151–165 (1997)
19. Gavanelli, M., Lamma, E., Riguzzi, F., Bellodi, E., Zese, R., Cota, G.: An abductive framework for Datalog+- ontologies. In: De Vos, M., Eiter, T., Lierler, Y., Toni, F. (eds.) Technical Communications of the 31st Int'l. Conference on Logic Programming (ICLP 2015). No. 1433 in CEUR Workshop Proceedings, Sun SITE Central Europe, Aachen, Germany (2015), http://ceur-ws.org/Vol-1433/tc_89.pdf
20. Gavanelli, M., Lamma, E., Riguzzi, F., Bellodi, E., Zese, R., Cota, G.: Abductive logic programming for Datalog+- ontologies. In: Ancona, D., Maratea, M., Mascardi, V. (eds.) Proceedings of the 30th Italian Conference on Computational Logic (CILC2015), Genova, Italy, 1-3 July 2015. pp. 128–143. No. 1459 in CEUR Workshop Proceedings, Sun SITE Central Europe, Aachen, Germany (2015), http://ceur-ws.org/Vol-1459/paper21.pdf
21. Gavanelli, M., Lamma, E., Riguzzi, F., Bellodi, E., Zese, R., Cota, G.: Abductive logic programming for normative reasoning and ontologies. In: Otake, M.,

Kurahashi, S., Ohta, Y., Satoh, K., Bekki, D. (eds.) New Frontiers in Artificial Intelligence (JSAI-isAI 2015 Workshops, LENLS, JURISIN, AAA, HAT-MASH, TSDAA, ASD-HR and SKL, Kanagawa, Japan, November 16-18, 2015, Revised Selected Papers). Lecture Notes in Artificial Intelligence, vol. 10091. Springer (2017)
22. Governatori, G., Rotolo, A.: BIO logical agents: Norms, beliefs, intentions in defeasible logic. Autonomous Agents and Multi-Agent Systems 17(1), 36–69 (2008)
23. Jaffar, J., Maher, M.: Constraint logic programming: a survey. Journal of Logic Programming 19-20, 503–582 (1994)
24. Jaffar, J., Maher, M.J.: Constraint logic programming: A survey. J. Logic Program. 19/20, 503–581 (1994)
25. Japanese Civil Code, part I. https://en.wikisource.org/wiki/Civil_Code_of_Japan/Part_I, retrieved on July 19, 2016
26. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive Logic Programming. Journal of Logic and Computation 2(6), 719–770 (1993)
27. Klarman, S., Endriss, U., Schlobach, S.: ABox abduction in the description logic $\mathcal{ALC}$. J. Autom. Reasoning 46(1), 43–80 (2011)
28. Onada, T., Bekki, D., McCready, E. (eds.): New Frontiers in Artificial Intelligence - JSAI-isAI 2010 Workshops, LNCS, vol. 6797. Springer (2011)
29. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics 7(1), 25–75 (1997), http://dx.doi.org/10.1080/11663081.1997.10510900
30. Sadri, F., Stathis, K., Toni, F.: Normative KGP agents. Computational & Mathematical Organization Theory 12(2-3), 101–126 (2006)
31. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In: Onada et al. [28]
32. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. Commun. ACM 29, 370–386 (1986)
33. Van Belleghem, K., Denecker, M., De Schreye, D.: A strong correspondence between description logics and open logic programming. In: Naish, L. (ed.) Proc. Fourteenth International Conference on Logic Programming. MIT Press (1997)
34. Wright, G.: Deontic logic. Mind 60, 1–15 (1951)

COLIEE-2016: Evaluation of the Competition on Legal Information Extraction and Entailment

Mi-Young Kim[1], Randy Goebel[1], Yoshinobu Kano[2], and Ken Satoh[3]

University of Alberta, Canada[1]
Faculty of Informatics, Shizuoka University, Japan[2]
National Institute of Informatics, Japan[3]
{miyoung2, rgoebel}@ualberta.ca, kano@inf.shizuoka.ac.jp, ksatoh@nii.ac.jp

Abstract. We present the evaluation of legal question answering of the Competition on Legal Information Extraction/Entailment (COLIEE)-2016. The COLIEE-2016 task consists of three sub-tasks: legal information retrieval (sub-task 1), recognizing entailment between articles and queries (sub-task 2), and combination of the previous two sub-tasks (sub-task 3). Participation was open to any group based on any approach, and the task attracted 11 teams. We received 7 submissions to the sub-task 1 (for a total of 14 runs), 6 submissions to the subtask 2 (for a total of 13 runs), and 5 submissions to the subtask 3 (for a total of 10 runs).

Keywords: legal question answering, recognizing textual entailment, information retrieval

1. Introduction

The Juris-Informatics workshop series was created to promote community discussion on both fundamental and practical issues on legal information processing, with the intention to embrace various backgrounds such as law, social science, information processing, logic and philosophy, and including the conventional "AI and law" area.

Information extraction and reasoning from legal data is one of the important targets of JURISIN, including legal information representation, relation extraction, textual entailment, summarization, and their applications. Participants in the JURISIN workshops have examined a wide variety of information extraction techniques and environments with a variety of purposes, including retrieval of relevant articles, entity/relation extraction from legal cases, reference extraction from legal cases, finding relevant precedents, summarization of legal cases, and legal question answering.

During the last two years, we held the first and second competitions on legal information extraction/entailment (COLIEE 2014, COLIEE 2015) on a legal data collection, and this helped establish a major experimental effort in the legal information extraction/retrieval field. We held the third competition (COLIEE-2016) this year, with the motivation of continuing to help create a research community of practice for the capture and use of legal information. The COLIEE competition is

about the legal question answering task, and the structure of this competition has been decomposed into three subtasks of retrieval, information extraction, and entailment.

2. The Legal Question Answering Task

This competition focuses on two aspects of legal information processing related to answering yes/no questions from Japanese legal bar exams (the relevant data sets have been translated from Japanese to English, and were available in both languages).

1) Phase 1 of the legal question answering task involves reading a legal bar exam question Q, and extracting a subset of Japanese Civil Code Articles S_1, S_2,..., S_n from the entire Civil Code, which are appropriate for answering the question such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and the entire Civil Code Articles, we have to retrieve the set of "S_1, S_2, ..., S_n" as the outcome of phase 1.

2) Phase 2 of the legal question answering task involves the identification of an entailment relationship such that

Entails(S_1, S_2, ..., S_n , Q) or Entails(S_1, S_2, ..., S_n , not Q).

Given a question Q and relevant articles S_1, S_2, ..., S_n, we have to determine if the relevant articles entail "Q" or "not Q". The answer of this track is binary: "YES"("Q") or "NO"("not Q"). This phase typically require some information extraction (e.g., named entity identification, relation extraction, etc.), followed by any variety of methods for textual inference, to confirm entailments. Details are given in the next section.

2.1 Phase 1 Details

Our goal is to explore and evaluate legal document retrieval technologies that are both effective and reliable. The task investigates the performance of systems that search a static set of civil law articles using previously unseen queries, and return relevant articles. We say an article is "Relevant" to a query if and only if the query sentence can be entailed from the meaning of the article. If combining the meanings of more than one article (e.g., "A," "B," and "C") can answer a query sentence, then all the articles ("A," "B,", and "C") are considered "Relevant." If a query can be answered by an article "D," and it can be also answered by another article "E" independently, we also say that both "D" and "E" are "Relevant." This task requires the retrieval of all the articles that are relevant to answering a query.

Japanese civil law articles (and English translation) have been provided, and training data consists of query and relevant article pairs. The process of executing the queries over the articles and generating the experimental runs should be entirely automatic. Test data includes only queries but no relevant articles.

2.2 Phase 2 Details

The goal of Phase 2 is to construct Yes/No question answering systems for legal queries, by entailment from the relevant articles. The task investigates the performance of systems that answer "Y" or "N" to previously unseen queries by somehow comparing the intersection of meanings between queries and relevant articles. Training data consists of triples: a query, relevant articles and a correct answer "Y" or "N." The process of executing the queries over the relevant articles and generating the experimental runs should be entirely automatic. Test data includes only queries and relevant articles, but no "Y/N" label.

2.3 Phase 3 Details

Phase 3 combines Phase 1 and Phase 2. If a "Yes/No" legal bar exam question is given, a legal information retrieval system retrieves relevant Civil Law articles. Then, the task investigates the performance of systems that answer "Y" or "No" to previously unseen queries by comparing the meanings of the queries and retrieved Civil Law articles. Test data includes only queries, but no "Y/N" label, and no relevant articles.

3. The Legal Question Answering Data Corpus

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

1) Phase 1 problem is to use an identified set of legal yes/no questions to retrieve relevant Civil Law articles. In this case, the correct answers have been determined by a collection of law students, and those answers are used to calibrate the performance of a program to solve Phase 1.

2) Phase 2 task requires some method of information extraction from both the question and the relevant articles, and then to confirm a simple entail relationship as described in the Section 2: either the articles confirms "yes" or "no" as an answer to the yes/no questions.

3) Phase 3 task is combination of Phase 1 and Phase 2. It requires both of the legal information retrieval system and textual entailment system. If legal yes/no questions are given, each participant's legal information retrieval system retrieves relevant Civil Law articles. After that, each participant confirms a "Yes/No" entailment relationship between input yes/no question and the retrieved articles.

Participants can choose which Phase they will attempt, amongst the three sub-tasks as follows:

Table 1. Example of a query and a relevant article

Question	*A person who made a manifestation of intention which was induced by duress emanated from a third party may rescind such manifestation of intention on the basis of duress, only if the other party knew or was negligent of such fact.*
Related Article	*(Fraud or Duress) Article 96 (1)Manifestation of intention which is induced by any fraud or duress may be rescinded.(2)In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.(3)The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.*

Table 2. Examples of sentence pairs holding different entailment relations

Question	A special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty.
Related Article	(Special Agreement Disclaiming Warranty)Article 572 Even if the seller makes a special agreement to the effect that the seller will not provide the warranties set forth from Article 560 through to the preceding Article, the seller may not be released from that responsibility with respect to any fact that the seller knew but did not disclose, and with respect to any right that the seller himself/herself created for or assigned to a third party.
Label	Yes
Question	A manager must engage in management exercising care identical to that he/she exercises for his/her own property.
Related Article	(Urgent Management of Business)Article 698 If a Manager engages in the Management of Business in order to allow a principal to escape imminent danger to the principal's person, reputation or property, the Manager shall not be liable to compensate for damages resulting from the same unless he/she has acted in bad faith or with gross negligence.
Label	No

1. Sub-task 1: legal information retrieval Task. Input is a bar exam “Yes/No” question and output should be relevant civil law articles. (Phase 1)

2. Sub-task 2: Recognizing Entailment between law articles and queries. Input is a pair of a question and relevant article(s), and output should be “Yes” or “No. (Phase 2)

3. Sub-task 3: Combination of sub-task 1 and sub-task 2. Input is a bar exam “Yes/No”question and output should be “Yes” or “No.” (Phase 3)

Table 1 shows an example of query and relevant articles.

In the entailment subtask, given two sentences A and B, a system must determine whether the meaning of B was entailed by A. In particular, a system is required to assign to each pair either the “Yes” label (when A entails B, viz., B cannot be false when A is true), or the “No” label (when A does not entail B). Table 2 shows examples of sentence pairs holding different entailment relations.

The subtask 3 is combination of subtask 1 and subtask 2.

The structure of the test corpora is derived from a general XML representation developed for use in RITEVAL, one of the tasks of the NII Testbeds and Community for Information access Research (NTCIR) project[1].

The RITEVAL format was developed for the general sharing of information retrieval on a variety of domains. The format of the JURISIN competition corpora is derived from an NTCIR representation for confirmed relationships between questions and the articles and cases relevant to answering the questions, as in the following example:

<pair label="Y” id="H18-1-2">
<t1>
(Seller's Warranty in cases of Superficies or Other Rights)Article 566 (1)In cases where the subject matter of the sale is encumbered with for the purpose of a superficies, an emphyteusis, an easement, a right of retention or a pledge, if the buyer does not know the same and cannot achieve the purpose of the contract on account thereof, the buyer may cancel the contract. In such cases, if the contract cannot be cancelled, the buyer may only demand compensation for damages. (2)The provisions of the preceding paragraph shall apply mutatis mutandis in cases where an easement that was referred to as being in existence for the benefit of immovable property that is the subject matter of a sale, does not exist, and in cases where a leasehold is registered with respect to the immovable property.(3)In the cases set forth in the preceding two paragraphs, the cancellation of the contract or claim for damages must be made within one year from the time when the buyer comes to know the facts.
(Seller's Warranty in cases of Mortgage or Other Rights)Article 567(1)If the buyer loses his/her ownership of immovable property that is the object of a sale because of the exercise of an existing statutory lien or mortgage, the buyer may cancel the contract.(2)If the buyer preserves his/her ownership by incurring expenditure for costs, he/she may claim reimbursement of those costs from the seller.(3)In the cases set forth in the preceding two paragraphs, the buyer may claim compensation if he/she suffered loss.
</t1>
<t2>
There is a limitation period on pursuance of warranty if there is restriction due to superficies on the subject matter, but there is no restriction on pursuance of warranty if the seller's rights were revoked due to execution of the mortgage.
</t2>

[1] As described at http://sites.google.com/site/ntcir11riteval/

</pair>

The above is an example where query id "H18-1-2" is confirmed to be answerable from article numbers 566 and 567 (relevant to Phase 1). The pair label "Y" in this example means the answer of query is "Yes," which is entailed from the relevant articles (relevant to Phase 2 and 3).

The COLIEE training data was built from the bar exam (short answer test) civil code part published from 2006-2013, and consists of 8 XML files. Each file corresponds to one year's publication. The total number of queries in the training data is 412, and each query has average 1.27 relevant articles. The test data size is 95 queries for Phase 1 (extracted from the bar exam of 2014), and 70 queries for Phase 2 (extracted from the bar exam of 2015). For Phase 3, we use the same test data from Phase 1.

4. Evaluation Metrics and Baselines

The measures for ranking competition participants are intended only to calibrate the set of competition submissions, rather than provide any deep performance measure. The data sets for Phases 1 and 2 are annotated, so simple information retrieval measures (precision, recall, F-measure [7], accuracy) can be used to rank each submission. As noted above, the intention is to start to build a community of practice regarding legal textual entailment, so that the adoption and adaptation of general methods from a variety of fields is considered, and that participants share their approaches, problems, and results.

For Phase 1, evaluation measure will be precision, recall and F-measure:

Precision =(the number of correctly retrieved articles for all queries)/(the number of retrieved articles for all queries),
Recall = (the number of correctly retrieved articles for all queries)/(the number of relevant articles for all queries),
F-measure = (2 x Precision x Recall)/(Precision + Recall).

For Phase 2, the evaluation measure will be accuracy, with respect to whether the yes/no question was correctly confirmed:

Accuracy = (the number of queries which were correctly confirmed as true or false)/(the number of all queries).

For Phase 3, the evaluation measure is the same with that of Phase 2.

For Phase 1, we consider the traditional TF-IDF [6] as the baseline model. For Phase 2, there is a balanced positive-negative sample distribution in the dataset (52.86% yes, and 47.14% no) for the formal run, so we consider the baseline for true/false evaluation is the accuracy when returning always "yes," which is 52.86%. Tables 3 and 4 present the performance of the baseline models.

Table 3. Baseline of Phase one

Phase One Baseline	F-measure
TF-IDF with lemmarization (choosing one most relevant article)	0.5310

Table 4. Baseline of Phase two

Phase Two Baseline	Accuracy
When 'Yes' labels are all chosen	0.5286

5. Submitted Runs and Results

Overall, 11 teams registered and 8 teams submitted their system results. Some participants submitted multiple runs for a task. We received 7 submissions to the subtask 1 (for a total of 14 runs), 6 submissions to the subtask 2 (for a total of 13 runs), and 5 submissions to the subtask 3 (for a total of 10 runs).

Tables 5 and 6 summarize the submitted systems' techniques. We present the results achieved by runs against the Information Retrieval and Entailment subtasks in Tables 7, 8, and 9, respectively. Some systems performed well above the best baselines from Tables 3 and 4.

John et al. [1] use the Support Vector Machine (SVM) and Random Forests (RF) as classifiers, and construct word embedding by transforming each word to vector representation. They generate a sentence representation by summing over the embeddings of all words in the sentence. These embeddings are the inputs to the classifiers, and they find one most similar article for each query. They submitted three runs for the subtask 1. The first and second runs are the results using SVM and RF respectively. For the third run, they used K-means clustering, and then calculated cosine similarity between a query and articles only in the same cluster.

For the subtask 2 that is textual entailment, they used various features such as string matching, word ordering, word overlap, part-of-speech overlap, dependency parsing, word embedding similarity, WordNet similarity, and vector space-based similarity. These features were used as input to the two classifiers: RF and Child-sum Tree LSTM. They submitted five runs for the subtask 2: two RFs, and three Child-sum tree LSTMs by differentiating parameter setting. Even though they proposed methods for both of the task 1 and task 2, they did not participate in task 3, which is the combination of task 1 and task 2.

M.-Y. Kim et al. [2] use the TF-IDF model to get similar documents for the subtask 1, and then re-rank the documents using the ranking SVM model. They submitted three runs for the subtask 1: TF-IDF, Ranking SVM using lemma and dependency bigram as features, and Ranking SVM adding TF-IDF score as features into the 2nd run. For

Table 5. Approaches of submitted systems for IR (Phase 1)

ID	Run	Approaches
JNLN1[7]	JULN1	Ranking SVM using dependency node semantic similarity, N-gram intersection, paragraph links, paragraph link types and negation similarity.
HUKB[8]	HUKB-1	BM25
	HUKB-2	BM25 with Expansion for the article to be applied mutatis mutandis)
	HUKB-3	BM25 with Query Keyword Expansion
	HUKB-4	BM25 with Query Keyword Expansion and Expansion for the article to be applied mutatis mutandis
JNLN2[6]	JNLN2	Ranking SVM using cosine similarity and various distance features as features
iLis7[5]	iLis7	LSM ensemble
JNLN3 [3]	JNLN3	SVM model which incorporates various distance features
UofA[2]	UofA-1	TF-IDF
	UofA-2	Ranking SVM lemma and dependency bigram as features
	UofA-3	Adding IR score as features into UofA-2
N01[1]	N01-1	SVM with word embedding input
	N01-2	Random Forests with Word Embedding input
	N01-3	K-means clustering and cosine similarity in the same cluster

the subtask 2, they first divide queries/articles into condition, conclusion, and exceptional parts, and then extract lexical semantic feature, negation feature, and antonym feature. For the term expansion, they used paraphrasing. And then they learned SVM classifier to get yes/no answers.

Do et al. [3] also built a ranking SVM model that incorporates various features such as TF-IDF, Euclidean distance, Manhattan distance, Jaccard distance, Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation for the subtask 1. For the subtask 2, they used a convolutional neural network using word embedding, LSI, and TF-IDF score as input features.

Taniguchi and Kano [4] participated only in the subtasks 2 and 3. They extracted the end-of-sentence expressions and case-role analysis. Using these two features, they constructed two kinds of heuristic rules: one is loose and the other is more strict. They also participated in the subtask 3 using the method of subtask 2. The subtask 3 is

Table 6. Approaches of submitted systems for the Entailment (Phase 2)

ID	Run	Approaches
JNLN1 [7]	JNLN1	A simple rule just by checking the maximum n-gram intersection.
KIS [4]	KIS-1	Two heuristic rules using subjective cases and an end-of-sentence expression for each sentence with the "provided that" sentence
	KIS-2	Two heuristic rules using subjective cases and an end-of-sentence expression for each sentence without the "provided that" sentence
	KIS-3	A loose rule using subjective cases and an end-of-sentence expression for each sentence with the "provided that" sentence
	KIS-4	A strict rule using subjective cases and an end-of-sentence expression for each sentence without the "provided that" sentence
iLis7 [5]	iLis7	Majority vote with decision tree, linear SVM, and CNN
JNLN3 [3]	JNLN3	Convolutional neural network using word embedding, LSI, and Tf-IDF as features
UofA [2]	UofA	Condition/Conclusion/Exception detection + Paraphrasing + SVM classification
N01 [1]	N01-1	Random Forests using various features
	N01-2	Random Forests using various features (different parameter setting)
	N01-3	Child-sum tree LSTM using various features
	N01-4	Child-sum tree LSTM using various features (different parameter setting)
	N01-5	Child-sum tree LSTM using various features (different parameter setting)

combination of information retrieval and textual entailment. However, without the information retrieval procedure, they just applied their textual entailment method between a query and each legal article, and counted 'yes' ('no') cases for the subtask3.

K. Kim et al. [5] used a variety of features, including lexical similarity, syntactic similarity, and semantic similarity, and also proposed ensemble similarity using a least square method (LSM ensemble) and linear discriminant analysis (LDA ensemble) for the subtask 1. Among their many approaches, LSM ensemble showed best performance for the training data.

Table 7. IR results(Phase 1) on the formal run data

Run	Prec.	Recall	F-m.	Run	Prec.	Recall	F-m.
JNLN1 [7]	0.6105	0.4427	0.5133	JNLN3 [3]	0.6526	0.4733	0.5487
HUKB-1[8]	0.6154	0.4886	0.5447	UofA-1 [2]	0.2000	0.1450	0.1681
HUKB-2[8]	0.6250	0.4962	0.5532	UofA-2 [2]	0.2737	0.1985	0.2301
HUKB-3[8]	0.6316	0.4580	0.5310	UofA-3 [2]	0.5895	0.4275	0.4956
HUKB-4[8]	0.6316	0.4580	0.5310	N01-1 [1]	0.3053	0.2214	0.2566
JNLN2 [6]	0.6211	0.4504	0.5221	N01-2 [1]	0.4211	0.3053	0.3540
iLis7 [5]	**0.7272**	**0.5496**	**0.6261**	N01-3 [1]	0.4000	0.2901	0.3363

Table 8. Entailment results (Phase 2) on the formal run data

Run	Accu.	Run	Accu.
JNLN1[7]	0.5286	UofA [2]	0.5571
KIS-1[4]	**0.6286**	N01-1 [1]	0.5429
KIS-2 [4]	**0.6286**	N01-2 [1]	0.5429
KIS-3 [4]	0. 5857	N01-3 [1]	0.4857
KIS-4 [4]	0.5857	N01-4 [1]	0.4857
iLis7 [5]	**0.6286**	N01-5 [1]	0.5714
JNLN3 [3]	0.4857		

Table 9. IR+Entailment results (Phase 3) on the formal run data

Run	Accu.	Run	Accu.
JNLN1 [7]	0.4000	iLis7 [5]	0.5368
KIS-1[4]	0.5158	JNLN3 [3]	0.4737
KIS-2 [4]	0.5158	UofA [2]	0.4632
KIS-3 [4]	0.5263	UofA [2]	0.5474
KIS-4 [4]	0.5263	**UofA [2]**	**0.5579**

For the subtask 2, they used word overlap, cosine similarity, substring similarity, the *Lucene* similarity score, 3 different role similarity scores, 6 *WordNet* similarity scores, negative term ratio, and length ratio. Majority vote with three classifiers (decision tree, linear SVM, and convolutional neural network (CNN) classifiers) showed best performance among the many ensemble methods.

Nguyen et al. [6] used word2vec and WordNet for query expansion on the subtask 1. They also used Ranking SVM using Cosine similarity and various distance features. For the phase 2, they proposed a tree-based convolutional neural network model (TBCNN). The inputs for TBCNN are vector representation from Word2Vec and tree representation of questions and articles. However, they did not submit their results on the phase 2.
Carvalho et al. [7] used dependency node semantic similarity, N-gram intersection, paragraph links, paragraph link types and negation similarity for the subtask 1. They provided their own relevant ranking score, and then re-ranked the relevance through ranking SVM. For the phase 2, they used a simple rule just by checking the maximum n-gram intersection.

Onodera and Yoshioka [8] participated only for the subtask 1. They calculated document similarity by BM25, and then used a Boolean IR model for legal terminology in query. As a result, they obtained top K-ranked articles. They submitted four runs: Baseline BM25, BM25 with expansion for articles to be applied mutatis mutandis, BM25 with query keyword expansion, BM25 with query keyword expansion and expansion for articles to be applied, mutatis mutandis.

We found that most papers used the morphological analysis and dependency parsing analysis results included in the released data.

In Table 7, we can see that there are four systems which show better performance than the baseline model: HUKB1, HUKB2, iLis7, and JNLN3. ILis7 shows the best performance using LSM ensemble, and it achieved significant increase of the F-measure compared to the baseline model. HUKB1 and HUKB2 used simple BM25 and additional expansion for the articles to be applied mutatis mutandis, respectively. JNLN3 used SVM model that incorporates various distance features.

JNLN1, HUKB3, HUKB4, JNLN2, JNLN3, UofA, and N01 systems produced only one most relevant article for each query (in total 95 articles). HUKB1 and HUKB2 produced 104 articles respectively, and iLis7 produced 99 articles. iLis7 produced 1.04 article per query, and they showed best precision and recall. HUKB2 and HUKB1 showed the second and third highest recall, and they produced average 1.09 articles per query. For the subtask 1, a lot of participants tried a ranking SVM, which showed best performance in the COLIEE-2015, but the LSM ensemble showed best performance in this year's competition.

Table 8 reports the results of Phase 2, where KIS-1, KIS-2, and iLis7 showed best accuracy, which is significantly better than the baseline performance. KIS-1 and KIS-2 used simple rules using subjective cases and an end-of-sentence expression, and iLis7 used majority vote with decision tree, linear SVM, and CNN using various features. Among the 13 systems, 9 systems showed better performances than the baseline model on the subtask 2. It is interesting that simple heuristic rule of KIS showed best performance than other complicated machine learning methods.

As shown in Table 9, the UofA-3 system showed best performance when two phases are combined, even though their phase 1 and phase 2 systems were not best. UofA is the only system that performed paraphrasing, and detected condition-conclusion-exceptions for the query/article; it was the only system that extracted the article segment for which the query is semantically related. In contrast to other systems that recognized textual entailment from the whole article to the query, UofA-3 compared the approxmate semantics from a specific article segment to the approximated semantics of the query.

In contrast to the last year's competition, many systems were more dependent on the syntactic/semantic linguistic features, and used negation information as key features. We hope to expand the data size in the next competition, to get more robust performance for each phase.

6. Conclusion

We have summarized the results of the COLIEE-2016 competition. Three subtasks were evaluated: (1) Phase 1: finding relevant articles (information retrieval) (2) Phase 2: answering yes/no questions by comparing the meanings between a query and relevant articles (textual entailment), (3) combination of Phase 1 and Phase 2. There were 11 teams who participated in this competition, and we received results from 8 teams. There were 7 submissions to Phase 1 (for a total of 14 runs), 6 submissions to Phase 2 (for a total of 13 runs), and 5 submissions to Phase 3 (for a total of 10 runs).

A variety of methods were used for Phase 1: ranking SVM, TF-IDF, BM25, query expansion, least squares method, linear discriminant analysis, random forests, and K-means clustering. Various features were also proposed: paragraph links, N-gram intersection, dependency node semantic similarity, and a lot of distance features. For Phase 2, while some participants used simple heuristic rules from the end-of sentence expression, case-role analysis, and maximum N-gram intersection, other participants used ensemble classifiers by combining various classifiers, or applied convolutional neural network, Child-sum tree LSTM, and random forests. For term expansion, one participant tried paraphrasing based on double translation. In contrast to previous years, many systems were dependent on the syntactic/semantic linguistic features, and used negation information as key features. Some systems significantly outperformed baseline systems, and we had an interesting result that the best system on Phase 3 did not show best results on each Phase 1 and Phase 2.

For future competitions, we need to expand the data sets in order to improve the robustness of results. We also need to more deeply investigate how the performance of each Phase 1 and Phase 2 affect the overall performance of Phase 3.

Acknowledgements

This research was supported by the Alberta Innovates Centre for Machine Learning (AICML), the Natural Sciences and Engineering Research Council (NSERC), and

This competition was partially supported by JSPS KAKENHI Grant Number JP26280091.

Reference

1. Adebayo Kolawole John , Luigi Di Caro, Guido Boella, and Cesare Bartolini, "TEAM-NORMAS' PARTICIPATION AT THE COLIEE 2016 BAR LEGAL EXAM COMPETITION", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **N01**)
2. Mi-Young Kim, Ying Xu, Yao Lu and Randy Goebel, "Legal Question Answering Using Paraphrasing and Entailment Analysis", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **UofA**)
3. Phong-Khac Do, Huy-Tien Nguyen, Chien-Xuan Tran, Minh-Tien Nguyen and Nguyen Le Minh, "Legal Question Answering using Ranking SVM and Deep Convolutional Neural Network", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **JNLN3**)
4. Ryosuke Taniguchi and Yoshinobu Kano, "Legal Yes/No Question Answering System using Case-Role Analysis", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **KIS**)
5. Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kihyun Hong and Young-Yik Rhim, "An Ensemble Based Legal Information Retrieval and Entailment System", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **iLis7**)
6. Truong-Son Nguyen, Viet-Anh Phan, Thanh-Huy Nguyen, Hai-Long Trieu, Ngoc-Phuong Chau, Trung-Tin Pham, and Le-Minh Nguyen, "Legal Information Extraction/Entailment Using SVM-Ranking and Tree-based Convolutional Neural Network", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **JNLN2**)
7. Danilo S. Carvalho, Vu Duc Tran, Khanh Van Tran, Viet Dac Lai, and Minh-Le Nguyen, Lexical to Discourse-level Corpus Modeling for Legal Question Answering", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **JNLN1**)
8. Daiki Onodera and Masaharu Yoshioka, "Civil Code Article Information Retrieval System based on Legal Terminology and Civil Code Article Structure", Tenth International Workshop on Juris-informatics (JURISIN), 2016 (Submission ID: **HUKB**)

AN APPROACH TO INFORMATION RETRIEVAL AND QUESTION ANSWERING IN THE LEGAL DOMAIN

Adebayo Kolawole John, Luigi Di Caro, Guido Boella, and Cesare Bartolini

Dipartimento di Informatica, Universita Di Torino
Corso Svizzera 185, Torino, 10149, Italy
{kolawolejohn.adebayo}@unibo.it, {dicaro,guido.boella}@di.unito.it,
{cesare.bartolini}@uni.lu,

Abstract. We describe in this paper, a report of our participation at COLIEE 2016 Information Retrieval (IR) and Legal Question Answering (LQA) tasks. Our solution for the IR part employs the use of a simple but effective Machine Learning (ML) procedure. Our Question Answering solution answers "YES or 'NO' to a question, i.e., 'YES' if the question is entailed by a text and 'NO' otherwise. With recent exploit of Multi-layered Neural Network systems at language modeling tasks, we presented a Deep Learning approach which uses an adaptive variant of the Long-Short Term Memory (LSTM), i.e. the Child Sum Tree LSTM (CST-LSTM) algorithm that we modified to suit our purpose. Additionally, we benchmarked this approach by handcrafting features for two popular ML algorithms, i.e., the Support Vector Machine (SVM) and the Random Forest (RF) algorithms. Even though we used some features that have performed well from similar works, we also introduced some semantic features for performance improvement. We used the results from these two algorithms as the baseline for our CST-LSTM algorithm. All evaluation was done on the COLIEE 2015 training and test sets. The overall result confirms the competitiveness of our approach.

Keywords: Child-Sum tree lstm, Information Retrieval, textual entailment

1 Introduction

COLIEE is an annual competition[1] for legal Information Extraction (IE) and Retrieval. COLIEE 2016 focuses on two main tasks, i.e. Textual Entailment (TE) and Information Retrieval (IR) all proposed as a form of Legal question answering task. The organizers divided the task into three sub-tasks, i.e., subtask one involves reading a legal bar exam question Q, presented as a query, and extracting a subset of the Japanese Civil Code Articles C_1,C_2,C_3...,C_n from the entire Civil Codes made available in the COLIEE Dataset. The set of civil codes retrieved by the system must sufficiently answer the initial query Q.

[1] collocated with the JURISIN workshop

Sub-Task two advances this objective by identifying entailment relationship between an article or a set of articles and a given query, e.g., $E_f(C_1, C_2, C_3..., C_n, Q)$, where E_f is the entailment function that produces a True or False answer. The third combines the efforts of the first and second tasks. For the tasks, the organizers released a corpus of Japanese Legal Civil Code Articles, the articles have been translated into English for obvious reasons.

This paper presents a report of our participation in the first and second tasks, as well as a description of the systems we developed for these tasks. The remaining parts of the paper is structured as follows. In section 2, we succinctly describe the general IR procedure and our own approach. Next, we describe the TE problem and the state of the arts. This is followed by elucidation of the features used for the SVM and RF classification and subsequently the Deep Learning approach presented. Finally, we discuss the Experiments and the results obtained.

2 Task 1- Information Retrieval

Information Retrieval (IR) involves retrieving a set of item(s), usually from among multiple options such that the result satisfies a user's information need (Query). Earliest approach to IR involves the use of Lexical Matching or Keyword Search. This is however grossly inefficient as it is not robust to Natural Language ambiguities (i.e., synonymy and polysemy). The Bag of Word (BOW) based models such as LSA [8] and Term Frequency Inverse Document Frequency (TFIDF) partially overcome this ambiguities by grouping terms that occur in a similar context into the same semantic cluster based on their vector representation in the semantic space. Topic Modeling approaches have also been used [18, 1]. The idea here is to probabilistically assign topics to documents as well as query based on their term composition. Similar documents tends to belong to similar topics, thus, the task would be to retrieve documents with similar topics to the query. LDA [4], a generative probabilistic algorithm excel in this category. Since the Query is usually a fraction of the document, a common approach for improving retrieval accuracy is by Query Expansion which uses the Part-of-Speech (POS) to retrieve synonyms for Query enrichment. While this partially helps, the approaches still do not necessarily capture the full semantics of a language since they discard useful information about word order and context without word-sense disambiguation. Recently, Machine Learning techniques are being employed to overcome this barrier, e.g., the authors in [9, 31] proposes Deep Learning (DL) architectures. There are now evolving some IR systems which can arguably be classified as Question Answering (QA) systems. These systems exploits the massive set of structured information[2] available on some Knowledge Base (KB) like the DBPedia, Freebase, Wikipedia and BabelNet etc., to retrieve information about any Query. QAKIS [6] and DEQA [19]

[2] These information are serialized as structured SPO triples e.g., Subject-Predicate-Object facts.

are examples of these systems and should not be confused with the demand of the underlying task.

While encouraging results have been reported with DL techniques, these systems generalizes better when trained on a large dataset. However, COLIEE training data is different in that the set of documents to be queried are not documents in actual sense but Japanese Civil Codes, each containing less than 4 sentences. Also, the total number of articles available is very small, actually less than 500 instances. Our idea is that a simple challenge should not be approached with a rather too complex system. More so, simple solutions in NLP (e.g. N-gram models) often compete favourably with some complex techniques. We submitted three runs using simple machine learning approaches. We describe our solution below.

2.1 Distributed Representation and Information Retrieval

It can be argued that IR is about finding semantic connection between a query and some document(s) in a corpus. In our case, the corpus to be queried is the set of Japanese Civil Codes. We limit ourselves to finding the most related of the civil codes to the query. As earlier explained, modeling language as BOW trades away several useful contextual information which Humans find useful in aggregating meaning. An obvious choice is to take advantage of the distributed representation of words. Distributional Semantics rely on the idea that words within proximity usually have similar meaning [33]. With application of Neural Networks, it is now possible for a language models to learn high dimensional representation of words and their relationships such that semantically related words are transformed into similar vector representations. Specifically, such representation has greatly shown to capture semantic relatedness and similarity, thus overcoming sparsity problems in atomic representation. The Word2Vec[3] [23] is a popular algorithm for inducing such vector space representations of words, commonly known as word embeddings. The method exploits the Distributional Hypothesis [33] and works best when trained on large corpora. Furthermore, algebraic computations could be performed on the embeddings, which still retain the latent semantic characteristics of the vectors, e.g., *Man* + *King* - *Woman* = *Queen*. Also, relying on compositionality[4] of words [11], atomic vectors could be combined to generate phrasal or sentential representation. This ensures that the meaning of words are not captured in isolation, since the true meaning of a sentence must take into account the interplay between the words it contains [25, 26, 11].

Our task is thus to generate a sentential representation for all the Articles as well as queries, with an hypothetical assumption that sentential vectors of semantically related articles and queries would point to the same direction. In other words, we simply create a mapping between the Article(s) representation and the query representation. Since this representations are vectors, a simple choice is to measure the distance between that of a query and each Article with

[3] Word2vec is available at https://code.google.com/p/word2vec/

[4] e.g., *man* + *marry* + *woman* is closer to the vectors of *child* than *car*.

metrics like the Cosine Similarity and then rank the scores. However, this may not generalize as expected. Without loss of generality, our embedding based approach is able to capture the relatedness between embeddings of query to some document even with little feature generation.

First, we describe how to generate the word embedding. we used a combination of the training and test data as well as 19,348 documents from Eur-Lex dataset[5]. There is no specificity in the choice other than we needed a fair collection of legal text to train the Word2Vec algorithm on. Moreover, this is still by order of magnitude a very small amount of data for proper distributed representation. We used the Gensim[6] implementation of the algorithm.[7] Gensim implements a variant of the algorithm known as *Skip-Gram,* which trains word representations using a model that when given a word will predict what words are likely to occur within a fixed window around it.[8]

2.2 Model Description

Assume that each Article (hence Document) contains words x_i,x_{i+1},x_{i+2},x_{i+3}...x_n. We associate each word w in our vocabulary with a vector representation $\mathrm{x}_w \in \mathbb{R}^d$. Each x_w is of dimension $d \times V$ of the word embedding matrix W_e, where V is the size of the vocabulary. For each Document D, we generate a representation by performing an element wise sum of each $\mathrm{x}_w \in$ D. We normalize the resulting vector sum by the length of the sequence such that

$$s_i = \frac{1}{|n|} \sum_{i=1}^{|n|} x_i, \quad s_i \in \mathbb{R}^{d \times V} \tag{1}$$

where s_i denotes the embedding representation of a sentence. In the COLIEE dataset for subtask 1, each Document is associated to Label. Actually, since the original XML document contains pairs of tag *t1* (article) and *t2* (hypothesis), it is possible to have more than one articles within a *t1* tag. In this case, we exclusively separate each article as an independent document along with their subheading and labels as below. The number and phrase in bold are the label and the title respectively, followed by the main text:
265 Content of Superficies A superficiary shall have the right to use the land of others in order to own structures, or trees or bamboo, on that land.

[5] http://www.ke.tu-darmstadt.de/resources/eurlex
[6] Gensim is a python library for an array of NLP tasks. It is available at https://radimrehurek.com/gensim/
[7] Parameters used: Context Window: 5, Neural Network layer size: 50, Minimum word count: 5.
[8] Skip-gram training generally performs better than an alternative Word2Vec model known as Continuous-Bag-of-Words (CBOW) that uses an alternative objective that tries to predict a word by conditioning on all of the words that surround it within a window.

As the titles carry important information, we combined them with the main text to form each complete article.

Based on our observation of the training sample, none of the articles have more than one unique labels, We therefore approach the problem as a modest multi-class classification task, i.e., Given a training data set of the form (d_i,y_i), where $\mathrm{d}_i \in \mathbb{R}^D$ is the ith example and $\mathrm{y}_i \in \{1, \ldots, \mathrm{K}\}$ is the ith class label, we aim at finding a learning model $\mathbb{H}$ such that $\mathbb{H}(\mathrm{q}_i) = \mathrm{y}_i$, where an unseen example $\mathrm{q}_i \in \mathbb{R}^Q$ is a query instance which by our assumption correlates to a document d_i and Q is the set of all the queries. Our goal is for $\mathbb{H}$ to capture semantic relatedness between (d_i, q_i).

In order to avoid excessive feature generation, we use the sentential representation, s_i of each article. Recall that this representation is a sum over all individual word vectors x_w for each document d_i. Additional, we use 4 simple features used to capture the semantic distance of each article to another. To do this, we obtained a corpus representation D_c, where like we did for each document, D_c is a sum over all the vectors of each $\mathrm{d}_i \in \mathbb{R}^D$. Likewise, we obtained a representation Q_c for all the Queries. Using cosine similarity, we measure the distance between each article and D_c and Q_c. The first is to capture how similar each article is to the semantic space of all articles while the second captures the gap to the semantic space of all queries. Also, using cosine similarity, we compare s_i of each article to other articles, we rank and pick the top two most similar articles to it. We use the similarity scores these top similar articles as the third and fourth features. We feed into our classifier, the embedding s_i of each document along with these four features appended. To the best of our knowledge, we did not encounter any works which uses the embeddings for classification as we have done here.

For the first two runs, we used the Scikit implementation of Support vector Machine (SVM) and Random Forest (RF) algorithms as the choice classifiers. We used Grid-Search to find the optimal set of parameters.

2.3 Word Clustering

Our baseline uses a flat clustering algorithm, K-Means [12]. Only constrained by the number of clusters K which must be set a priori, it maintains that number of so-called cluster centroids. The idea is to group related words based on their distribution in the corpus. Usually, similar documents tends to have similar word distribution and by extension, cluster distribution. The LDA [4] would have been a natural choice, being generative. However, we are constrained by the rather tiny size of data we are working with, especially since LDA performs best with big training data. We used K-Means cluster size of 15. The algorithm was fed with a BOW representation of the training documents which was then used to transform the queries into their cluster representations. The idea is that instead of comparing the cluster distribution for a query with that of all documents in the training set, we simply compare with those in the same class and then calculate the pairwise distance of their vectors with cosine similarity.

3 Task 2- Textual Entailment

The sub-task2 is a Textual Entailment (TE) task. TE aims at determining whether an hypothesis h is entailed by a text t [3]. In fact, most semantic representation tasks in NLP can be reduced to a TE task, e.g., text similarity [15]. Previous works approached TE as a classification task, using hand crafted features [10] and over relying on knowledge resources [24] e.g., WordNet, etc,. At times, these features could be as simple as n-gram overlap, string matching, presence of negation words e.g., never, shouldn't etc., as well as IR techniques such as TFIDF and cosine similarity [14]. The authors in [17] recently employed a Convolutional Neural Network (CNN) to model entailment relationship on COLIEE dataset, we use their system as a baseline for our results. We attempted the same problem with two different approaches, i.e., conventional ML with feature engineering and Deep Learning (DL) technique. For our ML solution, following similar works, we handcrafted some features typical of measuring similarity between two text snippets., while in the latter, we rely on the ability of neural networks to autonomously learn latent features from data for good generalization. We explain the two systems below. TE could be approached either as a binary classification(YES/NO) or multi-class classification (YES/NO/NEUTRAL). The organizers proposed a binary classification task.

3.1 Machine Learning Approach

We obtained the following features from the t and h in order to train the RF algorithm. The features include common similarity features, some semantic features that we introduced as well as a negation feature which captures the occurrence of words with negative sentiments e.g., Can't, Should't etc. We combined the features and fit into their labels using the RF and SVM algorithms. As in the previous experiment, we use grid-search for parameter optimization.
String matching:We compare each t and h character by character and obtained the Longest common Substring, Levenshtein Distance as well as Jaccard Similarity calculated as

$$Jac = \frac{t \cap h}{t \cup h} \tag{2}$$

Word Ordering: We use the union of all tokens in a pair of sentences to build a vocabulary of non-repeating terms. Building a vector for each sentence, from their position mapping in the vocabulary. If a vocabulary term is found in the sentence, the index number of that term in the vocabulary is added to the vector. Otherwise, similarity of the vocabulary term and each term in the sentence is calculated using a WordNet based word similarity algorithm. The index number of the sentence term with highest similarity score above a threshold is added. If the first two conditions does not hold, 0 is added to the vector.
Word Overlap: We use the word n-gram overlap feature [30]. The n-grams overlap is defined as the harmonic mean of the degree of mappings between the first and second sentence and vice versa, requiring an exact string match of

n-grams in the two sentences.

$$\text{Ng(A,B)} = \left(2(\frac{\mid A \mid}{A \cap B} + \frac{\mid B \mid}{A \cap B})^{-1}\right) \tag{3}$$

Where A and B are the set of n-grams in the sentence and hypothesis. We computed three separate features using equation 2 for each of the following character n-grams: unigram, bigrams and trigrams. We also use weighted word overlap which uses information content [28].

$$\text{wwo(A,B)} = \frac{\sum_{w \in A \cap B} ic(w)}{\sum_{w' \in B} ic(w')} \tag{4}$$

$$\text{ic(w)} = \left(\ln \frac{\sum_{w' \in C} freq(w')}{freq(w)}\right) \tag{5}$$

Where C is the set of words and *freq(w)* is the occurrence count obtained from the Brown corpus. Our weighted word overlap feature is computed as the harmonic mean of the functions *wwo(A,B)* and *wwo(B,A)*.

POS Overlap: We used the Stanford POS-tagger to tag the words in each sentence with their POS categories. We group the words by the following coarse grained POS categories: nouns, verbs, adjectives and adverbs. We then compare the overlap between these classes of part of speech in each sentence. The POS overlap features are defined as the Jaccard similarity of the two sentences across just the words in a particular POS category:

$$POV_{pos} = \frac{|A_{pos} \cap B_{pos}|}{|A_{pos} \cup B_{pos}|} \tag{6}$$

Where A_{pos} and B_{pos} are the set of terms in POS class *pos* of sentences 1 and 2, respectively. This results in 4 unique POS overlap features.

Dependency Parsing: We used *Stanford Parser* [21] to extract the *subject-verb-object* triples from each sentence. We compare the *subject* and *object* of both sentences. If any of the objects or subjects in a sentence is a NE, we simply compare with the corresponding one from the other sentence by pure string matching. When the same word takes either the role of *subject* or *object* in the two sentences, we assign a flat score of 0.5. If both the *subject* and *object* in the two sentences correspond to the same NEs as in the example above, we assign a score of 1.0. Otherwise, we assign a score of 0.0.

Word Embedding Similarity: Using the embedding from the Word2Vec model trained earlier, we obtained the cosine of the semantic representation of both the text and hypothesis. Where the semantic representation is an additive and multiplicative composition of words in each sentence.

WordNet similarity: Based on the ideas in [20], we obtained WordNet similarity between a text and hypothesis relying both on the path length as well as the depth function.

Negation Words: We use the presence of negation words to determine if entailment exist. First, we curated a list of negation words and also a dictionary of

anti-negation words e.g., *nobody:somebody, no one:someone, nothing:something, no where:somewhere, neither:either* etc, each well mapped to its respective negation word. We normalized all shortened form of a negation word to its true form e.g., *couldn't -¿ could not* etc. In all, we had 22 words. For any negation word found in t, we look for the same in h and viz versa. Using dependency parsing, we compare the role of such word in both. If they both have the same semantic role, we assign a fixed score of 1 and 0 otherwise. Secondly, for any negation word found in h, we look for the presence of its respective anti-negation word from the dictionary in t and also if they bear the same dependency roles, we assign a score 1, otherwise 0. Lastly, we compare the ratio of negation words in t to h, this behaves like overlap measure.

Vector Space based Similarity: We used the feature extraction module of the *scikit-learn* to extract the *TFIDF* weighted feature vectors for the two sentences and then calculated the cosine similarity between them:

$$\cos(A, B) = \frac{\sum_{t=1}^{n} TFIDF(w_{t,A})TFIDF(w_{t,B})}{\sqrt{\sum_{t=1}^{n} w_{t,A}^2 \sum_{t=1}^{n} w_{t,B}^2}} \tag{7}$$

Where A and B are the text and hypothesis.

3.2 Deep Learning Approach

Recently, researchers using algorithms like the CNN [16], Recurrent Neural Networks (RNN) [22], Long Short-Term Memory (LSTM) [13] have reported encouraging results on Language Modeling tasks. Most importantly, [5, 29, 2] showed that deep Neural architecture can match and outperform lexical classifiers which use hand-crafted features for TE. Inspired by these results, we employed a variant of Tree LSTM, the Child-Sum Tree LSTM proposed by Tai et al., [32] and which was previously applied for semantic relatedness task. Tree-LSTMs are specialized LSTMs that adopt the tree-structure topology, i.e., at any given time step t the LSTM is able to compose its states from input vector and hidden states of its child-nodes simultaneously. This is unlike the standard LSTM that assumes a single child per unit. The Child-Sum-Tree LSTMs transition is represented by the following equation:

$$\widehat{h_j} = \sum_{k \in C_{(j)}} h_k \tag{8}$$

$$i_j = \sigma\left(W^{(i)}x_j + U^{(i)}\widehat{h_j} + b^{(i)}\right) \tag{9}$$

$$f_j k = \sigma\left(W^{(f)}x_j + U^{(f)}\widehat{h_k} + b^{(f)}\right) \tag{10}$$

$$o_j = \sigma\left(W^{(o)}x_j + U^{(o)}\widehat{h_j} + b^{(o)}\right) \tag{11}$$

$$u_j = \tanh\left(W^{(u)}x_j + U^{(u)}\widehat{h_j} + b^{(u)}\right) \tag{12}$$

$$c_j = i_j \odot u_j + \sum_{k \in C_(j)} f_{jk} \odot c_k \tag{13}$$

$$h_j = o_j \odot \tanh c_j \tag{14}$$

C_j are the child-nodes in unit j and k $\in$ C_j. In order to generate a semantic representation for a pair of text and hypothesis, each child of a tree is a node from a dependency parse tree of a text or hypothesis as well as its child nodes. For each node, the algorithm takes as input the word vectors. We obtain a representation (h_{txt}, h_{hyp}) from both the text and hypothesis considering the distance and angle of their element-wise summed vectors as well multiplied vectors. We use Neural Network to predict the probability of either +1 or -1 class, where +1 signifies entailment and -1 means Non-entailment. At each node j, we use a softmax classifier to predict the label $\hat{y}_j$ given the inputs x_j observed at nodes in the subtree rooted at j. The classifier takes the hidden state h_j at the node as input, considering both the distance and angle between the pair as shown in the equations below:

$$h_\times = h_{txt} \odot h_{hyp},$$

$$h_+ = h_{txt} + h_{hyp},$$

$$h_\diagdown = |h_{txt} - h_{hyp}|,$$

$$h_s = \sigma\left(W^{(\times)}h_\times + W^{(\times)}h_\diagdown + W^{(+)}h_+ + b^{(h)}\right),$$

$$\widehat{p}\theta = softmax\left(W^{(p)}h_s + b^{(p)}\right),$$

$$\widehat{y_j} = \arg\max_y \widehat{p}\theta(y|x_j) \tag{15}$$

The cost function is given below:

$$J(\theta) = -\frac{1}{m}\sum_{k=1}^{m}(y^{(k)}|x^{(k)}) + \frac{\lambda}{2}||\theta||_2^2 \tag{16}$$

where m = 2, i.e., (YES/NO) and is an L2 norm regularization hyperparameter.

4 Experiments

The COLIEE corpuscontains pairs of sentences, i.e., the civil code article well labeled with an identifier, e.g., 'Article 572' as well as the query question. The Query question is ignored for the first task as it serves as the hypothesis for entailment in the second task. In all, 296 civil code articles each with distinct ID was released as training data for task one while 490 pairs of civil code articles an query was made available as training data for the second task. Furthermore, task 1 has 95 test data while task 2 has 66 test data. Since the organizers did not release the official Gold standard for the tasks and/or the official performance evaluation of the submitted systems, we only report an evaluation on the 2015 dataset.

Due to the setup of our IR system, more attention is focused on precision than recall. For instance, since we approached it as a classification task, the system only predict an article per query, i.e. even if a query has more than one related articles, the model only select the one with highest confidence score. In a way, this can also reduce recall but many unrelated articles can be avoided from the result set. For the purpose of evaluation, we simply use accuracy score, which is calculated as the number of correct prediction over the total number of prediction, i.e., if out of 10 queries, the system got 4 correctly, then the accuracy 0.40. Tables 1, 3 summarizes the result obtained on task 1 and 2. In particular, Table 3 shows the result on a 5-fold cross validation run for the second subtask. As shown in table 3, we used the result reported by Kim et al., [17] as our baseline score since their evaluation was on the same COLIEE data. We observe a slightly better performance from the DL approach which even outperforms the baseline system. We performed an ablation test on the features designed in order to see the best mix. To do this, we made 3 runs by randomly withholding 3 features per time. The table 2 shows the performance of the SVM for each runs.

Model	No of Train Article	No of Test Query	Accuracy
Run1 (SVM)	296	79	0.521
Run2 (RF)	296	79	0.560
Run3 (KM)	296	79	0.481

Table 1. Task 1 Evaluation

Model	Withheld Features	Accuracy
Run1 (SVM)	Word Embedding Similarity, POS overlap, Dependency parsing	0.460
Run2 (SVM)	Negation Words 1 and 2, word overlap	0.51
Run3 (SVM)	Negation Words 2, Word Embedding Similarity, Vector space similarity	0.394

Table 2. Feature Ablation Evaluation on Task 1

Model	Train	Test
SVM (Features+)	78.20	68.40
RF (Features+)	76.00	64.19
Child-Sum-Tree LSTM	81.33	70.10
Baseline (Kim et al.,) [17]	-	55.87

Table 3. Task 2 Evaluation on COLIEE dataset

We implemented a child-Sum Tree LSTM as proposed by [32]. As earlier stated, we used our pre-trained word embedding as input vector to the LSTM, instead of one-hot encoding representation with sequence to sequence learning. Our embedding has 50 dimension.

For implementation, we used the Keras[9] Deep Learning library . Our LSTM model has an hidden layer size of 300. We used a batch-size of 25. We used ADAM, a stochastic optimizer with learning rate set at 0.01 and a decay value of 1e-4 as well as momentum of 0.9. Typically, we set the number of training epochs to 200. However, throughout the experiments, we track the model for over-fitting by monitoring the peak accuracy on validation data as well as the epoch where the model does not seem to learn sufficiently again and accuracy begins to drop or does not increase anymore. Usually, we pick this epoch for our test runs. Since the organizers did not include any validation set, we divided the test data into two. We actually tested on half while the other half was used as the validation set. As observable, our system is able to generalize even with little training data. It should be noted that the quality of embedding used as input also influences the result obtained. However, we did not verify how the use of a bigger pretrained embedding like the Glove vectors [27] can improve performance

5 Conclusion

The paper describes our team's attempt at solving some of the challenges proposed by COLIEE2016 organizers. We participated in two tasks, i.e. IR and TE which is receiving lots of attention in NLP domain generally. For the first task, we submitted 3 runs, each based on SVM, RF and K-means clustering which was used as the baseline. In the second task, we designed some novel features combined and fit into the labels with SVM and RF algorithms. Secondly, we adapted the Child-sum tree LSTM for the textual entailment task. The result obtained on a 5-fold cross validation shows that the DL system slightly outperformed the SVM and RF systems.

Acknowledgments. Kolawole J. Adebayo has received funding from the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science

[9] Keras is a modular but powerful Deep Learning Library built on top of Theano and Tensorflow. Available at `https://github.com/fchollet/keras`

and Technology. Luigi Di Caro and Guido Boella have received funding from the European Union's H2020 research and innovation programme under the grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

1. Leif Azzopardi, Mark Girolami, and CJ Van Rijsbergen. Topic based language models for ad hoc information retrieval. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 3281–3286. IEEE, 2004.
2. Petr Baudiš and Jan Šedivỳ. Sentence pair scoring: Towards unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*, 2016.
3. Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. The seventh pascal recognizing textual entailment challenge. *Proceedings of TAC*, 2011, 2011.
4. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
5. Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.
6. Elena Cabrio, Julien Cojan, Alessio Palmero Aprosio, Bernardo Magnini, Alberto Lavelli, and Fabien Gandon. Qakis: an open domain qa system based on relational patterns. In *Proceedings of the 2012th International Conference on Posters & Demonstrations Track-Volume 914*, pages 9–12. CEUR-WS. org, 2012.
7. Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.
8. Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.
9. Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. Modeling interestingness with deep neural networks, June 13 2014. US Patent App. 14/304,863.
10. Miguel Angel Ríos Gaona, Alexander Gelbukh, and Sivaji Bandyopadhyay. Recognizing textual entailment using a machine learning approach. In *Mexican International Conference on Artificial Intelligence*, pages 177–185. Springer, 2010.
11. Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. Concrete sentence spaces for compositional distributional models of meaning. In *Computing Meaning*, pages 71–86. Springer, 2014.
12. John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
13. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
14. Sergio Jimenez, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal. Unal-nlp: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, 2014.

15. Adebayo Kolawole John, Luigi Di Caro, and Guido Boella. Normas at semeval-2016 task 1: Semsim: A multi-feature approach to semantic text similarity. *Proceedings of SemEval*, pages 718–725, 2016.
16. Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.
17. Mi-Young Kim, Ying Xu, and Randy Goebel. A convolutional neural network in legal question answering.
18. Mi-Young Kim, Ying Xu, and Randy Goebel. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*, pages 244–258. Springer, 2014.
19. Jens Lehmann, Tim Furche, Giovanni Grasso, Axel-Cyrille Ngonga Ngomo, Christian Schallhart, Andrew Sellers, Christina Unger, Lorenz Bühmann, Daniel Gerber, Konrad Höffner, et al. Deqa: deep web extraction for question answering. In *International Semantic Web Conference*, pages 131–147. Springer, 2012.
20. Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1138–1150, 2006.
21. Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
22. LR Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 2001.
23. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
24. Shachar Mirkin, Ido Dagan, and Eyal Shnarch. Evaluating the inferential utility of lexical-semantic resources. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 558–566. Association for Computational Linguistics, 2009.
25. Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *ACL*, pages 236–244, 2008.
26. Jeff Mitchell and Mirella Lapata. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 430–439. Association for Computational Linguistics, 2009.
27. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.
28. Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.
29. Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.
30. Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448. Association for Computational Linguistics, 2012.
31. Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014.

32. Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.
33. Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

Legal Question Answering Using Paraphrasing and Entailment Analysis

Mi-Young Kim[1], Ying Xu[1], Yao Lu[2], and Randy Goebel[1]

Dept. of Computing Science, University of Alberta, Edmonton, AB, Canada[1]
iLab Tongji, School of Software Engineering, Tongji University[2]

{ miyoung2,yx2,rgoebel}@ualberta.ca[1]
95luyao@tongji.edu.cn[2]

Abstract. Our legal question answering system combines legal information retrieval and textual entailment, and we describe a system that exploits paraphrasing and sentence-level analysis of queries and legal statutes. We have evaluated our system using the training data from the competition on legal information extraction/entailment (COLIEE)-2016. The competition focuses on the legal information processing required to answer yes/no questions from Japanese legal bar exams, and it consists of two phases: legal ad-hoc information retrieval (Phase 1), and textual entailment (Phase 2). Phase 1 requires the identification of Japan civil law articles relevant to a legal bar exam query. For this phase, we have used an information retrieval approach using TF-IDF and a Ranking SVM. Phase 2 requires decision on yes/no answer for previously unseen queries, which we approach by comparing the approximate meanings of queries with relevant articles. Our meaning extraction process uses a selection of features based on a kind of paraphrase, coupled with a condition/conclusion/exception analysis of articles and queries. We also identify synonym relations using word embedding, and detect negation patterns from the articles. Our heuristic selection of attributes is used to build an SVM model, which provides the basis for making a decision on the yes/no questions. Experimental evaluation demonstrates the value of our method, and the results show that our method outperforms previous methods.

Keywords: legal question answering, recognizing textual entailment, information retrieval, paraphrasing

1. Task Description

Our approach to legal question answering combines information retrieval and textual entailment. We achieve this combination with a number of intermediate steps. For instance, consider the question “Is it true that a special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty.” A system must first identify and retrieve relevant documents (typically legal statutes), and subsequently, identify a most relevant sentence. Finally it must extract and

compare semantic connections between the question and the relevant sentences, and confirm a threshold of evidence about whether an entailment relation holds.

The Competition on Legal Information Extraction/Entailment (COLIEE) 2016[1] focuses on two aspects of legal information processing related to answering yes/no questions from legal bar exams: a legal document retrieval (Phase 1), and whether there is a textual entailment relation between a query and relevant legal documents (Phase 2).

We treat Phase 1 as an ad-hoc information retrieval (IR) task. The goal is to retrieve relevant Japan civil law statutes or articles that are related to a question in legal bar exams, from which we can confirm a yes or no answer based on deciding if there is an entailment relation between the question (or the negation of the question) and the relevant statutes.

We approach the information retrieval part of this problem (Phase 1) with two models based on statistical information. One is the TF-IDF model [1], i.e., term frequency-inverse document frequency. The idea is that relevance between a query and a document depends on their intersecting word set. The importance of words is measured with a function of term frequency and document frequency as parameters. Our terms are lemmatized words, which mean verbs like "attending," "attends," and "attended" are lemmatized as the same form "attend."

Another popular model for text retrieval is a Ranking SVM model [2]. We use that model to re-rank documents that are retrieved by the TF-IDF model. The model's features are lexical words, dependency path bigrams and TF-IDF scores. The intuition is that the supervised model can learn weights or priority of words based on training data in addition to, or as an alternative to TF-IDF.

The goal of Phase 2 is to construct yes/no question answering systems for legal queries, by heuristically confirming entailment of a query (or its negation) from relevant articles. The answer to a question is typically determined by measuring some kind of semantic similarity between question and answer. Because the legal bar exam query and relevant articles are complex and varied, we need to carefully determine what kind of information is needed for confirming textual entailment. Here we exploit a kind of paraphrasing based on term expansion and word embedding for semantic analysis, coupled with condition/conclusion/exception analysis on the query and relevant articles. After constructing a set of pre-trained semantic word embeddings using *word2vec*[2], we train the model to learn models for semantic matching between question and corresponding articles. These feature extraction methods are coupled with negation analysis, then used to construct an SVM model to provide the required yes/no answers.

2. Phase 1: Legal Information Retrieval

2.1 IR Models

[1] https://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

[2] code.google.com/p/word2vec

[3] Lucene can be downloaded from http://lucene.apache.org/core/.

Our information retrieval model is a combination of the term frequency–inverse document frequency (tf-idf) model and a support vector machine (SVM) re-ranking model. We will describe the two components in the following.

2.1.1 the tf-idf model

One of our baseline models is a tf-idf model implemented in Lucene, an open source IR system[3].

The simplified version of Lucene's similarity score of an article to a query is:

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

The score *tf-idf*(*Q,A*) is a measure which computes the relevance between a query *Q* and an article *A*. First, for every term *t* in the query *A*, we compute *tf*(*t,A*), and *idf*(*t*). The score *tf*(*t,A*) is the term frequency of *t* in the article *A*, and *idf*(*t*) is the inverse document frequency of the term *t*, which is the number of articles that contain *t*. The final score is the sum of the scores of terms in both the article and the query. The bigger *tf-idf*(*Q,A*), the more relevance between the query *Q* and the article *A*.

The choice of terms in documents is as important as choosing the score functions. Instead of using the original words in a text, we lemmatize the text with the Stanford NLP tool [15]. After lemmatization, words such as *steal, stole, and steals* become *steal*. In this way, if there is *steal* in the question, but *stole* in the article, we can still retrieve the article as a match.

2.1.2 The ranking SVM

Previous tf-idf models rank the articles based on frequency information. However, other features, such as the matched phrases between the article and the queries, are useful too. We use an SVM Ranking model to learn the importance of such features and then re-estimate the score of each retrieved article from the tf-idf output.

The ranking SVM model was proposed by [2]. The model ranks the documents based on user's click through data, in our case, the correct articles in the training data. Given the articles retrieved from the tf-idf model, ranking SVM will learn to rank correct articles higher than incorrect ones. More precisely, given the feature vector of a training instance, i.e. a retrieved article set given a query, denoted by $\Phi(Q,A_i)$, the model tries to find a ranking that satisfies constraints:

$$\Phi(Q, A_i) > \Phi(Q, A_j) \quad (2)$$

where A_i is a relevant article for the query Q, while A_j is less relevant.

To use this ranking SVM, we incorporate the following types of features:

- Lexical words: the lemmatized normal form of surface structure of words in both the retrieved article and the query. In the conversion to the SVM's

[3] Lucene can be downloaded from http://lucene.apache.org/core/.

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, arising from a dependency parsing. The intuition is that, compared with the bag of words information, syntactic information should improve the capture of salient semantic content. Dependency parse features have been used in many NLP tasks, and improved IR performance [3]. This feature type is also converted into binary values.
- TF-IDF score (Section 2.1.1).

2.2 Experiments

The COLIEE legal IR task has several sets of queries with the Japan civil law articles as documents (1044 articles in total). Here follows one example of the query and a corresponding relevant article.

Question: A person who made a manifestation of intention which was induced by duress emanated from a third party may rescind such manifestation of intention on the basis of duress, only if the other party knew or was negligent of such fact.

Related Article: (Fraud or Duress) Article 96(1) Manifestation of intention which is induced by any fraud or duress may be rescinded. (2)In cases any third party commits any fraud inducing any person to make a manifestation of intention to the other party, such manifestation of intention may be rescinded only if the other party knew such fact.(3)The rescission of the manifestation of intention induced by the fraud pursuant to the provision of the preceding two paragraphs may not be asserted against a third party without knowledge.

Before the final test set was released, we received 8 sets of queries for a dry run. The 8 sets of data include 412 queries. We used the corresponding 8-fold leave-one-out cross validation evaluation. The metric for measuring our IR models is Mean Average Precision (MAP):

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{m}\sum_{k\in(1,m)}\mathrm{precision}(R_k) \tag{3}$$

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 first until the *k-th* article. In the following experiments, we set m as 3 for all queries, corresponding to the column MAP@3 in Table 1. SVM's parameters are set according to the 8-fold cross validation IR performance. Given the top 20 articles returned by the tf-idf model, the SVM model extracts features for every article and trains according to the order that correct articles are ranked higher than incorrect ones. Table 1 presents the results of different models. The result shows that the ranking SVM with all three features achieves the highest performance. We also show the standard deviation of the cross-validation, the smallest and largest MAP@3 for 8 folds to show the effect of small training data. It seems the TF-IDF model causes larger deviation than the SVM models.

Table 1. IR results on dry run data with different models.

Id	models	MAP@3	Standard deviation	Smallest	Largest
1	tf-idf with lemma	39.8	7.0	23.8	45.5
2	SVM-ranking with lemma	39.8	6.5	26.1	46.8
3	SVM-ranking with lemma and dependency pair	41.2	5.4	30.1	48.4
4	model 3 plus tf-idf score	43.1	7.1	27.2	49.0

3. Phase 2: Answering yes/no Questions

Our system uses word embedding for semantic analysis and paraphrasing for term expansion to predict textual entailment. We will look at the entailment types and extract features from sentences, as described in detail in the next subsections.

3.1 Entailment types

We identify a variety of types of entailment as shown in Table 2. By classifying a yes/no problem as one of these types, we can determine what kind of further information is required to provide a decision on entailment.

Table 2. Query-Article types

Query-Article type	Proportion	Query-Article type	Proportion
One article refers to another article	0.182	Question is a specific example	0.092
Multiple relevant articles	0.388	Multiple conditions	0.731
Exceptional case	0.148		

Table 2 shows our list of query-article types. Note that one article can refer to another, such as "*If there is any latent defect in the subject matter of a sale, the provisions of Article 566 shall apply mutatis mutandis."* This makes textual entailment more complex as we also need to analyze the meaning of the referred article.

In another case, one query can have multiple relevant articles, so we have to combine the multiple articles' meanings or choose one as most relevant for determining entailment.

Note also that many statutes have exceptional cases, so we need to recognize if the query is included in the exceptional case or not. In addition, a query may be one example of the article case. There are also cases where some articles have multiple conditions for one conclusion, so we must then confirm if each condition is satisfied in the query. Overall, many query-article types also require the identification of negation and synonym/antonym relations to confirm the correct entailment.

The overall description of our procedure of textual entailment is as follows:

1. Find the most relevant article with a query
2. Divide a query and the corresponding article into "Condition(s)," "Conclusion," and "Exception-condition(s)."
3. Term expansion using Paraphrasing
4. Negation and synonym detection
5. Extract features and perform learning using the features

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

3.2 Finding the most relevant article/sentences

In case that there are multiple relevant sentences, we choose the article with the most overlapping words with the query. In the chosen article, if there exist multiple regulations, we also choose the one regulation that has most overlapping words with a query.

3.3 Negation and synonym detection

We exploit a process for managing negation and antonyms as described in Kim et al. [10]. In addition, we approximate word semantic similarity by converting words to vector representations using the *word2vec* tool. The output of the semantic similarity is vector similarity. We used 1,044 legal law articles to train the *word2vec* tool (*vector dimension*=50).

3.4 Condition/Conclusion/Exception detection

From our analysis of the structure of statutes we extract components based on the following rules:

$$conclusion := segment_{last}(sentence, keyword),$$

$$condition := \sum_{i \neq last} segment_i(sentence, keyword),$$

$$condition := condition \text{ [or] } condition$$

$$condition := sub\_condition \text{ [and] } sub\_condition$$

$$exception\_conclusion := segment_{last}(sentence, exception\_keyword),$$

$$exception\_condition := \sum_{i \neq last} segment_i(sentence, exception\_keyword),$$

$$exception\_condition := exception\_condition \text{ [or] } exception\_condition$$

$$exception\_condition := sub\_exception\_condition \text{ [and] } sub\_exception\_condition$$

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

The original bar law examinations in the COLIEE data are provided in Japanese and English, and our initial implementation used a Korean translation, provided by the Excite translation tool.[4] 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. Therefore, the above rules may not be appropriate for all English sentences, because the segment order can differ.

The following is an example of condition and conclusion detection:

> <Civil law example>A person who employs others for a certain business, shall be liable for damages inflicted on a third party by his/her employees with respect to the execution of that business; Provided, however, that this shall not apply, if the employer exercised reasonable care in appointing the employee or in supervising the business, or if the damages could not have been avoided even if he/she had exercised reasonable care.
>
> *(1) Conclusion* => shall be liable for damages inflicted on a third party by his/her employees with respect to the execution of that business.
> *(2) Condition* => A person who employs others for a certain business
> *(3) Exception*
> *Conclusion* => this shall not apply (opposite of main conclusion)
> *Condition*
> *Condition* =>
> *Condition* => if the employer exercised reasonable care in appointing the employee
> *Condition (OR)* => in supervising the business

[4] http://excite.translation.jp/world/

Condition(OR) => if the damages could not have been avoided even if he/she had exercised reasonable care.

3.5 Term expansion using paraphrasing

There are many words with similar meanings but different lexical forms (e.g., 'obligor' vs. 'debtor', 'rescind' vs. 'cancel', 'lien' vs. 'privilege', etc.). To resolve these diverse terms, we use language translation-based paraphrasing. The idea of translation-based paraphrase is that translating from one language to another and then back, will often produce semantically similar but lexically distinct outputs. If we assume that the language translations preserve semantics, more or less, then lexically distinct terms can be considered as paraphrases. In our application of this idea, we translate the original English query/document into German, and then back-translate the German sentences into English. We then can detect pairs of words/phrases which can be considered as semantically related: the original English sentence and double-translated English sentence. We used Google translate, and chose German as the pivot language, which is a closely related to English, which we hope reduces the translation errors.

We performed double translation with 100 article laws in the Japanese Civil Code. We used the monolingual alignment tool of Sultan et al. [12] to create automatic word alignments in English. Table 3 shows examples of detected paraphrases using language translation. We can see that it also detects plural forms and past tense forms, in addition to words with similar meanings. We extract the top 100 paraphrases, and manually extracted corresponding Korean words in the Korean-translated Query-Article text.

Table 3 Examples of Detected Paraphrases

Original word	Paraphrased word	Original word	Paraphrased word
year	years	establishes	sets
makes	made	purpose	aim
warranties	guarantees	matter	area
released	relieved	pledge	commitment
assigned	transferred	demand	claim
respect	relation	referred	designated

3.6 Supervised learning with SVM

Since we cannot anticipate the impact of each linguistic attribute, we run a machine learning algorithm that learns what information is relevant in the text to achieve our goal. We have compared our method with SVM, as a kind of supervised learning model. Using the SVM tool included in the Weka [4] software library, we performed cross-validation for the 412 questions. We used a linear kernel SVM because it is popular for real-time applications as they enjoy both faster training and classification speeds (significantly reduced memory requirements compared with non-

linear kernels, because of the compact representation of the decision function). We used the following features:

a) Word Lemma
b) Lexical semantic features
c) Negation feature
d) Sentence analysis feature (condition, conclusion, and exception)

For concept features, we have exploited word embedding using *word2vec*. When we use word embedding, we assume the concepts of two words are the same if their vector cosine similarity >=0.8.

The detailed features that we use are as follows:

Feature 1: $If\ \exists_{i,j}\{(\text{concept}(w_i)\ ,\ \text{Query}_{\text{condition}})\ \bigcap\ (\text{concept}(w_j)\ ,\ \text{Article}_{\text{condition}})\}$

Feature 2: $If\ \exists_{i,j}\{(\text{concept}(w_i)\ ,\ \text{Query}_{\text{conclusion}})\ \bigcap\ (\text{concept}(w_j)\ ,\ \text{Article}_{\text{conclusion}})\}$

Feature 3: $If\ \exists Article_{sub\_condition}\bigcap$
$\exists_{i,j,k}\{(\text{concept}(w_i)\ ,\ \text{Query}_{\text{condition}})\ \bigcap\ (\text{concept}(w_j)\ ,\ \text{Article}_{\text{sub\_condition}k})=\varnothing\}$

Feature 4: $If\ \exists_{i,j}\{(\text{concept}(w_i)\ ,\ \text{Query}_{\text{condition}})\ \bigcap\ (\text{concept}(w_j)\ ,\ \text{Article}_{\text{exception\_condition}})\}$

Feature 5: $If\ \exists Article_{sub\_exception\_condition}\bigcap$
$\exists_{i,j,k}\{(\text{concept}(w_i)\ ,\ \text{Query}_{\text{condition}})\ \bigcap\ (\text{concept}(w_j)\ ,\ \text{Article}_{\text{sub\_exception\_condition}k})=\varnothing\}$

Feature 6 : $If\ neg\_level(\text{Query}_{\text{condition}})\ =\ neg\_level(\text{Article}_{\text{condition}})$

Feature 7 : $If\ neg\_level(\text{Query}_{\text{conclusion}})\ =\ neg\_level(\text{Article}_{\text{conclusion}})$

Feature 8 : $If\ neg\_level(\text{Query}_{\text{condition}})\ =\ neg\_level(\text{Article}_{\text{exception\_condition}})$

Features 1 and 2 check if there are overlapped concepts between a query condition (conclusion) and its relevant article condition (conclusion). Feature 3 checks if there is overlapped word between a query condition and its relevant article sub-condition. Because the article sub-condition is connected with other sub-condition(s), using "and" as a connector, the query should include the meanings of all the article sub-conditions. Feature 4 checks if there are overlapped concepts between a query condition and its article exception-condition. We want to check if the query is included in the exceptional case using the feature. Feature 5 checks if there is no overlapped word between a query condition and its relevant article sub-exception-condition. Features 6, 7, and 8 check the negation levels between the query condition, article condition, query conclusion, article conclusion, and article exception-condition. We compute negation value (neg-level()) according to Kim et al. [10].

4. Phase 2: Experimental Results

4.1 Experimental setup for Phase 2

Table 4. Experimental results on dry run data for Phase 2

Method	Accu(%)
(a) Baseline	51.70
(b) Our method using cross-validation with Supervised learning (SVM) not using word embedding but using lexical word itself	62.14
(c) Our method using cross-validation with Supervised learning (SVM) using word embedding	60.67
(d) Cross-validation using Kim et al. [10]	60.92
(e) Cross-validation using Kim et al. [11]	61.65
Without term expansion using paraphrasing from (b)	60.44
Without neg_level() from (b)	49.27

In the general formulation of the textual entailment problem, given an input text sentence and a hypothesis sentence, the task is to make predictions about whether or not the hypothesis is entailed by the input sentence. We report the accuracy of our method in answering yes/no questions of legal bar exams by predicting whether the questions are entailed by the corresponding civil law articles.

There is a balanced positive-negative sample distribution in the dataset (51.70% yes, and 48.30% no) for a dry run of COLIEE 2016 dataset, so we consider the baseline for true/false evaluation is the accuracy when always returning "yes," which is 51.70%. Our total data for the dry run has 412 questions.

4.2 Experimental results

Table 4 shows the experimental results. An SVM-based model showed accuracy of 62.14% when we did not use word embedding but used the lexical form of each word; the method of Kim et al. [10] showed 60.92% and that of Kim et al. [11] showed 61.65%. Our SVM augmented system outperformed Kim et al. [10, 11].

Table 4 also shows the experimental results arising from changing some features in our method. The accuracy was lower by 1.70% when we removed paraphrasing, and the accuracy was lower by 1.47% when we used word embedding, which suggests that word embedding does not help capture the semantics better than the lexical word itself. When we did not use the negation feature, the accuracy became lower by 12.87%, which proves that the negation feature is most valuable.

4.3 Error analysis

From unsuccessful instances, we manually classified the error types as shown in Table 5. The biggest error arises, of course, from the semantic similarity error, which we believe our word embedding is not enough to capture the semantic similarity. We will try to include the bar exam text in the training data for the word embedding. The second biggest error is because of complex constraints in conditions. As with the other error types, there are cases where a question is an example case of the corresponding article, and the corresponding article embeds another article. We also found cases that indicate the need to do more extensive temporal analysis.

Table 5. Error types

Error type	Accuracy (%)	Error type	Accu.(%)
Specific example case	9.62	Semantic similarity error	28.85
Incorrect detection of the most similar article sentence	10.90	Constraints in condition	25.00
Incorrect detection of Condition, Conclusion, and mismatch	11.54	Etc.	14.10

It will be interesting if we compare our performance using Korean-translated sentences with that using original Japanese sentences. We would expect the system using original sentences will show improved performance, because there exist no translation errors. As future work, we will construct a Japanese system using paraphrase/synonym/antonym dictionaries for Japanese, and then analyze how the translation affects performance.

5 Related work

A previous textual entailment method from W. Bdour et al. [5] provided the basis for a yes/no Arabic question answering system. They used a kind of logical representation, and compared the logical representation between queries and documents. This method will be appropriate for the task where queries and documents have similar logical representations so it is easier to confirm entailment from one logical representation to another. However, our task's entailment type is more complex, so we take an approach that approximates the logical content of queries and documents, rather than attempt any complete transformation to a logical form.

Nielsen et al. [6] extracted features from dependency paths, and combined them with word-alignment features in a mixture of an expert-based classifier. Zanzotto et al. [7] proposed a syntactic cross-pair similarity measure for RTE. Harmeling [8] took a

similar classification-based approach with transformation sequence features. Marsi et al. [9] described a system using dependency-based paraphrasing techniques.

Many methods have been proposed for paraphrasing. One of the methods is the idea of semantic parsing via paraphrasing [13]. They transform a sentence into a logical form, and then convert logical forms to canonical form using the Freebase database. Subsequently, they obtain an association between the original sentence and canonical forms. However, hundreds of logical/canonical forms have been generated per sentence in their method, and the method does not show how to choose the best amongst them.

The method of Zhang et al. [14] also uses a pivot language for paraphrasing. Like us, they translate one language to another, then re-translate from the translated language into the original language. They then obtain a paraphrasing set between the original utterance and double-translated utterance. They showed improved performance in paraphrase detection using the pivot language translation, so we also employ the language translation-based paraphrasing. But instead of their use of the GIZA++ alignment, we used the monolingual alignment tool of Sultan et al. [12], because GIZA++, which is for alignment between two different languages, did not show good performance for our dataset.

6 Conclusion

We have described our most recent implementation for the Competition on Legal Information Extraction/Entailment (COLIEE)-2016 Task.

For Phase 1, legal information retrieval, we implemented a Ranking-SVM model for the legal information retrieval task. By incorporating features such as lexical words, dependency links, and TF-IDF score, our model shows better mean average precision than TF-IDF.

For Phase 2, we have proposed a method to answer yes/no questions from legal bar exams related to civil law. We used a SVM model using paraphrasing and pre-trained word embedding and query/article condition/conclusion/exception analysis. We show improved performance over a previous system, and paraphrasing and negation detection contributed to the performance.

Acknowledgements

This research was supported by the Alberta Innovates Centre for Machine Learning (AICML). We are indebted to Ken Satoh of the National Institute for Informatics, who has had the vision to create the COLIEE competition.

References

1. K. Sparck Jones, A statistical interpretation of term specicity and its application in retrieval. In: Willett, P. (ed.) Document Retrieval Systems, pp. 132-142. Taylor Graham Publishing, London, UK, UK, 1988
2. T. Joachims, Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 133-142. KDD '02, ACM, New York, NY, USA, 2002
3. K.T. Maxwell, J. Oberlander, W.B. Croft. Feature-based selection of dependency paths in ad hoc information retrieval. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 507-516. Association for Computational Linguistics, Sofia, Bulgaria, August 2013
4. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. 2009
5 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
6. 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
7. 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
8. 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
9. 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
10. M-Y. Kim, Y. Xu, and R. Goebel. Alberta-KXG: Legal Question Answering Using Ranking SVM and Syntactic/Semantic Similarity. JURISIN Workshop, 2014
11. M-Y. Kim, Y. Xu, and R. Goebel. A Convolutional Neural Network in Legal Question Answering. JURISIN Workshop 2015.
11. M. A. Sultan, S. Bethard, and T, Sumner. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. Transactions of the Association for Computational Linguistics, 2, pp. 219-230, 2014
12. J. Berant, and L. Percy. Semantic Parsing via Paraphrasing, Proceedings of the conference of the Association for Computational Linguistics (ACL), pp. 1415-1425, 2014
13. W. Zhang, Z. Ming, Y. Zhang, T. Liu and T. S. Chua. Exploring Key Concept Paraphrasing Based on Pivot Language Translation for Question Retrieval. In AAAI, pp. 410-416, 2015
14. Christopher D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. *The Stanford CoreNLP Natural Language Processing Toolkit*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60, 2014

Legal Question Answering using Ranking SVM and Deep Convolutional Neural Network

Phong-Khac Do[1], Huy-Tien Nguyen[1] Chien-Xuan Tran[1],
Minh-Tien Nguyen[1,2], and Minh-Le Nguyen[1]

[1] School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
[2] Hung Yen University of Education and Technology (UTEHY), Hung Yen, Vietnam.
{phongdk, ntienhuy, chien-tran, tiennm, nguyenml}@jaist.ac.jp

Abstract. This paper presents a study of employing Ranking SVM and Convolutional Neural Network for two missions: legal information retrieval and question answering in the Competition on Legal Information Extraction/Entailment. From our perspective, a simple expansion of an article with the content of referential one is not always beneficial, so in our experiment, to simplify, we just ignore the references and citations. Additionally, only single-paragraph articles are used instead of multiple-paragraph ones. The result demonstrates that our model Ranking SVM outperforms other methods with respect to legal information retrieval task. For the legal question answering task, we found that additional statistical features integrated into Convolutional Neural Network results higher accuracy.

Keywords: Learning to Rank, Ranking SVM, Convolutional Neural Network (CNN), Legal Information Retrieval, Legal Question Answering.

1 Introduction

Legal text, along with other natural language text data, e.g. scientific literature, news articles or social media, has seen an exponential growth on the Internet and in specialized systems. Unlike other textual data, legal texts contain strict logical connections of law-specific words, phrases, issues, concepts and factors between sentences or various articles. Those are for helping people to make a correct argumentation and avoid ambiguity when using them in a particular case. Unfortunately, this also makes information retrieval and question answering on legal domain become more complicated than others.

There are two primary approaches to information retrieval (IR) in the legal domain [1]: manual knowledge engineering (KE) and natural language processing (NLP). In the KE approach, an effort is put into translating the way legal experts remember and classify cases into data structures and algorithms, which will be used for information retrieval. Although this approach often yields a good result, it is hard to be applied in practice because of time and financial cost

when building the knowledge base. In contrast, NLP-based IR systems are more practical as they are designed to quickly process terabytes of data by utilizing NLP techniques. However, several challenges are presented when designing such system. For example, factors and concepts in legal language are applied in a different way from common usage [2]. Hence, in order to effectively answer a legal question, it must compare the semantic connections between the question and sentences in relevant articles found in advance [3].

Given a legal question, retrieving relevant legal articles and deciding whether the content of a relevant article can be used to answer the question are two vital steps in building a legal question answering system. Kim et al. [3] exploited Ranking SVM with a set of features for legal IR and Convolutional Neural Network (CNN) [11] combining with linguistic features for question answering (QA) task. However, generating linguistic features is a non-trivial task in the legal domain. Carvalho et al. [2] utilized n-gram features to rank articles by using an extension of TF-IDF. For QA task, the authors adopted AdaBoost [19] with a set of similarity features between a query and an article pair [18] to classify a query-article pair into "YES" or "NO". However, overfitting in training may be a limitation of this method. Sushimita et al. [6] used the voting of Hiemstra, BM25 and PL2F for IR task. Meanwhile, Tran et al. [7] used Hidden Markov model (HMM) as a generative query model for legal IR task. Kano [8] addressed legal IR task by using a keyword-based method in which the score of each keyword was computed from a query and its relevant articles using inverse frequency. After calculating, relevant articles were retrieved based on three ranked scores. These methods, however, lack the analysis of feature contribution, which can reveal the relation between legal and NLP domain. This paper makes the following contributions:

- We conduct detailed experiments over a set of features to show the contribution of individual features and feature groups. Our experiments benefit legal domain in selecting appropriate features for building a ranking model.
- We analyze the provided training data and conclude that: (i) splitting legal articles into multiple single-paragraph articles and (ii) carefully initializing parameters for CNN significantly improve the performance of legal QA system.
- We propose to classify a query-article pair into "YES" or "NO" by voting, in which the score of a pair is generated from legal IR and legal QA model.

In the following sections, we first show our idea along with data analysis in the context of COLIEE. Next, we describe our method for legal IR and legal QA tasks. After building a legal QA system, we show experimental results along with discussion and analysis. We finish by drawing some important conclusions.

2 Proposed Method

2.1 Basic Idea

In the context of COLIEE 2016, our approach is to build a pipeline framework which addresses two important tasks: IR and QA. In Figure 1, in training phase,

a legal text corpus was built based on all articles. Each training query-article pair for LIR task and LQA task was represented as a feature vector. Those feature vectors were utilized to train a learning-to-rank (L2R) model (Ranking SVM) for IR and a classifier (CNN) for QA. The red arrows mean that those steps were prepared in advance. In the testing phase, given a query q, the system extracts its features and computes the relevance score corresponding to each article by using the L2R model. Higher score yielded by SVM-Rank means the article is more relevant. As shown in Figure 1, the article ranked first with the highest score, i.e. 2.6, followed by other lower score articles. After retrieving a set of relevant articles, CNN model was employed to determine the "YES" or "NO" answer of the query based on these relevant articles.

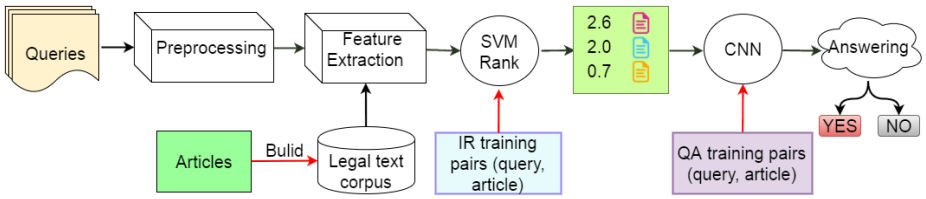


Fig. 1. The proposed model overview

2.2 Data Observation

The published dataset in COLIEE 2016[3] consists of a text file containing Japanese Civil Code and eight XML files. Each XML file contains multiple pairs of queries and their relevant articles, and each pair has a label "YES" or "NO", which confirms the query corresponding to the relevant articles. There is a total of 412 pairs in eight XML files and 1105 articles in the Japanese Civil Code file, and each query can have more than one relevant articles.

After analyzing the dataset in the Civil Code file, we realized that the content of a query is often more or less related to only a paragraph of an article instead of the entire content. Based on that, each article was treated as one of two types: single-paragraph or multiple-paragraph, in which a multiple-paragraph article is an article which consists of more than one paragraphs. There are 7 empty articles, 682 single-paragraph articles and the rest are multiple-paragraph.

Based on our findings, we proposed to split each multiple-paragraph article into several independent articles according to their paragraphs. For instance, in Table 1, the Article 233 consisting of two paragraphs was split into two single-paragraph articles 233(1) and 233(2). After splitting, there are in total 1663 single-paragraph articles.

Stopwords were also removed before building the corpus. Text was processed in the following order: tokenization, POS tagging, lemmatization, and stopword removal[4]. In [2], the *stopword removal* stage was done before the *lemmatization*

[3] http://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/
[4] http://www.nltk.org/book/ch02.html

Table 1. Splitting a multiple-paragraph article into some single-paragraph articles

Article ID	Content
233	(1) If a tree or bamboo branch from neighboring land crosses a boundary line, the landowner may have the owner of that tree or bamboo sever that branch. (2) If a tree or bamboo root from neighboring land crosses a boundary line, the owner of the land may sever that root.
233(1)	If a tree or bamboo branch from neighboring land crosses a boundary line, the landowner may have the owner of that tree or bamboo sever that branch.
233(2)	If a tree or bamboo root from neighboring land crosses a boundary line, the owner of the land may sever that root.

stage, but we found that after lemmatizing, some words might become stopwords, for instance, "done" becomes "do". Therefore, the extracted features based on words are more prone to be distorted, leading to lower ranking performance if *stopword removal* is carried out before *lemmatization* step. Terms were tokenized and lemmatized using NLTK[5], and POS tagged by Stanford Tagger[6].

2.3 Legal Question Answering

Legal Information Retrieval

In order to build a legal IR, traditional models such as TF-IDF, BM25 or PL2F can be used to generate basic features for matching documents with a query. Nevertheless, to improve not only the accuracy but also the robustness of ranking function, we need to take into account a combination of these features and other potential features. Hence, the idea is to build a L2R model, which incorporates various features to generate an optimal ranking function.

Among different L2R methods, Ranking SVM (SVM-Rank) [5], a state-of-the-art pairwise ranking method and also a strong method for IR [4,17], was used. Given n training queries $\{q_i\}_{i=1}^{n}$, their associated document pairs $(x_u^{(i)}, x_v^{(i)})$ and the corresponding ground truth label $y_{(u,v)}^{(i)}$, SVM Rank optimizes the objective function shown in Equation (1) subject to constraints (2), and (3):

$$min\frac{1}{2}\|w\|^2 + \lambda\sum_{i=1}^{n}\sum_{u,v:y_{u,v}^{(i)}}\xi_{u,v}^{(i)} \tag{1}$$

$$s.t. \quad w^T(x_u^{(i)} - x_v^{(i)}) >= 1 - \xi_{u,v}^{(i)} \quad \text{if} \quad y_{u,v}^{(i)} = 1 \tag{2}$$

$$\xi_{u,v}^{(i)} \geq 0, \quad i = 1, ..., n \tag{3}$$

where: $f(x) = w^T x$ is a linear scoring function, (x_u, x_v) is a pairwise and $\xi_{u,v}^{(i)}$ is the loss. The document pairwise in our model is a pair of a query and an article.

[5] http://www.nltk.org/
[6] http://nlp.stanford.edu/software/tagger.shtml

Based on the corpus constructed from all of the single-paragraph articles (see Section 2.2), three basic models were built: TF-IDF, LSI and Latent Dirichlet Allocation (LDA) [9]. Note that, LSI and LDA model transform articles and queries from their TF-IDF-weighted space into a latent space of a lower dimension. For COLIEE 2016 corpora, the dimension of both LSI and LDA is 300 instead of over 2100 of TF-IDF model. Those features were extracted by using `gensim` library [10]. Additionally, to capture the similarity between a query and an article, we investigated other potential features described in Table 2. Normally, the Jaccard coefficient measures similarity between two finite sets based on the ratio between the size of the intersection and the size of the union of those sets. However, in this paper, we calculated generalized Jaccard similarity as:

$$J(q, A) = J(X, Y) = \frac{\sum_i min(x_i, y_i)}{\sum_i max(x_i, y_i)} \tag{4}$$

and Jaccard distance as:

$$D(q, A) = 1 - J(q, A) \tag{5}$$

where $X = \{x_1, x_2, .., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$ are two TF-IDF vectors of a query q and an article A respectively.

Table 2. Similarity features for Ranking SVM

Feature	Description
TF-IDF	Term frequency and inverse document frequency
Euclidean	Euclidean distance computed from TF
Manhattan	Manhattan distance computed from TF
Jaccard	Jaccard distance computed from TF-IDF
LSI	Latent Semantic Indexing computed from TF or TF-IDF
LDA	Latent Dirichlet Allocation computed from TF

The observation in Section 2.2 also reveals that one of the important properties of legal documents is the reference or citation among articles. In other words, an article could refer to an article or refer to just a paragraph of another article. In [2], if an article has a reference to other articles, the authors expanded it with words of referential ones. In our experiment, however, we found that this approach makes the system confused to rank articles and leads to worse performance. Because of that, we ignored the reference and only took into account individual articles themselves. The results of splitting and non-splitting are shown in Table 6.

Legal Question Answering

Legal Question Answering is a form of textual entailment problem, which can be viewed as a binary classification task. To capture the relation between a question and an article, a set of features can be used. However, defining these features

is non-trivial and time consuming, especially in legal domain. To deal with this problem, we used Convolution Neural Network (CNN) [11] for the legal QA task.

The idea behind the QA is that we use CNN [3] with additional features. This is because: (i) CNN is capable to capture local relationship between neighboring words, which helps CNN to achieve excellent performance in NLP problems [12,3,13,14] and (ii) we can integrate our knowledge in legal domain in the form of statistical features, e.g. TF-IDF and LSI.

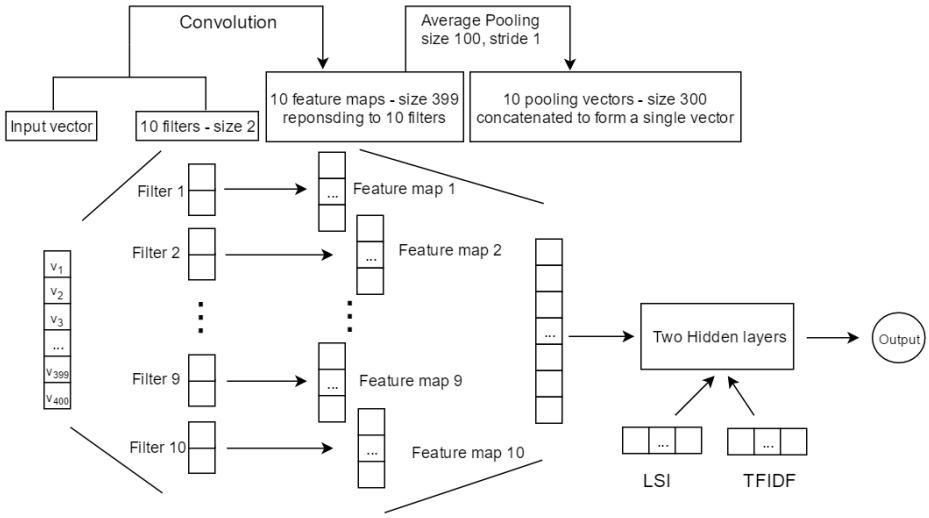


Fig. 2. The illustration of CNN model with additional features: LSI and TF-IDF. Given an input vector, CNN applies 10 filers (length = 2) to generate 10 feature maps of length 399. Afterward, an average pooling filter of length 100 is employed to produce average values from 4 feature maps. Finally, the average values with LSI and TF-IDF are used as input of two hidden neural network layers for QA

In Figure 2, a sentence represented by a set of words was converted to an input feature (a vector v_1^{400}) by using bag-of-words model (BOW) [15]. BOW model generates a vector representation for a sentence by taking a summation over embedding of words in the sentence. The vector is then normalized by the length of the sentence:

$$s = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{6}$$

where: s is a d-dimensional vector of a sentence, s_i is a d-dimensional vector of i^{th} word in the sentence, n is the length of sentence. A word embedding model ($d = 200$) was trained by using Word2Vec[20] on the data of Japanese law corpus[2]. The corpus contains all Civil law articles of Japan's constitution[7] with 13.5 million words from 642 cleaned and tokenized articles.

[7] www.japaneselawtranslation.go.jp

A filter was denoted as a weight vector w with length h; w will have h parameters to be estimated. For each input vector $S \in \mathbb{R}^d$, the feature map vector $O \in \mathbb{R}^{d-h+1}$ of the convolution operator with a filter w was obtained by applying repeatedly w to sub-vectors of S:

$$o_i = w \cdot S[i : i + h - 1] \tag{7}$$

where: $i = 0, 1, 2, ..., d - h + 1$ and $(\cdot)$ is dot product operation.

Each feature map was fed to a pooling layer to generate potential features by using average mechanism [16]. These features were concatenated to a single vector for classification by using Multi-Layer Perceptron with sigmoid activation. During training process, parameters of filters and perceptrons are learned to optimize the objective function.

In our model, 10 convolution filters of length 2 were applied to two adjacent input nodes because these nodes are the same feature type. An average pooling layer of length 100 is then utilized to synthesize important features. To enhance the performance of CNN, two additional statistic features: TF-IDF and LSI were concatenated with the result of the pooling layer, then fed them into a 2-layer Perceptron model to predict the answer.

3 Results and Discussion

3.1 Information Retrieval

Compared Results: For information retrieval task, 20% of query-article pairs are used for testing and the rest of training. As we only consider single-paragraph articles in the training phase, if a multiple-paragraph article is relevant, all of its generated single-paragraph articles will be marked as relevant. In addition, the label for each query-article pair is set either 1 (relevant) or 0 (irrelevant). In our experiment, instead of selecting top k retrieved articles as relevant articles, we consider a retrieved article A_i as a relevant article if its score S_i satisfies Equation (8):

$$\frac{S_i}{S_0} \geq 0.85 \tag{8}$$

where: S_0 is the highest relevant score. In other words, the score ratio of a relevant article and the most relevant article should not be lower than 85% (choosing the value 0.85 for this threshold is simply heuristic based). This is to prevent a relevant article to have a very low score as opposed to the most relevant article.

We ran SVM-Rank with different combinations of features listed in Table 2, but due to limited space, we only report the result of those combinations which achieved highest F1-score. We compared our method to two baseline models TF-IDF and LSI which only use cosine similarity to retrieve the relevant articles. Results from Table 3 indicate that *(LSI, Manhattan, Jaccard)* is the triple of features which achieves the best result and the most stability.

The comparison of our model and other participants' in COLIEE 2015 with the same setting is demonstrated in Table 4. Our model outperforms the two

Table 3. F1-score with different feature groups with the best parameters of SVM-Rank

Method	Features	F1-score
Cosine similarity	TF-IDF	0.5217
	LSI	0.4456
SVM-Rank	TF-IDF, Manhattan, Jaccard	0.582 ± 0.018
	TF-IDF, Euclidean, Jaccard	0.587 ± 0.011
	LDA, Manhattan, Jaccard	0.598 ± 0.011
	LSI, Manhattan, Jaccard	**0.603 ± 0.005**

Table 4. Compare to other models in COLIEE 2015.

Method	F1-score
TF-IDF with lemmarization (baseline)	0.4862
N-gram model [2]	0.5081
Ranking SVM [3]	0.5525
Our model	**0.5746**

best models for LIR, and is significantly higher than the baseline result (0.5746 vs. 0.4862).

Feature Contribution: The contribution of each feature was investigated by using leave-one-out test. Table 5 shows that when all six features are utilized, the F1-score is approximately 0.55. However when excluding Jaccard, F1-score drops to around 0.5. In contrast, when other features are excluded individually from the feature set, the result remains stable or goes up slightly. From this result, we conclude that Jaccard feature significantly contributes to SVM-Rank performance.

Table 5. F1 score when excluding some features from all feature set

Feature	F1-score	Feature Groups	F1-score
All	0.5543	All	0.5543
All \{TF-IDF}	0.5761	All \{TF-IDF, Manhattan, Jaccard}	0.3220
All \{Euclidean}	0.5638	All \{TF-IDF, Euclidean, Jaccard}	0.4891
All \{Manhattan}	0.5543	All \{LDA, Manhattan, Jaccard}	0.4207
All \{Jaccard}	**0.5069**	All \{LSI, Manhattan, Jaccard}	0.4506
All \{LDA}	0.5652	—	—
All \{LSI}	0.5652	—	—

We also analyzed the contribution of feature groups to the performance of SVM-Rank. When removing different triples of features from the feature set, it can be seen that *(TF-IDF, Manhattan, Jaccard)* combination witnesses the highest loss. Nevertheless, as shown in Table 3, the result of *(LSI, Manhattan, Jaccard)* combination is more stable and better.

Splitting vs. Non-Splitting: As mentioned, we proposed to split a multiple-paragraph article into several single-paragraph articles. Table 6 shows that after

splitting, the F1-score performance increases by 0.05 and 0.04 with references and without references respectively. In both cases (with and without the reference), using single-paragraph articles always results in a higher performance.

Table 6. IR results with various methods in COLIEE 2016.

Method	F1-score
Non-splitting + With references	0.5326
Non-splitting + No references	0.5652
With splitting + With references	0.5870
With splitting + No references	**0.6087**

Results from Table 6 also indicate that expanding the reference of an article negatively affects the performance of our model, reducing the F1-score by more than 0.02. This is because if we only expand the content of an article with the content of referential one, it is more likely to be noisy and distorted, leading to lower performance. Therefore, we conclude that a simple expansion of articles via their references does not always positively contribute to the performance of the model.

Tuning Hyperparmeter: Since *linear kernel* was used to train the SVM-Rank model, the role of trade-off training parameter was analyzed by tuning C value from 100 to 2000 with step size 100. More precisely, SVM-Rank model was trained on the data of COLIEE 2015 and COLIEE 2016. Results in Figure 3 indicate that F1-score in COLIEE 2015 dataset is stable for $C \leq 1000$ before fluctuating. For COLIEE 2016 dataset, F1-score peaks at 0.6087 with $C = 600$. We, therefore, use this value for training the L2R model.

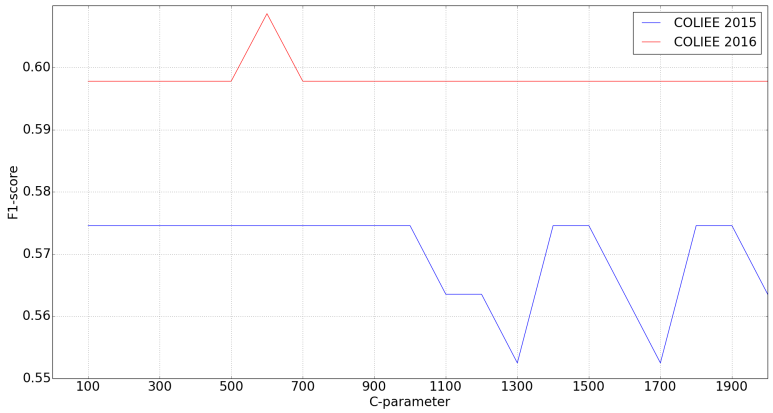


Fig. 3. F1-score with various C-parameters

3.2 Legal Question Answering

Compared Results: In Legal QA task, the proposed model was compared to the original CNN model and separate TF-IDF, LSI features. For evaluation, we took out 10% samples from training set for validation, and carried out experiments on dataset with balanced label distribution for training set, validation set and testing set.

Table 7. Phase 2 results with different models

Method	Accuracy%
Convolutional neural network	51.5
Convolutional neural network with TF-IDF	53.0
Convolutional neural network with LSI	54.5
Our model	**57.6**

In CNN models, we found that these models are sensitive to the initial value of parameters. Different values lead to large difference in results ($\pm$ 5%). Therefore, each model was run n times (n=10) and we chose the best-optimized parameters against the validation set. Table 7 shows that CNN performs better when it is provided with additional features. Also, CNN with LSI produces a better result as opposed to CNN with TF-IDF. We suspect that this is because TF-IDF vector is large but quite sparse (most values are zero), therefore it increases the number of parameters in CNN and consequently makes the model to be overfitted easily.

Tuning Hyperparmeter: To achieve the best configuration of CNN architecture, the original CNN model was run with different settings of number filter and hidden layer dimension. According to Table 8, the change of hyperparameter does not significantly affect to the performance of CNN. We, therefore, chose the configuration with the best performance and least number of parameters: 10 filters and 200 hidden layer size.

Table 8. Results of the original CNN with different settings

Hidden Layer Size	Filter 5	Filter 10	Filter 15
100	51.00	51.00	51.00
150	51.00	51.00	51.00
200	51.00	51.51	51.51
250	51.00	51.51	51.51

3.3 Legal Question Answering System

In this stage, we illustrate our framework on COLIEE 2015 data. The framework was trained on XML files, from H18 to H23 and tested on XML file H24. Given a legal question, the framework first retrieves top five relevant articles and then transfers the question and relevant articles to CNN classifier. The running of framework was evaluated with 3 scenarios:

– *No voting*: taking only a top relevant article to use for predicting an answer for that question.
– *Voting without ratio*: each of results, which is generated by applying our Textual entailment model to each article, gives one vote to the answer which it belongs to. The final result is the answer with more votes.
– *Voting with ratio*: similar to *Voting without ratio*. However, each of results gives one vote corresponding to article's relevant score. The final result is the answer with higher voting score.

Table 9. Task 3 results with various scenarios:

Scenario	Accuracy%
No voting	45.6
Voting without ratio	**49.4**
Voting with ratio	48.1

Table 9 shows results with different scenarios. The result of *No voting* approach is influenced by IR task's performance, so the accuracy is not as high as using voting. The relevant score disparity between the first and second relevant article is large, which causes a worse result of *Voting with ratio* compared to *Voting without ratio*.

4 Conclusion

This work investigates Ranking SVM model and CNN for building a legal question answering system for Japan Civil Code. Experimental results show that feature selection affects significantly to the performance of SVM-Rank, in which a set of features consisting of *(LSI, Manhattan, Jaccard)* gives promising results for information retrieval task. For question answering task, the CNN model is sensitive to initial values of parameters and exerts higher accuracy when adding auxiliary features.

In our current work, we have not yet fully explored the characteristics of legal texts in order to utilize these features for building legal QA system. Properties such as references between articles or structured relations in legal sentences should be investigated more deeply. In addition, there should be more evaluation of SVM-Rank and other L2R methods to observe how they perform on this legal data using the same feature set. These are left as our future work.

Acknowledgement

This work was supported by JSPS KAKENHI Grant number 15K16048, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST.

References

1. Maxwell, K. Tamsin, and Burkhard Schafer. "Concept and Context in Legal Information Retrieval." JURIX. 2008.
2. Danilo S. Carvalho, Minh-Tien Nguyen, Chien-Xuan Tran, Minh-Le Nguyen. "Lexical-Morphological Modeling for Legal Text Analysis". Ninth International Workshop on Juris-informatics (JURISIN), 2015
3. Mi-Young Kim, Ying Xu, Randy Goebel: "A Convolutional Neural Network in Legal Question Answering". Ninth International Workshop on Juris-informatics (JURISIN), 2015
4. Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, Hang Li. "Learning to rank: from pairwise approach to listwise approach." Proceedings of the 24th international conference on Machine learning. ACM, 2007.
5. Joachims, T. "Training Linear SVMs in Linear Time Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)." (2006).
6. Sushmita, A. Kanapala, and S. Pal. "Legal Information Retrieval Task: Participation from ISM, Dhanbad". Ninth International Workshop on Juris-informatics (JURISIN), 2015.
7. Vu Duc Tran, Anh Viet Phan, Long Hai Trieu, and Minh Le Nguyen. "An Approach for Retrieving Legal Texts". Ninth International Workshop on Juris-informatics (JURISIN), 2015.
8. Yoshinobu Kano. "Keyword and Snippet Based Yes/No Question Answering System for COLIEE 2015". Ninth International Workshop on Juris-informatics (JURISIN), 2015.
9. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.
10. Radim Řehůřek and Petr Sojka. "Software framework for topic modelling with large corpora". Proc. LREC Workshop on New Challenges for NLP Frameworks, 2010.
11. Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).
12. Yih, Wen-tau, Xiaodong He, and Christopher Meek. "Semantic Parsing for Single-Relation Question Answering." ACL (2). 2014.
13. Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng, and Grgoire Mesnil. "Learning semantic representations using convolutional neural networks for web search." Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014.
14. Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." arXiv preprint arXiv:1404.2188 (2014).
15. Yu, Lei, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. "Deep learning for answer sentence selection." arXiv preprint arXiv:1412.1632 (2014).
16. Boureau, Y-Lan, Jean Ponce, and Yann LeCun. "A theoretical analysis of feature pooling in visual recognition." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.
17. Minh-Tien Nguyen, Viet-Anh Phan, Truong-Son Nguyen, Minh-Le Nguyen. "Learning to Rank Questions for Community Question Answering with Ranking SVM." arXiv preprint arXiv:1608.04185 (2016).
18. Minh-Tien Nguyen, Quang-Thuy Ha, Thi-Dung Nguyen, Tri-Thanh Nguyen, Le-Minh Nguyen. "Recognizing textual entailment in vietnamese text: an experimental study." Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on. IEEE, 2015.

19. Freund, Yoav, and Robert E. Schapire. “A desicion-theoretic generalization of on-line learning and an application to boosting.” European conference on computational learning theory. Springer Berlin Heidelberg, 1995.
20. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. “Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).

Legal Yes/No Question Answering System using Case-Role Analysis

Ryosuke Taniguchi[1], Yoshinobu Kano[1,*]

[1] Faculty of Informatics, Shizuoka University, Japan
rtaniguchi@kanolab.net kano@inf.shizuoka.ac.jp

Abstract. A central issue of yes/no question answering is usage of knowledge source given a question. While yes/no question answering has been studied for a long time, legal yes/no question answering largely differs from other domains. The most different characteristics is that legal issues require roles and relationships of agents in sentences to be precisely analyzed. We developed a yes/no question answering system for legal domain. Our system uses case-role analysis, in order to find correspondences of roles and relationships between given problem sentences and knowledge source sentences. We applied our system to the JURISIN's COLIEE (Competition on Legal Information Extraction/Entailment) task. Our system performance was better than systems of previous task participants in Phase Two. This result shows the importance of the points described above, while we still have a couple of issues for our system accuracy to be improved as future work.

Keywords: COLIEE, Question Answering, Legal Bar Exam, Legal Information Extraction.

1 Introduction

Automatic question answering is attracting more interests recently. Due to the increasing expectation to the Artificial Intelligence (AI) technologies, people tend to regard question answering systems as a brand new technology emerged today. However, most successful systems employ rather traditional techniques of question answering which have decades of history [1] [2] [3] [4] [5] [6] [7], including series of shared tasks such as TREC [8], NTCIR [9] and CLEF [10]. This paper describes our challenge to the COLIEE 2016 legal bar exam, which asks participants to answer true or not based on the Civil Law Articles, given text drawn from the Japanese legal bar exam.

A variety of algorithms and systems have been proposed for question answering. Typically, these question answering systems used *big data* for answering questions [11] [12] [13] [14]. For example, Dumais et al. [15] focused on the redundancy available in large corpora as an important resource. They used this redundancy to simplify their algorithm and to support answer mining from returned snippets. Their system performed quite well given the simplicity of the techniques being utilized.

The now widely known IBM Watson system [16] would be considered as a typical example of such a question answering system of the big data approach. The IBM Watson system won in the Jeoperdy! Quiz TV program competing with human quiz winners. The core Watson system employed a couple of open source libraries, including the traditionally well-designed DeepQA system [17] as its skeleton of question answering processing. Because their target domain, the Jeoperdy! Quiz, could ask broad range of questions, they collected a huge amount of knowledge sources from the Internet, etc., extracting relevant knowledge by combining a couple of different natural language processing (NLP) techniques.

Answering university examinations is another example. The Todai Robot project [18] is a challenge to solve Japanese university examinations, focusing towards attaining a high score in the National Center Test for University Admissions by 2016, and passing the entrance exam of the University of Tokyo (Todai) in 2021 [19]. Although the Todai Robot project tries to achieve higher scores, their aim is rather to reveal the current performance and limitation of the existing AI technologies, using the examinations as its benchmark, similar to the COLIEE's legal bar exam task. In contrast to the COLIEE task, the challenge of Todai Robot project includes variety of subjects including Mathematics, English, Japanese, Physics, History, etc. all written in Japanese language. While solving any problem of these subjects could be considered as question answering, some problems require special technologies. For example, Mathematics and Physics require to process formula; Japanese requires to infer emotions of story characters. Solving the History subjects might be considered as rather an extension of the existing question answering issues. The Todai Robot project achieved better scores than the average of the real human applicants in their Mock Exam challenges.

Recognition of textual entailments (RTE or RITE) is another related issue. RTE has been intensively studied for recent days, including shared tasks such as RTE tasks of PASCAL [20][21], SemEval-2012 Cross-lingual Textual Entailment (CLTE) [22], NTCIR RITE tasks [23][24][25], etc. In the third PASCAL RTE-3 task, contradiction relations are included in addition to entailment relations [21]. In the RTE-6 task, given a corpus and a set of candidate sentences retrieved by a search engine from that corpus, systems are required to identify all the sentences from among the candidate sentences that entail a given hypothesis. NTCIR-9 RITE, NTCIR-10 RITE2, and NTCIR-11 RITEVal Exam Search tasks [25] required participants to find an evidence in source documents and to answer a given proposition by yes or no. Research of RTE normally tries to employ logical processing.

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

Another issue is whether the knowledge source needs to be "big data" or not. Regarding the COLIEE's legal problems, required knowledge source can be limited. In this paper, we suggest to use small data in a precise way, rather than to use enormous amount of data as knowledge source.

Based on these thoughts, we built our yes/no question answering system. Our system does not employ any machine learning. The main method of our system is a case-role

based analysis, coupling with end-of-sentence expressions which has two levels of abstraction. We tried a couple of different combinations of our methods, optimized to Phase Two, while also tried Phase Three. Our system achieved more than 7 points better score than the best participant's system in the previous COLIEE 2015. There are still many difficult issues remained to be solved though.

We describe datasets of NTCIR RITE challenge, and datasets of previous and this COLIEE tasks in Section 2. Section 3 describes our design of the yes/no question answering system. Section 4 shows our experimental results for this COLIEE task and the RITE task. We discuss our achievements and limitations in Section 5, comparing the different tasks together with other COLIEE participants' systems, mentioning possible future works in Section 6. We conclude our paper in Section 7.

2 Related Tasks and Datasets

2.1 Exam Search Subtask in NTCIR RITEVal

While there were a couple of subtasks in the NTCIR RITE series, we describe the exam search subtask of NTCIR-11 RITEVal because the COLIEE dataset adopted the same format as the RITEVal dataset. RITEVal is an evaluation-based workshop held in 2013, aiming to recognize entailment, paraphrase, and contradiction between sentences, which is a common problem shared widely among researchers of NLP and information access [26] [21].

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

RITEVal's dataset was developed from the past Japanese National Center Test questions for University Admissions (Center Test). The Center Test asks students multiple-choice style questions. The RITEVal dataset consists of three types of questions, "select the correct choice" type, "select the wrong choice" type, and "combination" type.

In the RITEVal task, the original multiple-choices were not given as a whole, but given one by one. In "select the correct choice" type questions, given a choice, RITEVal participant systems are asked to return a confidence value for that choice. Evaluation is performed by comparing confidence values for each original multiple-choices, regarding the largest value as the participant system's answer (smallest in case of "select

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

Fig. 1. An example of COLIEE legal bar problem which asks to answer t1 entails t2 or not. The correct answer is "yes" in this example.

wrong choice" type questions). In the "combination" type questions, the system is required to label Y or N for each choice and evaluated by a combination of these Y/N w.r.t the original multiple-choice question. In this paper, we focus on the "select correct/wrong choice" type questions.

Fig. 1 is an example of the COLIEE dataset but can also be regarded as an example set of choices in the RITEVal dataset. In this example, one of the four choices is the correct one.

2.2 JURISIN COLIEE datasets

The COLIEE shared task series is held in association with the JURISIN (Juris-informatics) workshop. The first one was the COLIEE 2014 shared task [27], and the second one was the COLIEE 2015 shared task [28]. This paper mainly describes our participation to the COLIEE 2016 shared task [28]. We call COLIEE 2016 simply as COLIEE in this paper.

The COLIEE shared task consists of three phases.

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

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

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

The corpus of legal questions is drawn from Japanese Legal Bar exams, and the relevant Japanese Civil Law articles were also provided. While there was an English translation version of the dataset provided, we only used the original Japanese version. Fig. 1 shows an example of the COLIEE dataset.

3 Method

3.1 Design Concepts

Wikipedia is a typical web sourced big data. However, we decided to use only the Civil Law data, or small data, in our system. A reason is that structures of Civil Law articles are clean. That is, the Civil Law articles use only one place (snippet) for one topic. Another reason is that we need precise analyses to solve the legal issues, rather than statistically calculate rough estimate values in a surficial way. We took an unsupervised approach for the same reasons.

We assume that roles and relationships in sentences play critical role in processing legal documents. In other words, the most important element should be the case-roles and its predicates of sentences. We focused on end-of-sentence expressions which normally correspond to the predicates.

Our yes/no question answering system is based on case-role analyses. We use JUMAN [29] and KNP [30] to obtain case-role analyses. JUMAN is a Japanese morphological analyzer where we added a custom dictionary for legal technical terms based on a Japanese legal term dictionary (“有斐閣法律用語辞典第4版”). KNP is a Japanese dependency case structure analyzer, works on top of JUMAN.

Using results of these tools, we obtain subjective cases and an end-of-sentence expression for each sentence. Unfortunately, there is no tool that can estimate deep case roles, surficial case roles are only available. A subjective case is normally specified by particles “が” or “は” in Japanese. We regard these cases as subjective cases.

When we analyze the Civil Law articles, we removed each header part “X条 (Article X)”, which includes an article name and numbers. When “ただし (only provided)” appears within a sentence, we may discard the entire sentence depending on methods we employ. This is because this word “ただし” marks conditional phrase which is additional rather than dominate in the entire sentence meaning.

3.2 Paring Subjective Case and End-of-Sentence Expressions (Method 1)

Fig. 2 shows a conceptual figure of Method 1with an example. Method 1 takes relations into account with subject cases and end-of-sentence expressions, which are extracted as described before. We describe details of Method 1 below.

First, we make a one-to-one pair of a subject case and a sentence end using results of the morphological analyzer and the case structure analyzer for each sentence. If a target phrase has no subject case extracted, we only extract end-of-sentence expressions.

Second, we simplify the extracted information to obtain better abstraction which helps to decide Yes/No. We remove superficial case particle of “が” and “は” from subjective case information. We also remove punctuation marks. Additionally, if the end-of-sentence expressions contain possibility, we replace them with *true* or *false* depending on the expression has a negation or not. If the end-of-sentence expressions does not contain possibility, we leave this end-of-sentence expression as it is.

In Phase Two, a problem paragraph (t2) and a knowledge source (Civil Law article) paragraph (t1) are given. We apply the process describe above both for t1 and t2 sentences. Then we compare results of the process for all sentence pairs between t1 and t2. When any pair of a t1 sentence and a t2 sentence has the same results, i.e. the same subject noun and end-of-sentence expression, our system answers Yes. If none of these conditions is satisfied, we determine Yes/No by matching only the end-of-sentence expression pair.

3.3 One-to-Many Subjective Case and Sentence End (Method 2)

Method 2 judges Yes/No more strictly than Method 1. Although Method 2 is basically same as Method 1, Method 2 infers possible end-of-sentence expressions given a subjective case. Therefore, there could be one or more end-of-sentences for each subjective case in Method 2.

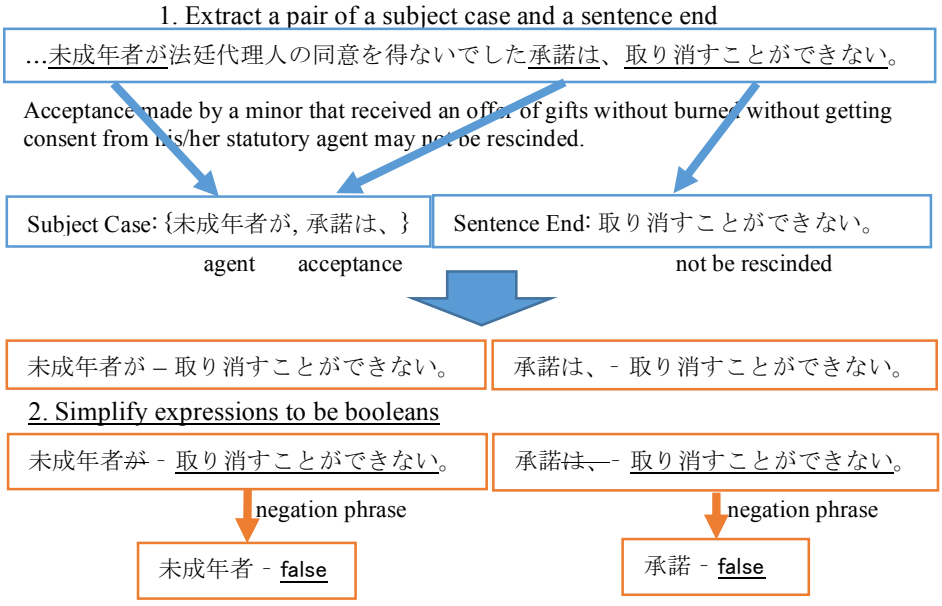


Fig. 2. A conceptual figure of Method 1.

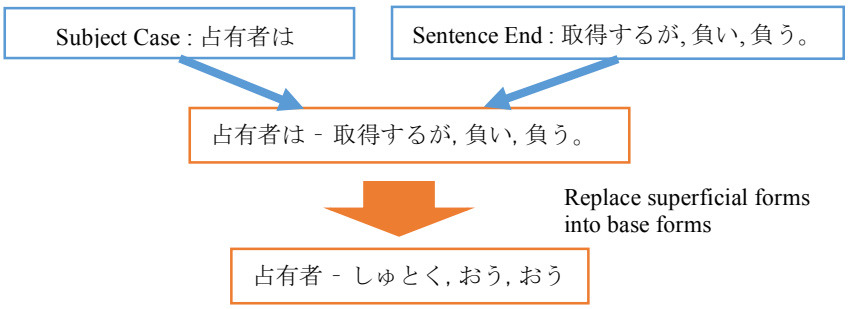


Fig. 3. An example of Method 2 process.

For example, from "善意の占有者は，占有物から生ずる果実を取得するが，強迫によって占有をしている者は，果実を返還する義務を負い，果実を既に消費している場合には果実の代価を償還する義務を負う。 (A possessor in good faith shall acquire fruits derived from Thing in his/her possession, but a person in possession Thing through duress shall be obliged to return fruits, and if the fruits have already been consumed, he/she shall be obliged to reimburse the cost of the fruits.)", we can obtain "占有者 (a possessor)" as a subjective case and "取得するが (acquire)", "負い (be obliged)" and "負う。(be obliged)" as end-of-sentence expressions.

In addition, this method uses base forms of the end-of-sentence expressions rather than the original forms. Fig. 3 illustrates this Method 2.

Method 1 is "looser" and Method 2 is "stricter" in terms of the Yes/No judgement criteria. Therefore, in addition to Method 1/Method 2 alone, we combined Method 1 and Method 2 by using Method 1 as a base method, and filter its results by Method 2. This Method 2 filter overwrites any Method 1 result when Method 2 outputs Yes. This combination method determines whether all of t1 results are included in t2 results or not.

4 Experiments

Experiments were conducted on the COLIEE 2016 Japanese subtask dataset. We did not use the Training/Development set because our method does not use any machine learning method i.e. unsupervised, while we used these data sets to check and improve results of our system.

Our system focused on Phase Two. Phase Three was answered using results of the Phase Two system.

All of the results use training set, because the evaluation of formal run results are not provided yet from the COLIEE task organizers before this paper submission.

4.1 Phase Two of COLIEE 2016

In Phase Two, we answered Yes or No determined by four rules as described below.

Rule1 : Using Method1 and Method2 with “ただし”
Rule2 : Using Method1 and Method2 without “ただし”
Rule3 : Using Method1 with “ただし”
Rule4 : Using Method1 without “ただし”

Fig. 3. Summary of the rules employed in our methods.
”ただし” means “only provided”.

Rule 1 uses Method 1 and Method 2. Rule 1 analyzes all sentences. Rule 2 is same as Rule 1 but excludes sentences from analyses when include a word “ただし (only provided)”. Rule 3 and Rule 4 only use Method 1, their difference is same as Rule 1 and 2. Fig. 3 summarizes these rules.

Table 1 shows results of our methods in the Phase Two training set. The H25 dataset corresponds to the previous COLIEE 2015 test dataset. Our result shows much better score (74.24) than the best participant’s system in COLIEE 2015 (66.67).

Table 1. Phase Two Results of COLIEE 2016 training dataset.
Cells with color are the best score in corresponding year’s data.

Dataset	H21	H22	H23	H24	H25
Rule1_Accuracy	66.07	63.83	65.85	67.09	72.73
Rule1_F-measure	66.06	63.76	65.67	66.44	72.32
Rule2_Accuracy	60.71	61.70	65.85	67.09	74.24
Rule2_F-measure	60.71	61.55	65.67	66.44	74.09
Rule3_Accuracy	64.29	68.09	60.98	73.42	71.21
Rule3_F-measure	64.10	68.03	60.95	73.55	70.67
Rule4_Accuracy	60.71	65.96	60.98	70.89	72.73
Rule4_F-measure	60.66	65.94	60.95	70.72	72.50

4.2 Phase Three of COLIEE 2016

In Phase Three, we used the same methods as Phase Two. We analyzed all Civil Low articles for each question regarding each article as t2 document using our Phase Two system, giving Yes/No answers. Then we counted numbers of Yes answers for each question. Finally, we set a threshold value of the Yes counts to answer Yes/No for Phase Three. We set the threshold to 50 in our experiments.

Table 2 Phase Three results of COLIEE 2016 training data set.

Dataset	H21	H22	H23	H24	H25
Rule1_Accuracy	48.21	48.94	56.10	58.23	43.94
Rule1_F-measure	48.20	45.98	56.07	58.20	43.93
Rule2_Accuracy	48.21	48.94	56.10	58.23	43.94
Rule2_F-measure	48.20	45.98	56.07	58.20	43.93
Rule3_Accuracy	48.21	46.81	58.54	46.84	43.94
Rule3_F-measure	39.74	41.63	53.06	39.80	36.19
Rule4_Accuracy	48.21	46.81	58.54	46.84	43.94
Rule4_F-measure	39.74	41.63	53.06	39.80	36.19

Table 2 shows results in Phase Three. The H25 result scores are quite lower than the ones in COLIEE 2015.

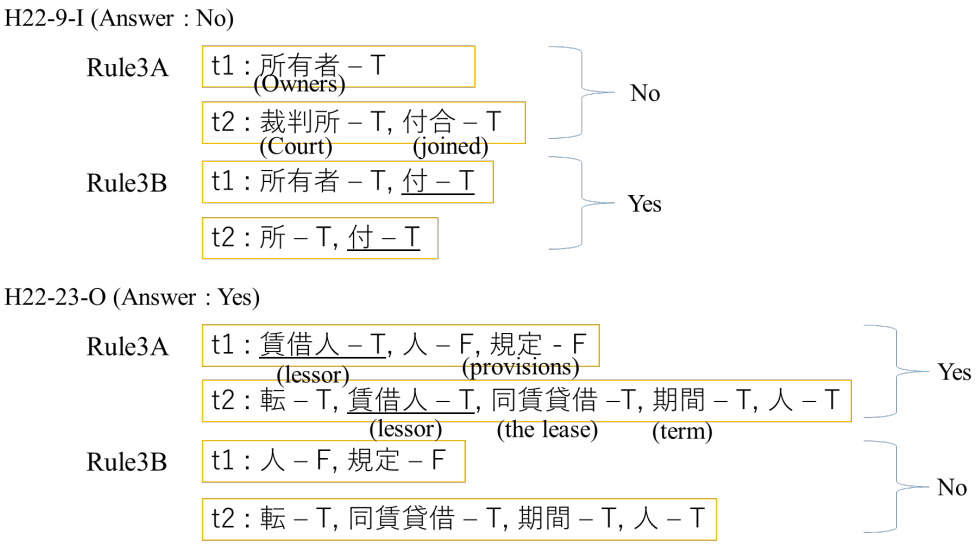
5 Discussion

Although it is still limited to the training data evaluation, our system performed better than other previous systems as far as we compared. In Phase Two, we obtained better results than last year's COLIEE 2015 (H25 data set). Let us discuss our results in detail for a couple of points below.

Firstly, we used a customized legal technical term dictionary for morphological analyses. Table 3 shows results when we did not use the legal dictionary but only used the default one. When comparing Table 1 with Table 3, Table 1 has better results than Table 3 entirely in most cells while sometimes slightly worse.

Table 3 Results of COLIEE2016 training dataset in Phase Two without our custom dictionary. Cells with blue color are worse than corresponding cells in Table 1. Cells with red color are better than corresponding cells in Table 1.

Dataset	H21	H22	H23	H24	H25
Rule1_Accuracy	66.07	63.83	65.85	65.82	74.24
Rule1_F-measure	66.06	63.76	65.67	65.47	73.95
Rule2_Accuracy	60.71	61.70	65.85	67.09	75.76
Rule2_F-measure	60.71	61.55	65.67	66.66	75.67
Rule3_Accuracy	64.29	63.83	58.54	72.15	71.21
Rule3_F-measure	64.10	63.76	58.54	72.11	70.67
Rule4_Accuracy	60.71	61.70	58.54	70.89	72.73
Rule4_F-measure	60.66	61.68	58.54	70.72	72.50



Rule 3A uses a customized legal technical term dictionary, Rule 3B does not use it. "t1" is a result from the Civil Law articles, and "t2" is from the problem setences. T = True, F = False

Fig. 4. Examples of failed process without the custom legal technical term dictionary.

In order to analyze this effect in detail, we focus on results in H22 (H22-9-I and H22-23-O) as shown in Fig. 4. In this figure, Rule 3 with the legal technical term dictionary is referred to as Rule 3A, Rule 3 without the legal technical term dictionary is referred to as Rule 3B. H22-9-1, the morphological analysis failed without the custom dictionary; "付合 (joined)" was divided into "付" and "合", "付" has matched with other fragment of morphemes, leading its yes/no answer to wrong "yes". H22-23-O shows an opposite case. "貸借人 (lessor)" was divided into "貸借" and "人" without the custom dictionary. Because "人 (person)" is a common word, it matches with other morpheme fragments frequently. In this case, this leads its yes/no answer to wrong "no".

As a whole, these results suggest that our legal technical term dictionary was effective in our system.

Secondly, we discuss effect of the word "ただし (only provided)". In Phase Three, no difference is observed between Rule 1 and Rule 2, Rule 3 and Rule 4, as shown in Table 2. Differences of Rule 1 and Rule 2, Rule 3 and Rule 4, are whether we eliminate phrases which include "ただし". On the other hand, we got better results when retaining phrases of "ただし" (Rule1 and Rule3) except for H25 in Phase Two. Because this word "ただし" could lead to the opposite result, there should have a positive effect when excluding phrases of "ただし", while the results imply not. Further consideration will be needed to examine the effect of this word.

Thirdly, results of Phase Two seem not correlate with results of Phase Three, while we used methods and results of Phase Two directly to Phase Three. This might be because our methods were optimized for Phase Two, not for Phase Three. Deeper analyses of threshold and parameter effects in Phase Three would be useful.

Fourthly, our system depends on the accuracy of the case structure analyzer, KNP. Because this analyzer was mainly trained by newswire texts, its accuracy in legal texts would not be sufficient. Analysis of errors in the case structure analyzer would be required.

Finally, we observe a range of different scores among years from H21 to H25. There could be different tendency among the datasets of these years. Precise analysis of each sentence would be needed to find causes of this score variation.

6 Future Work

A large difference with other past COLIEE systems is that we used Japanese while others used English. While the English dataset is a direct translation of the original Japanese dataset, this difference could result in different issues when solving the problems. For example, Japanese text requires explicit tokenization process. When this tokenization fails, the final result could also be failed. As far as we observed, our morphological analyses were fine with the legal technical term dictionary, but there were still some failures observed. Comparison with English version will help.

The technical terms in the COLIEE dataset tend to include logical relations implicitly. This sort of logical relation extraction would be still very difficult to perform in sufficiently high accuracy considering the current NLP technologies. Furthermore, there are sometimes "instantiations" in the Civil Law articles like "when A claims…". This sort of instantiations require higher level of abstraction process.

The document structure of the given knowledge source, the Japanese Civil Law articles, is special. The Civil Law articles have references; logic and conditions are described in a single sentence, or scattered across snippets.

The Civil Law articles include specific negation and acronym expressions which are critical in solving the problems. While our system handles these expression to some extent, there may be some expressions missing.

Future work would include solutions to the above issues, in addition to those described in the discussion section. The most difficult issue to solve would be the logic and abstraction.

7 Conclusion

Legal document processing requires different issues to solved than other domains. A large difference in legal yes/no question answering is that legal issues require roles and relationships of agents in sentences to be precisely analyzed. Based on this observation, we developed a yes/no question answering system for legal domain. Our system uses case-role analysis and end-of-sentence expressions, in order to find correspondences of

roles and relationships between given problem sentences and knowledge source sentences. We applied our system to COLIEE 2016 Japanese task. Our system performance was 7 points better than the best system of previous COLIEE task participants in Phase Two. While this result shows the importance of the points we employed in our system, there are still a couple of issues to be resolved as future work

Acknowledgements

This work was partially supported by MEXT Kakenhi and National Institute of Informatics.

References

[1] D. Lin and P. Pantel, "Discovery of Inference Rules for Question-answering," *Nat. Lang. Eng.*, vol. 7, no. 4, pp. 343–360, 2001.

[2] D. Ravichandran and E. Hovy, "Learning Surface Text Patterns for a Question Answering System," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 41–47.

[3] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003, pp. 129–136.

[4] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 235–242.

[5] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua, "Question Answering Passage Retrieval Using Dependency Relations," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, pp. 400–407.

[6] X. Xue, J. Jeon, and W. B. Croft, "Retrieval models for question and answer archives," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 475–482.

[7] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media," in *Proceedings of the 17th International Conference on World Wide Web*, 2008, pp. 467–476.

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

[9] N. Kando, K. Kuriyama, and T. Nozue, "NACSIS Test Collection Workshop (NTCIR-1) (Poster Abstract)," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 299–300.

[10] M. Braschler, "CLEF 2000 - Overview of Results," in *Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation*, 2001, pp. 89–101.

[11] C. C. T. Kwok, O. Etzioni, and D. S. Weld, “Scaling Question Answering to the Web,” in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 150–161.

[12] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Web-scale Information Extraction in Knowitall: (Preliminary Results),” in *Proceedings of the 13th International Conference on World Wide Web*, 2004, pp. 100–110.

[13] J. Jeon, W. B. Croft, and J. H. Lee, “Finding Similar Questions in Large Question and Answer Archives,” in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 2005, pp. 84–90.

[14] H. Kanayama, Y. Miyao, and J. Prager, “Answering Yes/No Questions via Question Inversion,” in *the 24th International Conference on Computational Linguistics (COLING 2012)*, 2012, pp. 1377–1391.

[15] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng, “Web Question Answering: Is More Always Better?,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 291–298.

[16] D. Ferrucci, “Introduction to ‘This is Watson,’” *IBM J. Res. Dev.*, vol. 56, no. 3.4, p. 1:1-1:15, 2012.

[17] D. A. Ferrucci, “IBM’s Watson/DeepQA,” *SIGARCH Comput. Arch. News*, vol. 39, no. 3, 2011.

[18] N. H. Arai, “The impact of AI—can a robot get into the University of Tokyo?,” *Natl. Sci. Rev.*, vol. 2, no. 2, pp. 135–136, Jun. 2015.

[19] “The Todai Robot project.” [Online]. Available: http://21robot.org/.

[20] I. Dagan, O. Glickman, and B. Magnini, “The PASCAL Recognising Textual Entailment Challenge,” in *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2006, pp. 177–190.

[21] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, “The Third PASCAL Recognizing Textual Entailment Challenge,” in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007, pp. 1–9.

[22] M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo, “Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization,” in *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, 2012, pp. 399–407.

[23] H. Shima, H. Kanayama, C. Lee, C. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda, “Overview of NTCIR-9 RITE: Recognizing Inference in TExt,” in *NTCIR-9 Workshop*, 2011, pp. 291–301.

[24] Y. Watanabe, Y. Miyao, J. Mizuno, T. Shibata, H. Kanayama, C.-W. Lee, C.-J. Lin, S. Shi, T. Mitamura, N. Kando, H. Shima, and K. Takeda, “Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10,” in *the NTCIR-10 Workshop*, 2013, pp. 385–404.

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

[26] 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.

[27] “Competition on Legal Information Extraction/Entailment (COLIEE-14), Workshop on Juris-informatics (JURISIN) 2014,” 2014. [Online]. Available: http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html.

[28] M.-Y. Kim, R. Goebel, and S. Ken, “COLIEE-2015: Evaluation of Legal Question Answering,” in *Ninth International Workshop on Juris-informatics (JURISIN 2015)*, 2015.

[29] “JUMAN (a User-Extensible Morphological Analyzer for Japanese).” [Online]. Available: http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN.

[30] “Japanese Dependency and Case Structure Analyzer KNP.” [Online]. Available: http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP.

An Ensemble Based Legal Information Retrieval and Entailment System

Kiyoun Kim, Seongwan Heo, Sungchul Jung, Kihyun Hong, and Young-Yik Rhim

Intellicon Meta Lab,
5th floor, 554, Nonhyeon-ro, Gangnam-gu, Seoul, Korea
{kykim,swheo,jsc,kihyun.hong,ceo}@intellicon.co.kr

Abstract. Legal information retrieval (IR) and recognizing textual entailment (RTE) become active research fields in legal text mining and play a crucial role in developing legal question and answering (QA) systems. To tackle the legal IR and RTE tasks, in this paper, we present measurement and abstract fusion based ensemble methods. To develop legal IR systems, we propose measurement based ensemble methods, which combine several individual lexical, semantic, and syntactic feature based similarity measures. To build legal RTE models, we present measurement and abstract level ensemble methods, which fuse the decision probabilities and decision results of classifiers, respectively. The experimental results show that the ensemble methods outperform the single algorithms in both retrieval and entailment problems. The presented works are applied to the competition on legal information extraction/entailment (COLIEE) 2016, which is held to solve legal IR and RTE problems.

Keywords: Legal text mining · Natural language processing · Information retrieval · Recognizing textual entailment · Machine learning · Deep learning

1 Introduction

Legal IR become one of the active research areas, since large collections of digitized legal information are available and the demands for the legal domain applications are increased.

As legal domain information composes mostly of textual data, legal IR applications adopt natural language processing (NLP) and understanding (NLU) technologies [1]. Moreover, legal IR applications are required to provide more accurate retrieval results than general NLP and NLU applications.

As measuring text similarity and RTE are core elemental processes in NLU applications, searching relevant legal articles in a legal data collection and determining the relationships between the articles are indispensable tasks in legal IR applications. However, it is more difficult and complex to apply standard IR and RTE methods to legal domain than other fields. One reason is that the terminology mainly consists of legal jargons, which are rarely used in other domains [2].

COLIEE 2016 [3] aims to solve these tasks. It presents two aspects of tasks related to the legal information processing. The first task is to retrieve Japanese civil code articles that are related to a query, which is extracted from Japanese Bar exam. The second task is to determine entailment relationship between a query and relevant articles. In the first task, it is needed to formulate an IR system that extracts the subset of relevant articles from entire civil code articles. In the second task, it is requested to clarify the entailment relationship, which determines whether the meaning of a query is inferred from given relevant articles; it is to construct a "Yes" or "No" answering system about the entailment relationship between a query and relevant articles.

For the competition tasks, we use ensemble methods to combine several single NLU algorithms. In designing ensemble systems, it is essential to assign proper weights in order to improve the performance quality. For developing the legal IR system, we utilize the least square method (LSM) and linear discriminant analysis (LDA) to compute appropriate ensemble weights. In relevant article retrieval, several similarity scores are combined with the assigned weights. For the RTE task, we apply majority vote and LSM for building the ensemble classification model.

Section 2 briefly describes the textual similarity measures and then explains the details of ensemble methods to fuse the similarity measures. Section 3 states the strategies for solving RTE, including a brief introduction to single classifiers and ensemble methods. Section 4 outlines the experiment setting and shows experimental results of the presented methods. Finally, Section 5 concludes this paper.

2 Relevant Article Extraction

Relevant article extraction is one of the most important tasks for legal IR, because it can be applied to informative legal article extraction among unconstrained legal articles. In this section, we explain the way to find the relevant articles to a given query. First, the textual similarity measures, which are utilized in our ensemble systems, are introduced. Then, the proposed ensemble methods are explained.

2.1 Textual Similarity Measure

Textual similarity indicates how the contents of articles are similar to those of a given query. We utilize several similarity measures such as lexical, syntactic and semantic similarity measures between a query Q and an article A to build an ensemble similarity score.

2.1.1 Lexical Similarity

To obtain lexical similarities, we utilize several measures.

Word overlap similarity [4] is measured by the size of the word overlap between Q and A. Let W_Q and W_A be the sets of words in Q and A, respectively. These are preprocessed word sets by tokenization, lemmatization, and stop word removal. The *Word overlap similarity* is calculated as follows:

$$Word\ overlap\ similarity(Q, A) = \frac{|W_Q \cap W_A|}{|W_Q \cup W_A|}. \quad (1)$$

It has a value ranging from 0 to 1.

Cosine similarity [4] is calculated by the degree between the vectors of Q and A, which are represented by *term frequency* (TF) in a non-redundant set of words.

$$Cosine\ Similarity(Q, A) = \frac{q \cdot a}{\|q\|\|a\|} = \frac{\sum_{i=1}^{n} q_i \times a_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \times \sqrt{\sum_{i=1}^{n} a_i^2}}, \quad (2)$$

where n is the size of vocabulary, q and a are the n-dimensional vectors of Q and A, and q_i and a_i is the i-th element of q and a, respectively. It has a value ranging from -1 to 1, which indicates the inner product of two vectors in a n-dimensional vector space.

Substring similarity [4] is computed by selecting the maximum value among the combinations of each sub sentence level cosine similarities.

Longest words sequence [5] is measured by the maximum length of the word sequence that appears in both Q and A.

We also use *Lucene similarity score* [6], which is a modified TF-IDF based similarity measure.

2.1.2 Syntactic Similarity

Role similarity [4] is computed by averaging the sentence part similarities in each sub sentence. In role similarity computation, we use three sentence parts such as subject, verb and object. These are extracted by the Stanford parser [7]. To compute the sentence part similarity, we utilize three types of role similarities, Wu-Palmer similarity [8], path similarity and Lin similarity [9], which use *WordNet* [10].

2.1.3 Semantic Similarity

Latent semantic indexing (LSI) [11] is an indexing and retrieving method that uses singular value decomposition (SVD) to identify patterns of the relationships between the terms and concepts in an unstructured collection of text. LSI is based on the principle that words used in the same contexts tend to have similar meanings.

WordNet similarity also shows semantic relationship. In this case, the part-of-speech (POS) similarity between two documents is computed by using the *WordNet*. It is different from the role similarity, which uses sentence parts. Wu-Palmer, path and Lin similarities are also used to compute the *WordNet* similarities. The following equations describe the noun and verb similarities:

$$WordNet\ noun\ similarity\ = \frac{similarity(noun\ in\ Q, noun\ in\ A)}{number\ of\ similarities\ calculated}. \quad (3)$$

$$WordNet\ verb\ similarity\ = \frac{similarity(verb\ in\ Q, verb\ in\ A)}{number\ of\ similarities\ calculated}. \quad (4)$$

2.2 Ensemble Similarity Measure

In this subsection, we present ensemble methods. As shown in Section 4, ensemble methods show the more improved performance in the retrieval task than using a single similarity measurement. We utilize two types of ensemble methods, which are LSM and LDA [12].

2.2.1 Least Square Method

LSM is a standard approach in regression analysis to obtain approximate solution of overdetermined systems. LSM provides the overall solution, which minimizes the sum of squares of the errors:

$$\begin{bmatrix} S_1(q_1, a_{r1}) & \cdots & S_L(q_1, a_{r1}) \\ \vdots & & \vdots \\ S_1(q_1, a_{n1}) & \cdots & S_L(q_1, a_{n1}) \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_{L-2} \\ \alpha_{L-1} \\ \alpha_L \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{5}$$

$$\hat{\alpha} = (S^T S)^{-1} S^T t, \tag{6}$$

where S_i is the i-th similarity measure between Q and A, α_i is the weight of i-th similarity measure, and L is the total number of similarity measures. a_{ri} is a relevant article of q_i and a_{ni} is a non-relevant article of q_i, which is sampled from entire articles. t is a binary vector, which is assigned by document relevancy. We note that the relevancy between Q and A are given in the training data. Then, the ensemble score is calculated by the linear combination of individual similarity scores:

$$LSM\ ensemble\ score = \sum_{i=1}^{L} \alpha_i S_i. \tag{7}$$

2.2.2 Linear Discriminant Analysis

LDA seeks to find maximally separable directions in the given binary class vectors. It is obtained by solving the following optimization problem.

$$\hat{\alpha} = arg\ max_{\alpha} \frac{\alpha^T S_B \alpha}{\alpha^T S_W \alpha}, \tag{8}$$

where S_B is "between-class scatter" matrix and S_W is "within-class scatter" matrix. LDA computes a transformation that maximizes "between-class scatter" while minimizing "within-class scatter". The solution is computed as follows:

$$\hat{\alpha} = {S_W}^{-1}(\mu_r - \mu_n), \tag{9}$$

where μ_r and μ_n are the mean vectors of relevant article and non-relevant article classes, respectively. Then, LDA ensemble score is calculated as follows:

$$LDA\ ensemble\ score = \sum_{i=1}^{L} \alpha_i S_i. \quad (10)$$

3 Recognizing Textual Entailment

As mentioned in Section 1, RTE plays a crucial role in legal information systems. In this section we explain our entailment decision methods, which utilize ensemble-based classification approaches. Several conventional classifiers and convolutional neural network (CNN) based *Siamese* network classifiers [13] are considered to build ensemble systems. First, we state that the vector representations of inputs of the classifiers, and then briefly summarize the classification methods utilized in the ensemble systems. Finally, we explain the ensemble methods for the legal RTE task.

3.1 Vector Representation

In the sentence based classifications, it is necessary to represent sentences with vectors. We utilize the LSI method explained in Section 2 to build sentence vectors. For *Siamese* CNN, we use *word2vec* [14] as word embedding, since it is hard to directly apply LSI vector representation to this classifier.

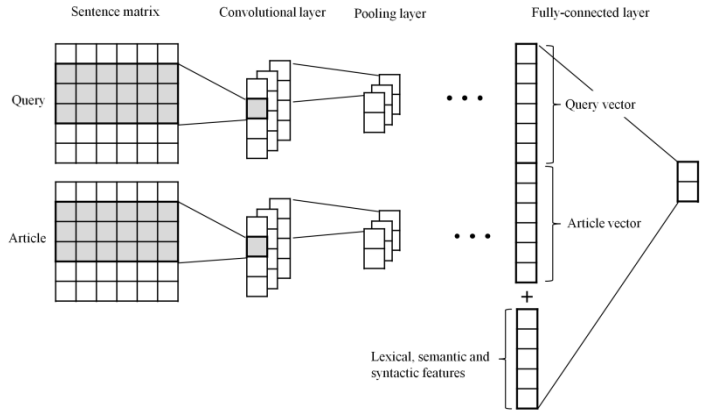
3.2 Classification

Training data consists of queries, relevant articles, and the labels. The labels are indicators that determine whether the relevant articles A entail Q or not Q. The label is binary, thus the entailment decision task can be considered as a binary classification problem. We utilize several conventional classification methods. These are *ordinary logistic regression* [15], *classification and regression tree* (CART) [12], and *support vector machine* (SVM) [12]. In these classifiers, input vectors compose of LSI vector and additional features, which are lexical, semantic and syntactic similarity measurements between Q and A.

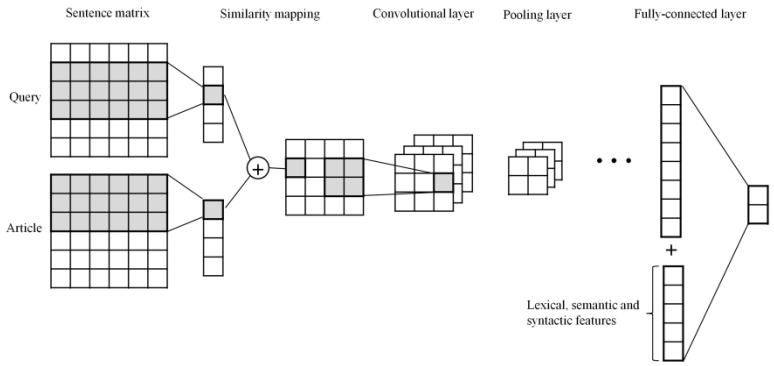
We use two *Siamese* CNN architectures introduced in [16]. The architecture 1 is the conventional *Siamese* CNN, as shown in Fig. 1 (a). It applies convolutional and pooling layers to the input channels of Q and A, and then trains the interaction between two vector representations with a fully connected layer. The architecture 2 is a modified version of the architecture 1. It uses a matrix formation to represent the similarity mapping, and then applies convolutional and pooling layers as illustrated in Fig. 1 (b).

The *Siamese* network is usually utilized to determine semantic textual similarity between two sentences. This approach is not enough to learn the entailment relationship between Q and A. Thus, we use additional lexical, semantic and syntactic features that are used in Section 2. Because some features in Section 2 are originally

proposed to represent textual entailment, it is expected to learn entailment relationship.



(a) Architecture 1



(b) Architecture 2

Fig. 1. Model architecture for *Siamese* CNN

3.3 Ensemble Methods

The main idea of ensemble methods is to combine several individual classifiers with their confidence levels in order to increase classification performance. In this paper, we utilize two ensemble methods, which are major voting and linear weighting.

Major voting is one of the simplest ways to implement abstract level fusions. The strong classifier, which is combined with several classifiers, results in majority vote

among its member classifiers as its decision. On the other hand, measurement level fusions with linear weighting combine the decision probabilities of their member classifiers with linear weights.

There are several ways to compute proper linear weights. To calculate classifier weights, we utilize the LSM method described in Section 2. The weights computed from the LSM method minimize the decision error of the strong classifier. From the classifier weighting perspective, the major voting method assigns equal weights to individual classifiers.

4 Experiments

This section describes the experimental results regarding the relevant article extraction and RTE. As preprocesses, NLP procedures, which are tokenization, lemmatization and stop word removal, are applied to all the documents.

The dry run data consists of 412 queries and 1098 articles in COLIEE 2016. The number of relevant articles to a query is ranging from 1 to 7. The total number of a pair of a query and a relevant article is 519. We simply consider a pair as an single data. We split the dry run data into training data and validation data. The numbers of the training and the validation data are 335 and 184, respectively.

In the RTE task, we use additional training data to train the classifiers, since the number of given dry run data are relatively small to train the classifiers. To obtain the additional data, we utilize a data augmentation technique. In this competition, both Japanese and English text data are provided. The additional English data are obtained by translating the Japanese data with different machine translators, Google, Microsoft, Systran translators.

4.1 Experiments of Relevant Article Extraction

We briefly summarize the evaluation measure of IR performance. Then, we show the experimental results.

4.1.1 Evaluation Measure

We use the mean average precision (MAP) to evaluate the performance of the relevant article extraction systems. MAP is a standard evaluation metric for IR and it is suitable when a query has one or more relevant documents. MAP for a set of queries is calculated by the mean of the average precision scores for each query:

$$\mathrm{MAP} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{N_j}\sum_{i=1}^{N_j}P(a_i), \tag{11}$$

where N_j is the number of retrieved articles for query j, N is the number of queries, and $P(a_i)$ is the precision at the i-th retrieved article.

4.1.2 Results of Single Similarity Measure and Ensemble Methods

The Table 1 shows MAP scores of the single similarity measures and the ensemble methods. MAP@k is the score when the top k similar articles are retrieved.

As shown in the Table 1, LSI results in the best performance at MAP@5 and MAP@10 in individual measurements. The last two rows show the results of the ensemble methods we propose. The LSM and LDA ensemble methods utilize 6 and 7 similarity measures, respectively. In the LSM ensemble method, LSI, *Lucene* similarity score, role similarity (path similarity), *WordNet* similarity (noun, Wu-Palmer), *WordNet* similarity (noun, Lin similarity), and *WordNet* similarity (verb, path similarity) are used to build the ensemble. LSI, overlap similarity, cosine similarity, substring similarity, role similarity (path similarity), *WordNet* similarity (noun, Wu-Palmer), and *WordNet* similarity (noun, Lin similarity) are utilized in the LDA ensemble implementation. Since we drop the similarity measures that have negative weights in each LSM and LDA optimizations, different measurements are used in both methods. In this experiment, the ensemble approaches show better performance than single measure based methods.

Table 1. Mean accuracry precision on the training data

Measurement	MAP@5	MAP@10
Word overlap	0.505	0.513
Cosine similarity	0.412	0.427
Sub-string similarity	0.386	0.392
Lucene similarity score	0.498	0.500
Longest words sequence	0.431	0.439
Role similarity (Wu-Palmer)	0.082	0.088
Role similarity (path similarity)	0.082	0.091
Role similarity (Lin similarity)	0.106	0.114
WordNet similarity (Noun, Wu-Palmer)	0.263	0.275
WordNet similarity (Noun, Path similarity)	0.097	0.109
WordNet similarity (Noun, Lin similarity)	0.118	0.128
WordNet similarity (Verb, Wu-Palmer)	0.040	0.042
WordNet similarity (Verb, Path similarity)	0.064	0.069
WordNet similarity(Verb, Lin similarity)	0.040	0.043
LSI	0.647	0.665
LSM ensemble	0.711	0.715
LDA ensemble	0.682	0.693

4.2 Experiments of Recognizing Textual Entailment

As described in Section 3.1, LSI vector and the additional textual similarity features are utilized in the logistic regression, decision tree and linear SVM classifiers; 22 features including 15 features made in Section 2 and 7 features suggested in [17] are used. The lists of features are word overlap, cosine similarity, substring similarity,

Lucene similarity score, 3 role similarity scores, 6 *WordNet* similarity scores, negative term ratio, and length ratio. We note that the 7 features in [17] are combinations of word matching, tree structured, and lexical semantic features. On the other hand, *word2vec* based pre-trained word vectors and the 22 features are used in CNN type classifiers.

Table 2 shows the results of the classifiers. Baseline accuracy is 49.46 % when all queries are predicted as "YES". The accuracy of ordinary logistic regression is 48.27 %, which is below baseline accuracy. Linear SVM and CNN architectures show the same accuracy of 49.65 %, slightly better than the baseline accuracy. Among single classifiers, CNN architecture 1 produces the best performance.

In this experiment, we use 2 different sets of classifiers to build the ensemble classifiers - 3 (decision tree, linear SVM, and CNN architecture 1 classifiers) and 5 (additional two classifiers - logistic regression and CNN architecture 2) member based ensembles. The ensemble method, which utilizes majority vote with 3 classifiers, outperforms single classifiers and its accuracy is 59.31 %. However, the LSM based ensemble method produces less accurate results than CNN architecture 1. It is expected that the performance of the ensemble method can be improved when a large volume of training data set is available.

Table 2. Accuracy on the validation data

	Methods	Accuracy (%)
Single classifier	Baseline	0.4946
	Logistic regression	0.4827
	Decision Tree	0.5448
	Linear SVM	0.4965
	CNN architecture 1	0.5793
	CNN architecture 2	0.4965
Ensemble method	Majority vote with 3 classifier	0.5931
	Majority vote with 5 classifier	0.5310
	Least square weighting with 3 classifiers	0.5172
	Least square weighting with 5 classifiers	0.5448

5 Conclusion

In this paper, we proposed the ensemble methods for legal IR and RTE tasks. For legal IR task, we use LSM and LDA ensemble methods to combine single similarity measures. For legal RTE, major voting and LSM ensemble methods are applied to build the classification models. Our experimental results show that the ensemble methods produce better performances than single algorithm results in both problems.

Despite of our elaborate ensemble modeling, there are a couple of difficulties to develop the models for legal IR and RTE. Firstly, the size of data in COLIEE competition is insufficient to train the ensemble classifiers. Secondly, linguistic features in our model tend to represent the characteristics of common texts, not the legal text. As

a future work, it is important to deploy legal domain specific knowledge to the ensemble models.

6 References

1. Maxwell, K. T., Schafer, B.: Concept and Context in Legal Information Retrieval. In: JURIX, pp. 63-72 (2008)
2. Peters, W., Sagri, M. T., Tiscornia, D.: The Structuring of Legal Knowledge in LOIS. Artif. Intell. Law, **15**(2), 117-135 (2007)
3. Competition on Legal Information Extraction/Entailment on Juris-informatics 2016, http://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016
4. Gomaa, W. H., Fahmy, A. A.: A Survey of Text Similarity Approaches. Int. J. Comput. Appl., **68**(13) (2013)
5. Ahonen-Myka, H., Heinonen, O., Klemettinen, M., Verkamo, A. I.: Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery. In: Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, pp. 1-9 (1999)
6. Lucene Similarity Score, http://lucene.apache.org
7. De Marneffe, M. C., Manning, C. D.: The Stanford Typed Dependencies Representation. In: Coling 2008 Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp. 1-8, Association for Computational Linguistics (2008)
8. Wu, Z., Palmer, M.: Verbs Semantics and Lexical Selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133-138, Association for Computational Linguistics (1994)
9. Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML, vol. 98, pp. 296-304 (1998)
10. Kilgarriff, A., Fellbaum, C.: WordNet: An Electronic Lexical Database (2000)
11. Dumais, S. T.: Latent Semantic Analysis. Annual Review of Information Science and Technology, **38**(1), 188-230 (2004)
12. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer, Berlin: Springer Series in Statistics (2001)
13. Chopra, S., Hadsell, R., LeCun, Y.: Learning a Similarity Metric Discriminatively, with Application to Face Verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, pp. 539-546 (2005)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 (2013)
15. Agresti, A., Kateri, M.: Categorical Data Analysis. Springer Berlin Heidelberg. pp. 206-208 (2011)
16. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional Neural Network Architectures for Matching Natural Language Sentences. In: Advances in Neural Information Processing Systems, pp. 2042-2050 (2014)
17. Kim, M. Y., Xu, Y., Goebel, R.: Legal Question Answering Using Ranking SVM and Syntactic/Semantic Similarity. In: JSAI International Symposium on Artificial Intelligence, pp. 244-258, Springer Berlin Heidelberg (2014)

Legal Information Extraction/Entailment Using SVM-Ranking and Tree-based Convolutional Neural Network

Truong-Son Nguyen, Viet-Anh Phan, Thanh-Huy Nguyen, Hai-Long Trieu, Ngoc-Phuong Chau, Trung-Tin Pham, and Le-Minh Nguyen

Japan Advanced Institute of Science and Technology
{nguyen.son,anhphanviet,huy.nguyen,trieulh,
cnphuong,tinpt,nguyenml}@jaist.ac.jp

Abstract. This paper presents a system for the legal information retrieval/entailment task. The system includes two phases. For the first phase, relevant articles were retrieved given a question based on two steps. A set of relevant articles was extracted using cosine tf-idf vectors with query expansion techniques. The most relevant article was then obtained using SVM ranking. We experimented and submitted the result to the COLIEE 2016 competition. For the second phase, we used a tree-based convolutional neural network to recognize entailment relations between questions and articles.

Keywords: legal question answering, information retrieval, entailment extraction

1 Introduction

Studying legal issues from the perspective of informatics is a topic that draws interests from researchers recent years. In this task, one of the important targets is the information extraction/reasoning from legal data including legal relation extraction, textual entailment, etc. Legal question answering is also an essential task in legal issues. Recent work proposed some of the first efforts for this task [7, 12, 5].

In this work, we build a system for the legal information extraction/entailment task. In the first phase which is the information retrieval task, a set of relevant articles were extracted given a question before ranking to obtain the most relevant article. In the second phase, the article-question pairs were then used in an entailment task based on the tree-based convolution neural network to determine whether the article can be used to answer the question or not. We experimented our system on the training data of the COLIEE 2016 competition and submitted the result to the COLIEE 2016 shared task.

We describe phases in our system in Section 2 and Section 3. The experiments are analyzed in Section 4. Section 5 reviews related work in this task. Conclusions are drawn in Section 6.

2 Retrieving Related Articles-Questions

The architecture of our system in the IR task is described in Fig. 1. They have two main steps including information retrieval and ranking. In the first step, for given questions, the system will retrieval top 100 relevant articles/a question based on cal-culate the similarity between a given question and articles based on cosine similarity between their TF-IDF vectors. After the information retrieval step, we re-rank 100 relevant articles of the first step using SVM rank algorithm with the model learned from the training corpus. Finally, for each question we choose the most relevant article after the ranking step and format the results according to the rule of the COLIEE 2016 competition and send it to the Organizer for evaluating the system. In the development phase, we choose files from H18 to H23 for training, H24 and H25 for testing to choose the best configuration. However, when we submit the result to the Competition, we train our models on all provided files (H18 to H25). The details of these steps will described in the remain of this section.

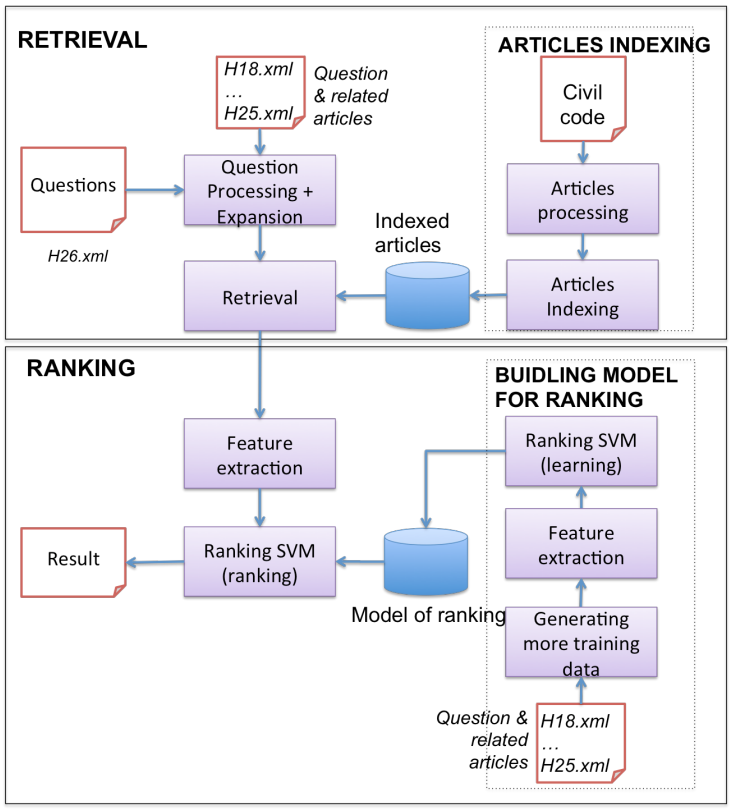


Fig. 1. The model of information retrieval phase

2.1 Articles Processing

This step includes finding the reference expressions in legal texts such as *In paragraph x of articles y; in the provision of article x; ....* Beside that, we also stem articles before indexing them.

2.2 Query Expansion

We used two methods for query expansion using word2vec and wordnet.

Query expansion using word2vec[8]: From questions and their relevant articles in training corpus , extract similar word pairs of word in question and articles base on cosine similarity of their word embedding representation. We select word-pairs that have the cosine similarly value ≥ 0.5 as the related words for query expansion. We think that this methods can retrieve articles that does not share words with the given query.

Query expansion using WordNet[9]: From question, we expand the question by adding synonyms and hypernyms of words in that question. However, this way is not effective because it make the precision score reduce sharply.

2.3 Ranking

Additional Training Data Given a question in a set of questions, learning to rank algorithms require relevant articles and non-relevant articles for that question as the training data to build the model for ranking. However, the training data of COLIEE 2016 only contains question and relevant articles, so it is not enough for learning to rank algorithm work effective. To get more training data for learning to rank algorithm, we generate non-relevant articles of a question by selecting articles that have the low cosine similarity value base on TF-IDF vectors.

Features Extraction We used the following feature list to train the ranking model.

- Cosine similarity between question and article of tf-idf unigram vectors
- Cosine similarity between questions and articles of tf-idf bigram vectors
- Cosine similarity between question and article of tf-idf trigram vectors
- Cosine similarity between nouns in questions and articles
- Cosine similarity between verbs in questions and articles
- Distance feature between questions and article: Euclidean, Manhattan, Levenshtein, Jaccard

Learning and Predicting For ranking, we used SVMRank[1]. Many previous works showed that SVM Rank are an effective tools for training learning to rank models . [6] also used SVM ranking for ranking the result after the retrieval phase in the COLIEE task. The above feature set that we have investigated also improved the performance of systems.

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

3 Identifying Entailment Relationships

To recognize entailment between law articles and queries, we use a tree-based convolutional neural network model (TBCNN) [10]. Firstly, we generate the dependency trees for each question and the relevant articles and then apply the TBCNN-pair model to predict the relation between them. The TBCNN-pair model has two main components including a tree-based convolutional neural network model for capturing document semantics and a matching layer for combining documents' information by heuristics.

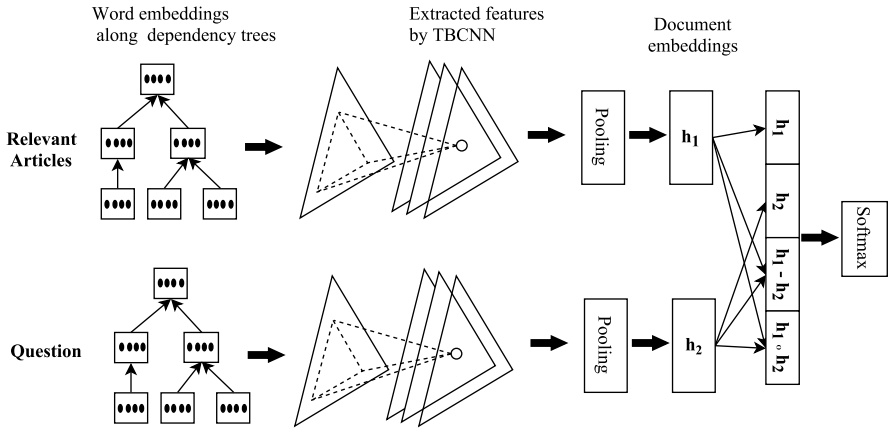


Fig. 2. The TBCNN-pair model for recognizing the relation between a question and the relevant articles

The overview of the model is shown in Figure 2. The tree representations of questions or the relevant articles are made up from the parse trees of their sentences. Firstly, a sentence is parsed to a dependency tree [2]. Eeach node in the tree corresponds to a word in the sentence; a dependency edge is a binary relation:a grammatical relation holds between a governor (also known as a regent or a head) and a dependent. The label of edges shows the grammatical relations between parent nodes and their children. Vector representations of words are trained by a Word2Vec model [3]. Then, the parse trees of sentences in the document are merged to form a hierarchical tree by proposing several hypothesis nodes, which point to the root nodes of the sentences.

After obtaining tree representations for questions and articles, we feed them to the a same model of TBCNN to capture semantics of them. The basic idea of TBCNN is designing a set of subtree feature detector sliding entire the parse

[2] http://nlp.stanford.edu/software/lex-parser.shtml
[3] https://radimrehurek.com/gensim/models/word2vec.html

trees to capture the semantic information. The feature maps of the convolutional layers and the input data have the same lengths and sizes, which are depended on the lengths of articles and questions. To control the number of parameters in the convolutional layer, we apply parameter sharing scheme, in which each word in the corpus has it own weights. In other words, the same words in articles/questions share the same weights and bias. Thus, the model is able to process long articles. To deal with varying sizes, a dynamic pooling layers is added to gather information along different parts of tree. We use the max pooling operation, which takes the maximum value in each dimension. Then, a fully-connected hidden layer is added to mix the information in a questions or its relevant articles. The obtained vector representations of them are denoted as h called *document embeddings*.

The second component of TBCNN-pair model is the matching heuristic layer. The aim of this layer is to aggregate information of a question and its relevant articles by using simple heuristics such as concatenation, element-wise product and difference. These matching layers are futher concatenated, given by

$$m = [h_1; h_2; h_1 - h_2; h_1 \cdot h_2] \tag{1}$$

4 Experiments

We experimented our system using the data of the COLIEE 2016 competition. [4] Although we reported two phases including the information retrieval phase and the entailment phase, we have not obtained results from the phase two so far because of some technical problems. In this section, we show experimental results of our system on phase one. The experimental results of the phase two should be investigated in future work.

For phase one, we analysed aspects of the information retrieval task in our system including: stemming, query expansion, stop-words, and ranking. We conducted experiments and compare different configurations to find the best performance for the system which was submitted to the COLIEE 2016 competition.

4.1 Metrics

Metrics are used in this task including simple information retrieval measures (precision, recall, F-measure [3], accuracy).

4.2 Results

We used the training data: H18-H25. Then, we evaluated on the H27 data set.

Stemming versus No-stemming We show comparison between using stemming and no-stemming while indexing and retrieval. The results are described in Table 1.

[4] http://webdocs.cs.ualberta.ca/ miyoung2/COLIEE2016/

Table 1. Stemming vs No-Stemming while indexing and retrieval, F-measure (%).

Data	H18-H25	H27
Stemming	0.525	**0.538**
No-Stemming	0.505	0.526

Query Expansion versus No-Query Expansion We used the training data in H18-H25 to get similar pairs (consine between word2vec vectors ¿= 0.5) from pairs of questions and relevant articles; then the data H27 was used to evaluate. The experimental results are shown in Table 2. The query expansion step improves the result of information retrieval system step.

Table 2. Retrieval options: query expansion vs no-query expansion. Index options: tf-idf 1gram, no-stemming, no-remove-stopword. F-measure (%).

Data	H18-H25	H27
No-query expansion	0.505	0.526
Query expansion	0.525	**0.550**

Combining Query Expansion and Stemming We made use of combination between stemming and query expansion. The result is described in Table 3.

Table 3. Combining query expansion and stemming. F-measure (%).

Data	H18-H25	H27
No-stemming, no-query expansion	0.505	0.526
Stemming, no-query expansion	0.525	0.538
No-stemming, Query expansion	0.525	0.550
Stemming+Query expansion	0.510	0.550

Removing Stop-words versus No-Removing Stop-words We compared removing stop-words and non-removing stop-words. The results are shown in Table 4. The results showed that removing stop-words leads to the worst performance.

The Effectiveness of Ranking Step The ranking model is trained using SVMRank with cosine similarities and distance features between question and

Table 4. Removing Stop-words vs Non-Removing Stop-words. F-measure (%).

Data	H18-H25	H24	H25	H27
Non-removing stop-words	0.525	0.508	0.595	0.550
Removing stop-words	0.514	0.479	0.581	0.550

relevant articles in training set (H18-H25). The model is evaluated on H24, H24-H25, and H27. Table 5 shows the effectiveness of the ranking step. The results showed that ranking step improved the results on all experimented data sets.

Table 5. Ranking results, F-measure (%)

Data	H24	H24,H25	H27
Before Ranking	0.508	0.547	0.550
After Ranking	0.562	0.583	0.573

In summary, from the experimental results, the best result of the information retrieval phase can be obtained based on the combination of these configurations: no-stemming, query expansion and no-removing stop-words. In the second step of phase one, ranking helped to contribute the better results.

5 Related Work

A basis for a Yes/No Arabic Question Answering System was proposed in [1] using a textual entailment method. For the task of legal information retrieval/entailment task, [7] proposed a system for answering yes/no questions in legal bar exams.

The first shared task of the legal information retrieval/extraction (COLIEE2014) was reported in [12]. There are many methods proposed to solve the task. [14] focused on exploiting reference information to build a question answering system restricted to the legal domain. [6] used ranking SVM and syntactic/semantic similarity.

In the COLIEE 2015 competition, [5] proposed using a convolutional neural network in legal question answering. D. S. Carvalho et al. [2] used lexical and morphological characteristics to extract n-gram features from the query and articles; their own relevance score was then used for Phase 1. They used the query and relevant article are represented in vector space model for Phase 2. In order to classify the queries between Yes label and No label, they used AdaBoost (basic classifier: DecisionStump). Sushimita et al. [11] used voting of three information retrieval techniques: Hiemstra, BM25 and PL2F for the information retrieval task. In [4], a keyword weighting method was proposed and snippet scoring after diving data into snippets. Tran et al., 2015 [13] proposed using hidden markov model for the legal information retrieval task.

6 Conclusion

We build a system for the legal information retrieval/entailment task. For the first phase, the most relevant article given a question was obtained based on two steps. A set of relevant articles were extracted using cosine tf-idf vectors with some additional techniques to improve the performance such as query expansion with word embedding sources and Wordnet. The articles were then re-ranked based on SVM rank to achieve the most relevant article. We experimented and submitted the result to the COLIEE 2016 competition.

For the second phase, we used a tree-based convolutional neural network model to recognize entailment between articles and questions. The model was trained using the COLIEE 2016 data sets. Nevertheless, we have not yet obtained results so far. These experiments should be more investigated and reported in future researches.

Acknowledgment

This work was supported by JSPS KAKENHI Grant number 3050941, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST.

References

1. Wafa N Bdour and Natheer K Gharaibeh. Development of yes/no arabic question answering system. *International Journal of Artificial Intelligence and Applications*, 4(1):51–63, 2013.
2. Danilo Carvarlho, Minh-Tien Nguyen, Chien Tran, and Le-Minh Nguyen. Lexical-morphological modeling for legal text analysis. In *Lecture notes in computer science:New Frontiers in Artificial Intelligence*. Springer, 2016.
3. George Hripcsak and Adam S Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.
4. Y. Kano. Keyword and snippet based yes/no question answering system for coliee 2015. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.
5. Mi-Young Kim and Ken Goebel, Randy Satoh. Coliee-2015 : Evaluation of legal question answering. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.
6. Mi-Young Kim, Ying Xu, and Randy Goebel. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*, pages 244–258. Springer, 2014.
7. Mi-Young Kim, Ying Xu, Randy Goebel, and Ken Satoh. Answering yes/no questions in legal bar exams. In *JSAI International Symposium on Artificial Intelligence*, pages 199–213. Springer, 2013.
8. T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
9. George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

10. Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 130, 2016.
11. S. Pal Sushmita, A. Kanapala. Legal information retrieval task: Participation from ism, dhanbad. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.
12. Satoshi Tojo. Eighth international workshop on juris-informatics (jurisin 2014). In *JSAI International Symposium on Artificial Intelligence*. Springer, 2014.
13. Duc-Vu Tran, Viet-Anh Phan, Hai-Long Trieu, and Le-Minh Nguyen. An approach for retrieving legal texts. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.
14. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. Answering legal questions by mining reference information. In *JSAI International Symposium on Artificial Intelligence*, pages 214–229. Springer, 2013.

Lexical to Discourse-level Corpus Modeling for Legal Question Answering

Danilo S. Carvalho*, Vu Duc Tran, Khanh Van Tran,
Viet Dac Lai, and Minh L. Nguyen

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
{danilo, vu.tran, tvkhanh, vietld, nguyenml}@jaist.ac.jp

Abstract. On top of the traditional Question Answering (QA) problems, Legal QA presents its own set of particular challenges. These challenges mainly involve terminology resolution, searching on heterogeneous information and solving complex abstraction-realization mappings. In this work, we propose a three-stage model for answering legal questions, focused on the terminology and abstraction issues, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE). The three stages comprise: (i) Relevance analysis and ranking of legal articles, (ii) relevance re-ranking and (iii) textual entailment recognition. A set of textual features ranging from lexical to discourse-level is used to train Machine Learning models applied to (ii) and (iii). Experimental results on the previous competition data indicate competitive performance with state-of-the-art methods for the same task. Additionally a discussion on the proposed method's strengths and shortcomings is provided.

1 Introduction

Answering questions is one of the fundamental goals in the field of Natural Language Processing (NLP), receiving an unfading stream of improvements, ranging from base NLP techniques to sophisticated new approaches. It is also highly *domain dependent*, as questions are asked in different ways (i.e., syntactically), with different terminology, and expect diverse answer formulations, according to predefined discourse standards. In the legal domain, demand for higher efficiency Question Answering (QA) methods is on the rise, as we experience an explosive growth in legal document availability on the World Wide Web and specialized systems. Such growth is not accompanied by a matching increase in information analysis capabilities, leading to under-utilization of available legal resources and to potential for information quality issues [1]. The under-utilization of legal resources also brings up the matter of professional ethics and liability on law practice, since having relevant and correct information is of vital importance in legal case solving.

* Supported by CNPq – Brazil scholarship grant

Legal QA, as other QA tasks, can take advantage of improvements made in base NLP techniques, e.g. POS-tagging, parsing, and also from Knowledge Engineering, such as ontology construction and analysis methods. Nevertheless, it has its own set of challenges, which can be grouped in the following three aspects: *terminology*, *information heterogeneity* and *abstraction-realization*. The terminology problem relates to the way that words are used in legal text and how the underlying concepts represented will often differ from common language use, making distributional language modeling much less reliable. The information heterogeneity problem is related to the multitude of information types, such as past decisions, laws and facts, involved in answering a legal question, and the difficulty in identifying and composing them appropriately. The abstraction-realization problem regards the fact that laws are written with abstraction in mind, to cover most possible scenarios of any predicted situation, whereas the practice of the law is aimed at realization of the written law, in order to characterize and substantiate its application. A strong semantically motivated NLP framework is necessary to deal with these challenges, by uncovering term relationships in the legal corpora, facilitating information type identification and linking, and both describing and resolving abstraction relations between parts of the discourse.

In this work, we propose a three-stage method for Legal Question Answering, in the context of the Competition on Legal Information Extraction/Entailment (COLIEE), covering the tasks of Information Retrieval (IR) and Recognition of Textual Entailment (RTE). The stages are: 1) Relevance analysis, 2) Relevance re-ranking and 3) Entailment classification. The method is based on a mixed n-gram language model for relevance analysis, and a set of syntactic, semantic and discourse features, used to train separate Machine Learning models for pair-wise ranking and binary classification, for the relevance re-ranking and entailment classification stages, respectively. The terminology aspect of Legal QA is accounted in all three stages, while the abstraction-realization aspect is only accounted for stages 2 and 3.

The remainder of this paper is organized as follows: Section 2 presents related works and relevant results; Section 3 details the Legal Question Answering problem and the COLIEE competition shared task; Section 4 explains our approach to the competition problem; Section 5 presents the experimental setting, results and some discussion about the findings; Finally, Section 6 offers some concluding remarks.

2 Related Work

Recent studies in the Legal Information Retrieval task can be divided according to their focus in Knowledge Engineering (KE) or Natural Language Processing (NLP) methods. On the KE front, Kim et. al. [2], presented an ontology-based model for law retrieval centered on a R&D and business perspective, applied to Korean law. A query is regarded as a network of legal terms, which is positioned in the document network according to its semantic relations, using the page-rank algorithm, and then ranked based on a classical TF-IDF weighting

scheme over the nouns found in the query through morpheme filtering. Santos et. al. [3] addresses the information heterogeneity problem through strong ontology conceptualization, on the perspective of consumer disputes in the air transport passenger domain, applied to European law. Both approaches [2] and [3] share solutions for improving law accessibility, but also the downsides of building and maintaining legal ontologies.

On a more traditional IR and NLP front, Liu, Chen and Ho [4] presented a method called three-phase prediction (TPP) for retrieval of relevant statutes in Taiwanese criminal law, given general language queries. The method was a hierarchical ranking approach for law corpora, featuring a combination of several Information Retrieval techniques, as well as Machine Learning and feature selection.

For the Recognition of Textual Entailment task, an application to the legal domain can be found in the work of Tran et al. [12], which addressed legal text QA with an inference method based on requisite-effectuation structures of legal sentences and similarity measures, on Japanese National Pension Law.

The previous COLIEE competition had two works with improvements over the baseline of the Legal Information Retrieval task (winner and runner-up respectively): Kim et. al. [5] presented a ranking method based on the SVM algorithm, using lemmatized words intersection, dependency pairs and TF-IDF scoring as features for training the model. Carvalho et. al. [6] presented a ranking method called R_2NC (Ranking Related N-gram Collections), based on a mixed size n-gram language model, which used links between the documents (articles) in the legal corpus to build n-gram collections for each of them, and a variant of TF-IDF scoring to rank them. The Recognition of Textual Entailment task improvements were led by Kim et. al. [5], with a Convolutional Neural Network (CNN) based method for classification, using *word2vec* [13] word embeddings and encoded dependency tree structures as features, with dropout regularization for over-fitting avoidance.

3 Legal Question Answering – COLIEE

Answering a legal question consists in: (i) finding out the necessary knowledge for understanding a given law related question and (ii) providing the appropriate and correct answer to it. In the context of the *Competition on Legal Information Extraction/Entailment (COLIEE)*[1], the question is a legal statement (broad or situational) and the necessary knowledge is encoded in the law itself, presented as a set of articles that compose a fragment of the Japanese Civil Code. The legal statement can be true or false, according to the interpretation of the relevant civil code articles, hence the appropriate answer is either affirmative or negative. The Japanese Civil Code is composed by a collection of numbered articles, each one containing a set of declarations pertaining to a specific topic of the law, e.g., labor contracts, mortgages.

[1] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

In COLIEE 2016, activities (i) and (ii) are separated in corresponding *phases*, with an additional one combining both:

- Phase One (IR): retrieving relevant articles to a given question from all Japanese Civil Code, given a set of YES/NO questions.
- Phase Two (RTE): evaluating the entailment relationship between the question and retrieved articles.
- Phase Three: combination of Phase One and Phase Two, the system shall retrieve a list of relevant articles given a question, and then decide the entailment relationship between the retrieved articles and the provided question.

Legal text is inherently different from other types of written communication, due to the nature of both its content and intent: it is written to express rules and situations on which they apply. This should be done in an abstract, but at the same time unambiguous way, such that the rules will be applied only to the intended cases and no case is covered by multiple, conflicting rules. Those requirements produce a language with stricter terminology and syntax, a higher abstraction level, and with semantics that are foreign or even conflicting with common language use. Such characteristics make the use of *Distributed Semantics* to be *corpus specific* on legal text. However, for answering legal questions it is important to distinguish between the corpus specific and common senses of terms, since both of them are used, specially in questions. Another noteworthy characteristic of legal text is its preference for longer sentences, making automatic parsing more difficult.

Knowing the characteristics of legal text makes it possible to focus on a specific set of concerns when applying NLP to question answering, namely the correct identification of corpus specific vs. common sense of terms, and also the breakdown and capture of important syntactic structures, e.g., argument modification (for negation), requisite and effectuation support. For COLIEE, information heterogeneity is a lesser problem, since the answers are guaranteed to be found on the specified fragment of the civil code.

4 Proposed Approach

Once a core set of goals and concerns was established, the question answering problem was divided into three stages: (i) relevance analysis and ranking, (ii) relevance re-ranking, (iii) entailment classification. This division was motivated by the authors previous experience in the competition and to enable both independent retrieval and entailment recognition in phases one and two, as well as sequential processing in phase three.

The answering process flow is as follows: Firstly, given a single question, a ranked list with a limited number of relevant articles is obtained by using R_2NC [6] (Section 4.3). Next, a set of syntactic, semantic and discourse features is extracted from each retrieved article (Section 4.2) and the relevant article list is re-ranked using SVM^{Rank} [7] with the extracted features. The selection of relevant articles is then decided by taking into account the final re-rank position and the original R_2NC score (Section 4.4). Finally, the same set of features, plus

the R_2NC score is provided to a linear kernel SVM binary classifier, for each selected article, to obtain an entailment relationship between it and the given question. After classification result, a set of bias thresholds is used to decide the relationship. A diagram of the overall process flow is shown in Fig. 1.

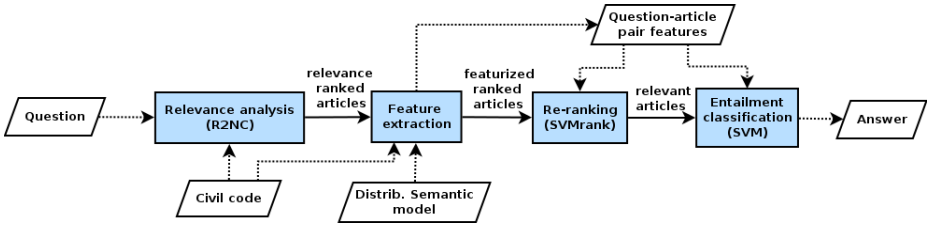


Fig. 1. The process flowchart for the legal question answering method. For COLIEE, the answer is a binary YES/NO value.

Training is done separately for each stage. For relevance analysis, both the civil code articles and training data are analyzed and a mixed n-gram model is built, containing both lemmatized n-gram and link information (Section 4.3). For the re-ranking stage, a leave-one-out session of relevance analysis is run on the training data, generating a R_2NC ranked list of articles for each question. Those lists are used as training data for SVM^{Rank}, after the featurization process described in Section 4.2, producing an article-question re-ranking model. The entailment classification stage is trained by simply featurizing phase two training data the same way as in the re-ranking stage and providing it as training data for a linear kernel SVM classifier, obtaining a question-article entailment classification model.

Sections 4.1 to 4.5 explain in detail the corpus analysis and feature construction processes, as well as each stage of the question answering process.

4.1 Corpus analysis

Analysis of the Japanese Civil Code was conducted in the aspects of lexical and semantic content, syntactic dependency structures and discourse links. As terminology plays an important role in Information Retrieval, the goal of the first was to establish a common lexical-semantic index that would allow efficient question-article term association, while at the same time being able to distinguish between corpus-specific and common language sense use. This was done in two simultaneous ways: (a) the calculation of corpus-specific n-gram statistics and (b) Distributional Semantics modeling on combined *common + legal* text data. The first would allow coarse-grained filtering through lexical relatedness measures, e.g., TF-IDF, while the latter enables more fine-grained semantic distinction.

Applying Distributional Semantics modeling, however, is complicated by the difference in volume of the available data for common vs. legal text uses, with the former being much more available than the latter. To deal with this problem, a dataset balancing technique was employed, in which the legal text was replicated until it composed a certain fraction (around 25%) of the combined data. The resulting data was used to train a *word2vec* [13] embedding model.

On the matter of syntax, dependency structures were chosen due to the interest in capturing clause modifiers, specially negations, e.g. "no", "never", that apply to both effectuation clauses and article links, e.g., "...the Manager shall **not** be liable to compensate for damages...". Those structures have low corpus specificity and can be obtained by using state-of-the-art dependency parsers with no modifications. Structural matching of sentences is also facilitated by comparing dependency tags, and can be combined with lexical and semantic similarity. The stock version of *SyntaxNet* [8] is used for dependency parsing.

Furthermore, discourse links are also an important source of information, since they bind together complementary information about a topic and allow contradictions to be found. In an Information Retrieval setting where articles are considered *documents*, the article and paragraph references are defined as the discourse links and are typed into a set of classes ("plain", "case" and "provisioning") and indexed. Finally, to take maximum advantage of the discourse link information, all text processed from the civil code is indexed at paragraph level, allowing efficient reference resolution. Fig. 2 illustrates the corpus structure.

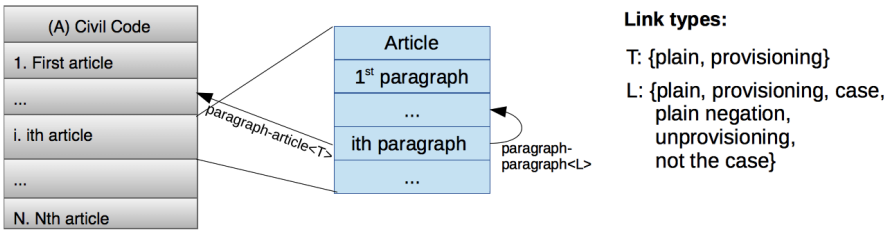


Fig. 2. The structure of the Japanese Civil Code corpus. Link types specify how the arguments in both sides of the reference are related. For example, a "case" link type defines an argument as valid on the particular condition denoted by the referent.

4.2 Feature construction

Once a clear understanding of the corpus structure was reached, it was necessary to find an appropriate way to associate questions with their respective answers. For this reason, a set of features was developed, incorporating information from all the previously mentioned corpus aspects (lexical, semantic, syntactic dependency and discourse links), with the goal of training Machine Learning models to differentiate among several association characteristics.

The features are defined as follows:

- **Dependency node semantic similarity**: *word2vec* embedding cosine similarity, calculated over words matching predefined pairs of dependency tags in the question and article paragraph, respectively. For example, a pair (*nsubj*, *dobj*) would match a question's "nsubj" (a noun subject) and an article paragraph's "dobj" (a direct object), and the similarity would be calculated. Each pair is regarded as a single feature. Dependency pairs considered for semantic similarity were: *(nn,nsubj), (nsubj,nsubjpass), (pobj,dobj), (poss,nn), (pobj,nsubj), (dobj,nsubj)*, plus same element pairs.

- **N-gram intersection**: The size of the n-gram intersection (with n from 1 up to 10) between a question and paragraph, normalized by the paragraph's n-gram set size.
- **Paragraph links**: normalized index of link destinations for each paragraph (up to 6).
- **Paragraph link types**: boolean indicating the pertinence relationship of the respective link to a certain type or its negation. There are 3 possible types: "plain", "case", "provisioning", and their respective negations: "plain negation", "not the case" and "unprovisioning". They define the discourse relations between a paragraph and the content it refers to. For example, an unprovisioning link can be found in Article 295, paragraph 2: "The provisions of the preceding paragraph shall not apply…".
- **Negation similarity**: *word2vec* embedding cosine similarity between negated terms in the question and paragraph, normalized by the number of negated terms. Negated terms are obtained from negation links in the dependency tree. If either one of the question or paragraph does not contain negated terms, similarity is set to zero. If neither contain negated terms, similarity is set to one.

The features are calculated for the association question-paragraph, for each paragraph of each article evaluated. Since articles have different numbers of paragraphs, the paragraph number was fixed at 4, as there were very few cases of articles longer than that. Exceeding paragraph features were filled with placeholder values (0 for similarity and boolean features and -1 for link indexes). Similarity features were designed to facilitate terminology sense distinction and abstraction-realization issues, e.g., "neighbor" $\leftrightarrow$ "agent" $\leftrightarrow$ "person", while the link features were intended to highlight discourse conformity/contradiction, and facilitate entailment classification.

4.3 Relevance analysis

The relevance analysis stage was done entirely with R_2NC [6], which can be summarized in the following process:

1. Collect the entire content for each article, including section title;
2. Check references between articles and annotate accordingly;
3. Tokenize and POS-tag;
4. Remove stopwords: determiners, conjunctions, prepositions and punctuation;
5. Lemmatize words;
6. Generate n-grams;
7. Expand the n-gram set, by including references n-grams;
8. Associate article number and references;
9. Store the model.

Except for step 4, each step is responsible for adding new information to the model. The information is obtained either from the text, e.g., section title, references, or from morphological analysis, e.g., POS-tags, lemmas. If an article

have references, its n-gram set is expanded with the references' n-grams. This is done so that all the necessary information for interpretation of any single article is self-contained. Besides the n-grams, links between the articles are also stored. To include the training data information, the same process is repeated for the questions, and n-gram sets from the trained questions are used to expand the associated articles' n-gram models. Tokenization and lemmatization were done using *NLTK*[2] (v. 3.0.2) with the Punkt tokenizer and WordNetLemmatizer modules, respectively. Those modules were used with their unchanged default models and settings, trained with the Punk corpus and WordNet, respectively. POS-tagging was done using *Stanford Tagger*[3] (v. 3.5.2), using the unchanged *english-left3words-distsim* model, which is trained on the part-of-speech tagged WSJ section of the Penn Treebank corpus. Fig. 3 illustrates the n-gram model creation scheme.

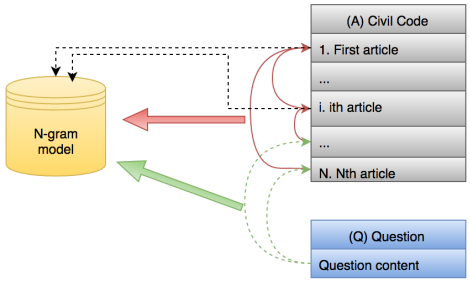


Fig. 3. The n-gram model construction scheme. Both article-article and question-article links are stored, and the respective document n-gram sets are associated. A single association index is generated for each article.

The relative relevance of an article with regard to the content of a question is ranked by applying the following scoring formula:

$$score = \frac{\sum_{\forall t} idf(t)}{I_q \times |q\_ng\_set| + I_{art} \times |art\_ng\_set|}, \quad t \in (q\_ng\_set \cap art\_ng\_set) \quad (1)$$

where q_ng_set is the set of n-grams for the question, art_ng_set is the set of n-grams for the article in the stored model, I_q is the relative significance of the question n-gram set size and I_{art} is the relative significance of the article n-gram set size. $idf(t)$ is the Inverse Document Frequency for the term t over the articles collection

$$idf(t) = log\frac{N}{df_t} \quad (2)$$

where N is the total number of articles and df_t is the number of articles in which t appears. Both I_q and I_{art} are parameters.

[2] www.nltk.org
[3] nlp.stanford.edu/software/tagger.shtml

4.4 Relevance re-ranking

Preliminary analysis of the previous competition data showed that R_2NC could achieve a 0.8 recall when using only the first 20 ranked articles (out of 1106). This meant that most unrelated content was already being filtered out and a directed re-ranking approach was appropriate to improve precision on the reduced relevant set. Further observation of the ranked lists revealed that the abstraction issue was responsible for a considerable part of the loss in precision.

The features described in Section 4.2 were designed taking this into consideration and were used to train a question-article re-ranking model using SVM^{Rank} [7]. Training data is obtained by running a leave-one-out session of R_2NC on COLIEE's training questions. This results in a ranked list of relevant articles for each question (limited to 20), for which the features are calculated. The article lists are then presented as training inputs to SVM^{Rank}. If an article list does not contain the correct relevant ones, those are added to the end of the list, with their corresponding ranks (1^{st}, 2^{nd}, ...). The re-ranking stage is done by featurizing R_2NC outputs and using them as inputs for the trained SVM^{Rank} model. Final selection of the relevant articles is done by taking the first ranked from the list, and optionally the following ones for which the distance $d = score(q, a[1]) - score(q, a[i]) < extend\_thresh$, where q is the given question, $a[i]$ is the i^{th} ranked article, $score(q, a)$ is the R_2NC score and $extend\_thresh$ is a threshold parameter. Fig. 4 illustrates the re-ranking process.

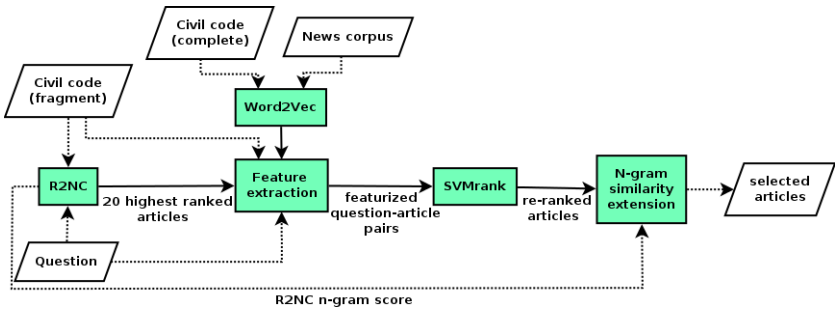


Fig. 4. The re-ranking process flowchart. The n-gram similarity extension step is decided upon a user-defined threshold $extend\_thresh$. This way, articles arbitrarily close to the 1st ranked one may be also selected.

4.5 Recognition of Textual Entailment

For the final stage, observation of the previous competition data pointed to the interpretation of argument negation as a key factor in textual entailment recognition performance. To deal with this, discourse link type and negation information where included in the feature set, under the hypothesis that they carry enough information to induce entailment discrimination on the appropriate cases, when used with a Machine Learning classifier. In this stage, the classifier of choice was the linear kernel SVM, with the SVM^{perf} [9] implementation.

Training the classifier was done by calculating the same features as in the previous stage, but directly on COLIEE's training data, with R_2NC score as an

additional feature. All pairs question-article were featurized and the labels set to the corresponding question label.

Entailment recognition was done in the following way:

1. Featurize the relevant/selected articles for a given question.
2. Classify each pair question-article.
3. For each pair question-article:
   (a) If maximum n-gram intersection feature > *strong_intersection_thresh*, then question answer is "Y".
   (b) If class = "Y" and the maximum n-gram intersection > *weak_intersection_thresh*, then question answer is "Y".
   (c) Else, class = "N", then question answer is "N".

where *strong_intersection_thresh* and *weak_intersection_thresh* are threshold parameters, intended to bias the entailment classification towards the answers found for articles with highly intersecting question-paragraph pairs.

5 Experiments and Results

5.1 Experimental Setup

The legal question answering dataset was obtained from the published data for the COLIEE shared task [4], consisting in a text file with a fragment of the Japanese Civil Code translated into English and a set of XML files with training data. The training set for the three tasks contains 412 pairs (question, relevant articles). Experiments where divided in phases one and two only, dealing with Information Retrieval and Textual Entailment methods respectively.

Additional data used in the experiments include the training segment of "1 billion word language model benchmark" corpus [10] and the complete Japanese Civil Code[5], which were used to train the Distributional Semantics model. The combined size of the corpora after balancing is approximately 1.2 billion words.

Experiment data was divided into training and validation, taking advantage of the fact that the previous competition test data was distributed as part of the current one's training data. By using the previous competition as validation data, a direct performance comparison was possible, facilitating the evaluation process. Competition files *riteval_H{18..23}.xml* were used for training and *riteval_H{24, 25}.xml* for validation.

5.2 Parameter adjustment

Parameter adjustment was done separately for each stage. R_2NC parameter k (the maximum n-gram size) was kept as in [6] ($k = 3$), since changing it did not offer performance improvements on a leave-one-out test over COLIEE 2016 training data. However, increasing I_q to 0.98 and thus decreasing I_{art} to 0.02 incurred in a 0.04 F-score increase. For re-ranking, SVM^{rank} was run with parameter $C = 2000$ and $extend\_thresh = 0.5 \times 10^{-7}$. C was adjusted by starting

[4] webdocs.cs.ualberta.ca/~miyoung2/COLIEE2016/

[5] www.japaneselawtranslation.go.jp

from 200 and increasing its value by 200 until performance dropped or model convergence could not be reached in the validation set. *extend_thresh* was initially set to 0.5 and then divided by 10 until no improvements could be found. For entailment classification, SVM^{perf} was run with parameter $C = 400$ and thresholds *strong_intersection_thresh* and *weak_intersection_thresh* were set to 0.9 and 0.4 respectively. Adjustment was made in the same way as in the re-ranking stage, but with C increasing by 100 and *strong_intersection_thresh* and *weak_intersection_thresh* starting at 1.0 and decreasing by steps of 0.1. *word2vec* parameters were set as: $d = 200$, $cbow = 0$, $window = 0$, $negative = 0$. SyntaxNet was run with default parameters. Feature relevance analysis was performed on the training and validation data through SVM attribute ranking [11]. It revealed the most relevant attributes to be the dependency node similarities between relative clause modifiers and nouns (*rcmod* and *nn* dependencies), followed by the discourse link types. The least relevant were the paragraph links from higher numbered links ($\geq 4^{th}$). Removing less relevant features did not improve results in the validation data, thus no feature selection was performed.

5.3 Baselines

As a COLIEE directed effort, the single baseline used in this work was the previous competition results, favored by the possibility of direct performance comparison given by the release of the corresponding test data with ground truth labels. The results are compared for both the winner [5] and the runner up [6] in phase one, and only for the winner [5] in phase two. Phase three results were considered as consequence of the performance in phases one and two, so they were not evaluated.

5.4 Evaluation Method

For the relevance analysis stage, leave-one-out validation was used to evaluate the potential recall of the model for a limited size ranked list of articles. Performance for phase one was evaluated using precision (P), recall (R) and F-measure (F) as metrics (Eqs. (3), (4) and (5)). In phase two, accuracy (A) measurement is used (Eq. (6)).

$$P = \frac{Cr}{Rt} \quad (3) \qquad R = \frac{Cr}{Rl} \quad (4) \qquad F = \frac{2(P * R)}{P + R} \quad (5) \qquad A = \frac{Cq}{Q} \quad (6)$$

where Cr counts the correctly retrieved articles for all queries, Rt counts the retrieved articles for all queries, Rl counts the relevant articles for all queries, Cq counts the queries correctly confirmed as true or false and Q counts all the queries.

5.5 Pre-competition Results

Experiment results on the validation set are presented in Tables 1 and 2.

The results indicate a noticeable improvement in both tasks, specially in phase one. Phase one results also indicate a recall consistent with state-of-the-art methods for similar legal corpora, slightly surpassing TPP's [4] mark of 0.52

Table 1. Experiment results for phase one (IR). All systems were tested on the file *riteval_H24.xml* from the COLIEE dataset.

Method	Precision	Recall	F-measure
Kim et. al. [5]	0.6329	0.4902	0.5525
Carvalho et. al. [6]	0.5663	0.4608	0.5081
This method	0.6707	0.5392	**0.5978**

Table 2. Experiment results for phase one (RTE). All systems were tested on the file *riteval_H25.xml* from the COLIEE dataset.

Method	Accuracy
Kim et. al. [5]	0.6667
This method	**0.6969**

for the top 3 ranked law articles, while using considerably less training data: 267 documents for R_2NC against 1518 documents for *TPP*.

Phase two results also indicate that the feature set developed in this work can help deciding on textual entailment by including relevant discourse information, in the form of reference typing and argument negation matching.

5.6 Error Analysis and Discussion

Observation of the misranked and misclassified cases helped understanding the improvements obtained over R_2NC, as well as the limitations of the proposed method.

The following example shows a case of R_2NC misrank, while the proposed method succeeds:

Table 3. Example question for which the re-ranking approach improves the result. R_2NC rank shows the position before the re-ranking.

ID	Question	Rel. article	R_2NC rank	re-rank
H24-19-1	In cases where the obligee (C) exercises the credit of sale value vested in the obligor (A) against (B), and filing the action regarding to the credit, if the upholding judgment of the action is established, the credit is deemed to extinguish by the performance	Article 423 (1) An obligee may exercise the right vested in the obligor in order to preserve his/her own claim;provided, however, that, this shall not apply to rights which are exclusive and personal to the obligor. (2) Until exercised by way of subrogation admitted in a judicial proceeding, the obligee may not exercise the right set forth in the preceding paragraph unless and until his/her claim has become due;provided, however, that, this shall not apply to any act of preservation.	19th	1st

In this case, the use of structural similarity features allowed abstraction to be applied (e.g., "right" → "credit of sale value") and the lexical aspect to be overcome in the ranking. In this case, 18 other articles had a greater share of term occurrences such as "obligee", "right" and "credit" as keywords.

In the example shown in Table 4, however, the proposed method is still unable to correctly rank the articles. In that case, a very high lexical match (from Article 21, not included due to space constraints) overweights all the remaining features. A completely different semantic approach would be needed to address such case.

Table 4. Example question for which the re-ranking approach is still unable to improve the position of the relevant article. R_2NC rank shows the position before the re-ranking.

ID	Question	Rel. article	R_2NC rank	re-rank
H24-2-4	In cases where a person with limited capacity manipulates any fraudulent means to induce others to believe that he/she is a person with capacity, his/her juristic act has effect even if there is a mistake in any element of the juristic act in question.	Article 95 Manifestation of intention has no effect when there is a mistake in any element of the juristic act in question; provided, however, that the person who made the manifestation of intention may not assert such nullity by himself/herself if he/she was grossly negligent.	2nd	2nd

For phase two, an intriguing development was found, as no relevant article in phase two validation data contains explicit clause negations, although some questions do. Despite that, questions were misclassified when the negated link features were removed, suggesting an indirect (implicit) link between question clause modifiers and the articles link chain. However, a more detailed investigation is needed to expand such hypothesis.

6 Conclusion

Legal Question Answering presents a set of particular challenges, on top of the traditional QA problems. These challenges mainly revolve around terminology resolution, searching on heterogeneous information and solving complex abstraction-realization mappings. In the context of the Competition on Legal Information Extraction/Entailment (COLIEE), we propose a three-stage model for answering legal questions, focused on the terminology and abstraction issues.

Starting with relevance analysis, a mixed n-gram model is built from the Japanese Civil Code corpus and training pairs of questions and their relevant articles. The model is then used to rank civil code articles according to their relevance to the question. Next, a limited number of articles is taken from the previously ranked list and then re-ranked, by extracting a set of lexico-syntactic, semantic and discourse link features from the ranked question-article pairs, and using then as inputs for a Learning-to-Rank method. Agglutination of close matches by first stage score threshold is also performed, to increase the recall. The re-ranking system acts as a fine grained relevance analyzer, working on a limited sample but with more detailed information. In the final stage, the same features used in the second stage is combined with the first stage relevance score to serve as input for a binary classifier on the entailment relationship of question-article pairs. Posterior biasing is applied to decide the entailment for questions with multiple relevant articles.

Experimental results with the previous competition data indicate that the proposed method improves over the winning results by 0.04 F-score on phase one (Information Retrieval) and by 3% accuracy on phase two (Recognition of Textual Entailment). The results are also consistent with state-of-the-art results in similar legal corpora. Analysis of the results provided important information about limitations of the proposed approach. Those shall be addressed in future work.

As future work, a secondary level of semantic processing, aimed at argument reasoning, can be developed for addressing some difficult questions. An alternative Distributional Semantic approach with a broader scope shall also be considered.

Acknowledgements

This work was supported by JSPS KAKENHI Grant number 15K16048, JSPS KAKENHI Grant Number JP15K12094, and CREST, JST.

References

1. Robert C. Berring: "The heart of legal information: The crumbling infrastructure of legal research". *Legal information and the development of American law*. St. Paul, MN: Thomson/West, 2008.
2. Wooju Kim, Youna Lee, Donghe Kim, Minjae Won, and HaeMin Jung: "Ontology-based model of law retrieval system for R&D projects." *In Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World*, p. 26. ACM, 2016.
3. Cristiana Santos, Vctor Rodriguez-Doncel, Pompeu Casanovas, and Leon van der Torre: "Modeling Relevant Legal Information for Consumer Disputes." *In International Conference on Electronic Government and the Information Systems Perspective*, pp. 150-165. Springer International Publishing, 2016.
4. Yi-Hung Liu, Yen-Liang Chen and Wu-Liang Ho: "Predicting associated statutes for legal problems." *Information Processing & Management* 51.1: 194-211, 2015.
5. Mi-Young Kim, Ying Xu, Randy Goebel: "A Convolutional Neural Network in Legal Question Answering." *In Proceedings of the 9th International Workshop on Juris-informatics (JURISIN 2015)*, pp. 211-222, 2016.
6. Danilo S. Carvalho, Minh-Tien Nguyen, Chien-Xuan Tran, and Minh-Le Nguyen: "Lexical-Morphological Modeling for Legal Text Analysis." *Lecture Notes in Computer Science: New Frontiers in Artificial Intelligence*, 2016.
7. Thorsten Joachims: "Training Linear SVMs in Linear Time." *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.
8. Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins: "Globally normalized transition-based neural networks." *arXiv preprint arXiv:1603.06042*, 2016.
9. Thorsten Joachims: "A Support Vector Method for Multivariate Performance Measures." *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.
10. Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson: "One billion word benchmark for measuring progress in statistical language modeling." *arXiv preprint arXiv:1312.3005*, 2013.
11. Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik: "Gene selection for cancer classification using support vector machines". *Machine learning 46*, no. 1-3, 389-422, 2002.
12. Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen and Akira Shimazu: "Answering Legal Questions by Mining Reference Information". *New Frontiers in Artificial Intelligence*. Springer International Publishing: 214-229, 2014.
13. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean: "Distributed Representations of Words and Phrases and their Compositionality". *In Proceedings of NIPS*, 2013.

Civil Code Article Information Retrieval System based on Legal Terminology and Civil Code Article Structure

Daiki Onodera and Masaharu Yoshioka

Graduate School of Information Science and Technology Hokkaido University
onodera@kb.ist.hokudai.ac.jp, yoshioka@ist.hokudai.ac.jp

Abstract. In this paper, we propose a civil code article information retrieval (IR) system that supports to find relevant articles of civil code for the questions of Japanese Bar Exam. Since IR system for question and answering should retrieve documents with important keywords in the questions, we use ABRIR[1] that calculate the similarity of documents by using BM25 and applying penalties based on the information of the existence of such important keywords. In order to utilize ABRIR for civil code article IR system, it is necessary to analyze important keywords for the questions of Japanese Bar Exam. Articles of civil code use specific legal terminology and have special article structure. For example, articles of general articles introduce general code including definition of legal specific terms and followings articles discuss the specific cases. Therefore, we propose to use terms of legal terminology as important keywords and a framework to divide articles into condition parts and other parts. However, in addition to the legal terminology of civil code, there are several legal specific terms used in the questions. In order to utilize such legal terms for retrieving civil code articles, we propose a framework to find related legal terms in the civil code by using definition sentence retrieved by Google. Additionally, since most of the articles for applied mutatis mutandis just contains reference to the other articles, we propose to use combination articles that contains keywords of the article and keywords of referred articles for retrieving combination of these articles. We evaluate our system by using Japanese Bar Exam data set from 2006 to 2013 (COLIEE 2015 dataset[2]) and confirm its effectiveness.

Keywords: legal documents, terminology, civil code, article structure, boolean IR model.

1 Introduction

We propose an IR system that supports to find relevant articles of civil code for questions in Japanese Bar Exam based on legal terminology and civil code article structure, and evaluated our system by using Japanese Bar Exam dataset (COLIEEE 2015 dataset[2]). Queries in Bar Exam have many types. For example, there are queries that uses almost same terminology used in civil code (e.g., a

simple modification of the civil code articles). On the contrary, there are queries that uses terms that are not used in the civil code articles (e.g., keywords related to concrete examples and/or legal specific terms that are not used in the civil code articles). For the former case, it is not so difficult to retrieve such related articles, but it is difficult to retrieve corresponding articles because of the term mismatch. In such a case, an IR system tries to retrieve documents based on the comparison of general terms used in the articles and questions for calculating the similarity. In this situation, it is difficult to retrieve the relevant document which contains significant legal terminology by simple word matching. Therefore we assume that IR system for Japanese Bar Exam question answering should retrieve important keywords in the questions, we use ABRIR [1] for legal terminology to calculate the similarity of documents and use the Okapi BM25[3] weighting to calculate the score of each document. In addition, since there are many articles that explain the exceptional case of articles introduced in general provision, it is better to consider structure of articles (condition parts and other parts) for retrieving related articles. For example, when the question has a part that explain the condition of the case, it is better to compare such part of a query with condition parts of such articles. Therefore, we propose a framework to split the articles and queries into condition parts and rest parts and calculate similarity based on the comparison with condition parts, rest parts and all sentences.

Even though terms from legal terminology in civil code is important keywords for retrieving articles, there are many questions that uses different terminologies used in civil code. There are several cases for this terminology mismatch. One is simple case related to the compound words. For example, there are many compound legal terms are used in the civil code (e.g., ” 損害賠償請求権 (Right to Demand Compensation) ”), but questions may refer to this right by using decomposed terms in different expression (e.g., ” 損害 (Right) を (wo: postpositonal particle) 賠償 (Compensation) する (do) 権利 (Right)”). Therefore, we propose to add index of the sentence by using three different vocabulary sets. First is terms in legal terminology, and second is decomposed legal terminology (set of terms extracted by decomposing legal terminology using POS tagger). The last is terms extracted from all articles by POS tagger.

In these approach, we use Boolean IR with the penalty for legal terminology and Okapi BM25 for words based on ABRIR[1], and each weight for calculating similarity for conditional part and rest part as arbitrarily parameters. For example, we can set the parameter of weight as conditional part similarity 25%. rest part similarity 25%, and all part similarity 50%. In such a case final similarity score is weighted summation of similarity score of those three scores. In addition, we also propose two additional features to improve the performance of the IR system. One is query keyword expansion based on the legal terms that are not used in civil code. The other is a method to construct combination articles for representing applied mutatis mutandis. We analyze the general characteristics of our proposed framework by conducting experiment by using COLIEE2015 data and discuss the effectiveness of the proposed IR system. COLIEE tasks

are composed of three phases, we applied our system for phase 1(legal information retrieval Task, input is a bar exam 'Yes/No' question and output should be relevant civil law articles).

2 Boolean IR Model and Probabilistic IR Model for Legal Information Retrieval

Our task is to retrieve the appropriate relevant document(articles in civil code) for the query in Japanese Bar Exam. To identify the feature of article, legal terminology have very important role, for example, " 成年後見人 (Guardian of Adult)" and " 保証人 (Guarantor)". When the article is similar with the query in terms of general words and not similar in terms of legal terminology, we assume that the article is not important. Therefore, our system is an IR system that has following main ideas.

1. we use the Boolean IR model[1] for civil code focused on legal terminology, and ranked for each query with Okapi BM25[3] which is the probabilistic IR model to three vector space(legal terminology, morphemes of legal terminology and all words removed stop words).
2. if the query and the article have conditional part, we calculate the similarity for conditional part and rest part separately.

However, civil code has some features, such as articles quoted from other articles -"A については X 条の規定を準用する (The article X shall apply mutatis mutandis to A)"-, abstract legal terminology containing meaning of several legal terminology appear only in queries-" 用益物権 (Usufruct), 担保物権 (Collateral)"-. Our system takes solutions for these problem into account, and the overview is shown in Figure 1.

2.1 Legal Terminology and Civil Code Article Structure

We pre-processed based on legal terminology and civil code article structure, and it is shown below.

(1) extract legal terminology
There are many legal terms, such as " 物権 (Real Rights)"、" 債権 (Claims)"、" 表見代理 (Apparent Authority)" and those terms are also used in queries about civil code. Since those terms have special meaning, existence of such legal terms in a civil code article is a strong clue to identify whether the article is relevant or not. In order to extract such legal terminology we use Wikibooks version of " コメンタール民法 (Kommentar Civil Code)" [1] for extracting such terms. In this site, important terms used in the articles have links to other page in the Wikibooks or Wikipedia. We collect all terms

[1] https://ja.wikibooks.org/wiki/%e3%82%b3%e3%83%b3%e3%83%a1%e3%83%b3%e3%82%bf%e3%83%bc%e3%83%ab%e6%b0%91%e6%b3%95

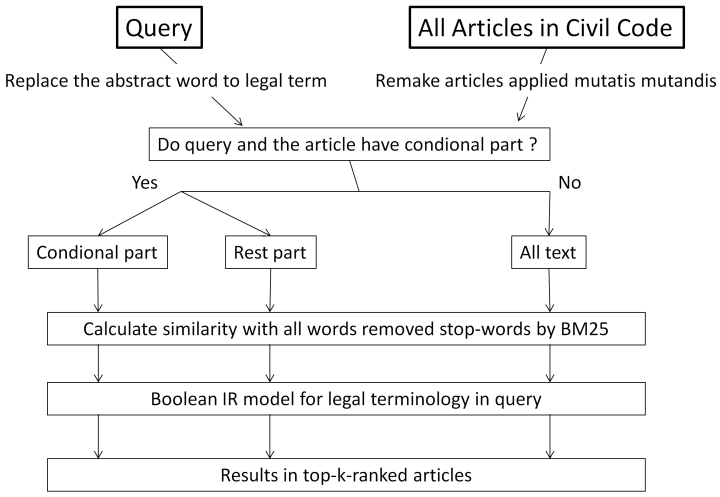


Fig. 1. Overview of IR system

that have such links as candidate legal terms. However, since there are several articles that miss to add links to the legal terms that have links in other articles, we decide to use a list of legal terms and extract such terms by comparing articles with this list. In addition to the link terms, there are varieties of legal terms that are compound nouns using common suffix such as " 権 (right)". In order to extract such compound nouns, we apply technical term extraction system TermExtract[4] to the all civil code articles. TermExtract extracts domain technical terms by checking the term sequence (e.g., compound nouns) and produces ranked list from them. From the output of the system, we selected terms related to actor such as " 通訳人 (Translator)", " 相続人 (Heir)" and right's names such as " 占有権 (possessory right)", " 親権 (parental authority)" from them. Finally, our legal terms list contains 635 terms and 494 terms are from the link information of Wikibooks. By using TermExtract we extract 191 terms and 73 terms (total 113 and 71 terms including terms from the link) as actor and rights respectively.

(2) difference expression between query and article

Besides above compound words, morphemes of compound words sometimes appear in query, such as compound word " 取消権者 (Persons with the Right to Rescind Act)" often appears " 取消 (rescind)" and " 者 (Person)" separately in query. Thus, we use three vector space which one for compound word, another for morphemes of compound words, the other for all words removed stop words, and hyponyms of Japanese WordNet[5] in consideration of appearance of concrete example of articles.

(3) civil code article structure

Since there are many articles that explain the exceptional case of articles introduced in general provision, it is better to compare such part of a query with condition parts of such article:

– Article 124 成年被後見人は、行為能力者となった後にその行為を了知したときは、その了知をした後でなければ、追認をすることはできない。(If an adult ward recognizes his/her act after he/she has become a person with capacity to act, he/she may ratify such act only after such recognition.).

The part of " ～したときは、(If ～)," is conditional part of this article, and the rest is the rest part. If both of the query and the article has conditional part " ～の場合には (If ～)", " ～とき (When ～)" , we divided these into conditional part and rest part, and calculate similarity separately. However, in the case of the query does not have conditional part and relevant document have conditional part, this segmentation have negative effect sometimes. Therefore, when both of the query and the article have conditional part, we split the text into conditional part and rest part by using dependency parsing Cabocha[6], and calculate conditional part's similarity, rest part's similarity and all text(not separated text) similarity separately. We can use weight like conditional part:25%, rest part:25% and all text:50% to calculate similarity as optional parameter.

2.2 Boolean IR Model and Probabilistic IR Model

We use combination of following two methods.

(1) Probabilistic IR model BM25

Our system uses Okapi BM25[3] to rank each article. In Okapi BM25 RSJ weight is used for calculating the importance of query terms based on the existence of the term in the (pseudo-)relevant documents. However, in this problem setting, number of relevant document(s) is small, it is not appropriate to use (pseudo-)relevant document(s). Therefore, we use IDF instead of RSJ:

$$\sum_{t \in q} \left[log \frac{N}{df_t} \right] \cdot \frac{(k_1+1)tf_t}{k_1((1-b)+b \times (L_d/L_{ave}))+tf_t} \cdot \frac{(k_3+1)tf_q}{k3+tf_q} \quad (1)$$

where, N is the count of all documents of civil code, df_t is the document frequency of the term, L_d is the document length, L_{ave} is the average length of all documents, tf_t is the term frequency in the document, tf_q is the term frequency in the query, and b,k_1 and k_3 are control parameters. However, if the query and the article have conditional part, we modified a little the equation(1) to equation(2):

$$\sum_{t \in q} \left[log \frac{N}{df_t} \right] \cdot \frac{(k_1+1)tf_t}{k_1((1-b)+b \times (L_d/L_{ave}))+tf_t} \cdot \frac{(k_3+1)tf_q}{k3+tf_q} \cdot weight \quad (2)$$

where weight is the parameter for calculating similarity by all text or conditional part or rest part. For example, we can set the weight 0.5 for all text, weight 0.25 for conditional part, weight 0.25 for rest part.

We apply BM25 to compound words, morphemes of compound words(hereinafter, this is called "elems") and all words removed stop words(hereinafter, this is called "words").

(2) Boolean IR model for legal terminology

We assume that the document does not contain the legal term in query is not usefulness, and introduce notion of penalty by Boolean IR model[1] for legal term. For example, after we construct the index word list for elem, we calculate penalty to the article does not contain the legal term in the query:

$$Penalty(w) = \beta \cdot \left[log\frac{N}{df_t}\right] \cdot \frac{(k_3+1)tf_q}{k3+tf_q} \tag{3}$$

where N is the count of all document in civil code, df_t is the document frequency of the term, tf_q is the term frequency in the query, and β and k_3 are control parameter.

2.3 Article to be applied mutatis mutandis and legal terms not included in Civil Code

The article to be applied mutatis mutandis described that certain juristic act have similar or equal effect of other articles, such as "A については X 条から Y 条までの規定を準用する ("A" shall apply mutatis mutandis to the case from Article X to Article Y)". In this case, we should retrieve the referred articles additionally because article to be mutatis mutandis and referred article are related. Thus, we reconstruct the article to be applied mutatis mutandis to use referred article, specifically in the case of "A については X 条から Y 条までの規定を準用する ("A" shall apply mutatis mutandis to the case from Article X to Article Y)", we reconstructed the text by adding text from Article X to Article Y and added "about A" in the end of the sentence, and we retrieved combination of these articles.

However, since such combination articles are longer than the original articles and there are several cases that important keywords in referred article are also used in the explanation of applied mutatis mutandis. In such a case, when the query is for finding out the referred articles only, the system tends to score those combination articles higher than the original one. In order to avoid such unintended situation, we introduce the constraints that query should have description related to the additional condition explained in the explanation of applied mutatis mutandis. In the system, When the combination articles are retrieved, the system check the existence of the legal terms that only exist in the explanation of applied mutatis mutandis are included in the query and exclude articles that do not have such terms from the results.

Actually, legal terminology can be divided into four groups as shown in Figure 2. In our system, information of the words in either of one of civil code and query(Group2 and Group3 in Figure 2 is not useful, however, it is useful to replace the other legal terms(Group1 in Figure 2). Additionally, the terms of Group2 are abstract legal terms containing several meanings of legal

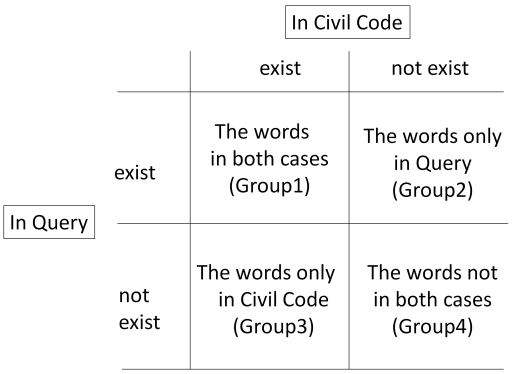


Fig. 2. Four groups of legal terminology

terms(Group1), such as ” 用益物権 (Real Rights)” contains ” 地上権 (Superficies)”,” 永小作権 (Emphyteusis)” and ” 地役権 (Servitudes)”. In our system, we collect the definition sentence from Google searching and replace the abstract legal terminology with top-two-ranked compound words by calculating TF-IDF shown in Figure 3.

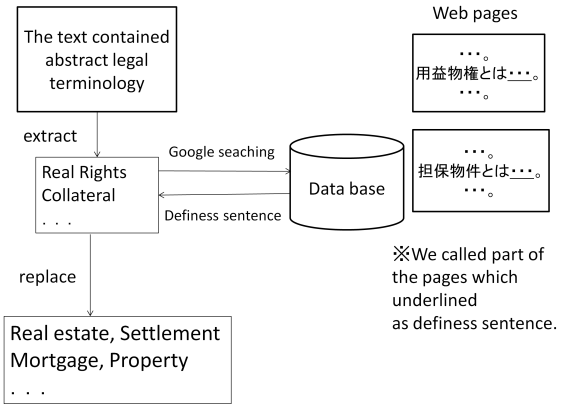


Fig. 3. To replace the abstract legal term

3 Experiment

3.1 Experimental Setup

The COLIEE Japanese Bar Exam task data was used to evaluate the proposed system. We construct articles database as follows.

1. Since name of chapter and section shows important keywords for the articles, we use those texts in addition to the main article text for the articles.
2. All texts are analyzed by using MeCab [1]for extracting index terms and construct three index term lists for each article; i.e., "compound words", "elems", and "words" index uses selected terms exist in legal terminology, selected terms that are used in legal terminology and terms selected by part of speech(POS) and stop words list respectively. In this system, words with following POS type noun (excluding pronoun, number, non-independent, suffix), verbs, adjectives, and adverbs and not included in the stop words list (e.g., general verb (する (do)、なる (become)) and general noun (こと (thing))).
3. CaboCha [6] is used for analyzing dependency structure for splitting conditional parts and applied mutatis mutandis parts.

Because the characteristics of the task data(from 2006 to 2013) are different every year, we evaluate the divided data in consideration of the characteristics of each year. The count of the query is shown in Table 1. The parameter values

Table 1. The count of the query

Year	'06	'07	'08	'09	'10	'11	'12	'13
Number of queries	36	42	45	56	47	41	79	66

used for the experiments were as follows.We used $k_1 = 1$, $k_3 = 1000$, $b = 1$, and $\beta = 2$, and if the query and article have the conditional part,we set the weight for $alltext$: 0.5, $conditional\,part$: 0.25, and $rest\,part$: 0.25, and we use top-one-ranked article as result. When the top-one-ranked article is a combination article of applied mutatis mutandis article and referred article, those two articles are returned as a result of the query. We show the summary of settings for test in Table 2 to evaluate in terms of precision and recall.

Table 2. Settings of tested data

	Article data	Index words list	penalty	BM25	weight	qke
0	single articles	words	words	on	off	off
1	single articles + combination articles	elems	elems		on	on

where, Article data column represents type of database. "0" corresponds to the database with single articles only. "1" uses combination article applied mutatis mutandis in addition to the single articles. Index words list column is the list to calculate similarity by BM25, and words is all words removed stop words and elems is morphemes of the legal terminology. Penalty column is the penalty value based on ABRIR[1], and for example, in case of elems, if the

[1] http://taku910.github.io/mecab/

elem in query does not exist in the article, we add the penalty to the article. Weight column express whether we calculate the similarity conditional part and rest part separately, if the value is "on", we calculate separately. Qke column express whether we replace the term in Group2(in Figure 2), if the value is "on", we replace the term in Group2 in query to the other legal term in Group1. Specifically, when we use the model that each article data included part name and chapter name, and index words list constructed by words, and penalty based on elems, and expansion for weight and term in Group2, we can express the model case:000011 based on Table 2.

3.2 Experimental Results

We show the results of the evaluation that effectiveness of calculating penalty, calculating weight of conditional part and rest part separately. Additionally, we evaluate our system from the viewpoint of precision, recall, and f-measure:

$$Precision = \frac{|\{relevant\ documents\}| \cap |\{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{4}$$

$$Recall = \frac{|\{relevant\ documents\}| \cap |\{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{5}$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6}$$

These are descriptions of the versions of the system tested for calculating penalty.

Penalty+Weight(case:001010). Calculate the penalty to elems, and weight to conditional part and rest part.

No penalty+Weight(case:00-010). Not calculate the penalty, and weight to conditional part and rest part.

Table 3. Precision, recall and f-measure

	case:001010			case:00-010		
year	precision	recall	f-measure	precision	recall	f-measure
2006	0.556	0.400	0.465	0.389	0.280	0.326
2007	0.619	0.510	0.559	0.595	0.490	0.538
2008	0.622	0.467	0.533	0.533	0.400	0.457
2009	0.571	0.478	0.520	0.589	0.493	0.537
2010	0.660	0.608	0.633	0.745	0.686	0.714
2011	0.707	0.558	0.624	0.585	0.462	0.516
2012	0.544	0.406	0.465	0.456	0.340	0.389
2013	0.682	0.549	0.608	0.621	0.500	0.554
All	0.617	0.489	0.546	0.563	0.447	0.498

In Table 3, we underlined the best case for each data set in terms of precision, recall and f-measure. From the result of all cases, we confirm using penalty is better than one without penalty. Query H19-5-3 " 占有者がその占有開始時に目的物について他人のものであることを知らず、かつ、そのことについて過失がなくても、その後、占有継続中に他人の物であることを知った場合には、悪意の占有者として時効が計算される。(When the possessor didn't know about the target that belongs to another person and without negligence and the possessor recognize the fact that target belongs to another person while occupying, Prescription pursuant as the possessor in bad faith.)" is a typical example that shows the effectiveness of using penalty. Followings are results of the top-one-ranked article for each case.

- [Penalty+Weight]Article 162.
- [No penalty+Weight]Article 197.

In case of No penalty+Weight, query and article 197 are similar in terms of words. However, in case of Penalty+Weight, Article 162 rose in ranking because article 162 contains the relata legal terms, such as " 時効 (Prescription)"," 所有者 (Possessor)".

However, difference between these two cases is not statistically significant by using the Wilcoxon signed rank test at a significance level of 0.05 for two-sided test. Because there are several queries in 2009 and 2010 where the system fails to select mandatory legal terms from the queries.

These are descriptions of the versions of the system tested for calculating weight.

Not Weight(case:001000). Calculate the penalty to elems, and not calculate the weight to conditional part and rest part.

Weight(case:001010). Calculate the penalty to elems, and calculate the weight to conditional part and rest part.

Table 4. Precision, recall and f-measure

	case:001000			case:001010		
year	precision	recall	f-measure	precision	recall	f-measure
2006	0.361	0.260	0.302	0.556	0.400	0.465
2007	0.548	0.451	0.495	0.619	0.510	0.559
2008	0.556	0.417	0.476	0.622	0.467	0.533
2009	0.482	0.403	0.439	0.571	0.478	0.520
2010	0.574	0.529	0.551	0.660	0.608	0.633
2011	0.561	0.442	0.489	0.707	0.558	0.624
2012	0.481	0.358	0.411	0.544	0.406	0.465
2013	0.576	0.463	0.514	0.682	0.549	0.608
All	0.519	0.412	0.460	0.617	0.489	0.546

The result in Table 4 that uses calculating weight is better than the result that does not use weight in all cases, and difference between these two cases is

statistically significant by using the Wilcoxon signed rank test at a significance level of 0.05 for two-sided test. In many cases, existence of conditional part have important role to calculate similarity, and we find that concrete example frequently appear in rest part, however, similar expression frequently appear in conditional part. These are descriptions of the versions of the system tested for appropriate settings of index word list and calculating penalty.

Index is words, penalty is elems(case:001010). Index word list is words, and calculate the penalty to elems.

Index is elems, penalty is words(case:010010). Index word list is elems, and calculate the penalty to words.

Table 5. Precision, recall and f-measure

	case:010010			case:001010		
year	precision	recall	f-measure	precision	recall	f-measure
2006	0.500	0.360	0.419	0.556	0.400	0.465
2007	0.643	0.529	0.581	0.619	0.510	0.559
2008	0.778	0.583	0.667	0.622	0.467	0.533
2009	0.625	0.522	0.569	0.571	0.478	0.520
2010	0.766	0.706	0.735	0.660	0.608	0.633
2011	0.805	0.635	0.710	0.707	0.558	0.624
2012	0.646	0.481	0.551	0.544	0.406	0.465
2013	0.742	0.598	0.662	0.682	0.549	0.608
All	0.689	0.547	0.610	0.617	0.489	0.546

In Table 5, contrary to our assumption, the setting that index word list is elems and calculate penalty to words is better than the setting that index word list is words and calculate penalty to elems, and difference between these two cases is statistically significant by using the Wilcoxon signed rank test at a significance level of 0.05 for two-sided test.. For example, about query H20-3-2 ” 成年被後見人が、後見人の同意を得ずに電気料金を支払った行為は、取り消すことができない。(When adult ward pay electric bill without the consent of the guardian of adult, the contract can not be canceled)”, we can obtain the top-1-article:

- [Index is words, calculate the penalty to elems]Article 864.
- [Index is elems, calculate the penalty to words]Article 9.

The correct is article 9 in this query, however, because the article 864 have the many legal terms(” 後見人 (guardian of adult)”, ” 未成年被後見人 (Minors ward)”, ” 元本 (principal)”), the effect of legal term is too strong to retrieve article 9. We find that it is important to consider about balance of legal term and other word(such as ” 取り消す (cancel)”), we can get good balanced result from the setting that index word is elems, calculate the penalty to words.

Next, we evaluate the junyou(article to be applied mutatis mutandis) and query keyword expansion(about the term in Group2(in Figure 2). These are descriptions of the versions of the system tested for these expansion.

Baseline BM25(case:010010). Index word list is elems, and calculate the penalty to words.

Baseline+Junyou(case:110010). Almost same setting with Baseline.Expansion for junyou(article to be applied mutatis mutandis) used.No expansion for query keyword(to the term in Group2(in Figure 2).

Baseline+Query Keyword Expansion(case:010011). Query Keyword expansion(to replace the term in Group2(Figure 2)) used.No expansion for junyou(article to be applied mutatis mutandis).

Baseline+Query Keyword Expansion+Junyou(case:110011). Query Keyword expansion and Junyou expansion used.

Table 6. Precision for each versions of the system

	case:010010	case:110010	case:010011	case:110011
2006	0.500	0.462	0.500	0.462
2007	0.643	0.609	0.643	0.609
2008	0.778	0.760	0.778	0.760
2009	0.625	0.596	0.643	0.614
2010	0.766	0.755	0.745	0.735
2011	0.805	0.773	0.805	0.773
2012	0.646	0.642	0.633	0.630
2013	0.742	0.718	0.758	0.718
All	0.689	0.668	0.689	0.666

In Table 6, we show the result about precision. As a result, query keyword expansion(to Group2 in 2) has slightly influence, positive influence in 2009,2013, negative influence in 2010,2012. For Wilcoxon Singed Rank test at a significance level of 0.05 for two-sided test, case:010010 and case:110011 is better than case:110010 and case:110011, and results are statistically significant.

Table 7. Recall for each versions of the system

	case:010010	case:110010	case:010011	case:110011
2006	0.360	0.360	0.360	0.360
2007	0.529	0.549	0.529	0.549
2008	0.583	0.633	0.583	0.633
2009	0.522	0.507	0.537	0.522
2010	0.706	0.725	0.686	0.706
2011	0.635	0.654	0.635	0.654
2012	0.481	0.491	0.472	0.481
2013	0.598	0.622	0.610	0.622
All	0.547	0.563	0.547	0.561

In Table 7, we show the result about recall. Case:110010 and case:110011 have almost same recall, and case:010010 and case:010011 have almost same recall.

For Wilcoxon Singed Rank test at a significance level of 0.05 for two-sided test, case:110010 and case:110011 is better than case:010010 and case:010011, and results are statistically significant.

Table 8. F-measure for each versions of the system

	case:010010	case:110010	case:010011	case:110011
2006	0.419	0.404	0.419	0.404
2007	0.581	0.577	0.581	0.577
2008	0.667	0.691	0.667	0.691
2009	0.569	0.548	0.585	0.565
2010	0.735	0.740	0.714	0.720
2011	0.710	0.708	0.710	0.708
2012	0.551	0.556	0.541	0.545
2013	0.662	0.667	0.676	0.667
All	0.610	0.611	0.610	0.609

In Table 8, it is difficult to judge which system is good with f-measure. In terms of the precision, the best case are case:010010 and case:010011, in terms of the recall, the best case are case:110010 and case:010011. Because expansion for query keyword expansion have slightly influence, these settings have the almost same value, and sometimes this expansion have positive effect and sometimes have negative effect. Junyou expansion have positive influence for recall, however, in terms of precision, no expansion setting is the best case. In these experiments, we can calculate similarity based on legal terminology and civil code article structure, and expansion for specific term and junyou. However, we cannot compare to infer the inclusion relation, such as ” マンション (Mansion)” and ” 不動産 (Real Estate)”, we still have room for like this improvement. In this experiment, we used the top-one-ranked article, however, in terms of recall, if we use the top-k-ranked data(such as top-two-ranked data, top-three-ranked data), it is likely that recall is improving.

In Table 9, we show that the result of the all cases in our experiment by using all data. The setting that index word list is elems, calculate the penalty to words,

Table 9. Precision, recall and f-measure

case	precision	recall	f-measure
00-000(baseline)	0.580	0.461	0.513
001010	0.617	0.489	0.546
00-010	0.563	0.447	0.498
001000	0.519	0.412	0.460
010010	**0.689**	0.547	0.610
110010	0.668	**0.563**	**0.611**
010011	**0.689**	0.547	0.610
110011	0.666	0.561	0.609

and weight to conditional part and rest part is the best case in terms of precision. The setting that index word list is elems, calculate the penalty to words, and weight to conditional part and rest part, and expand for junyou(article to be applied mutatis mutandis) is the best case in terms of recall and f-measure. Finally, we submitted the result that using index word list as elems, calculating penalty to words, and weight to conditional part and rest part, and four types of setting to expand for query keyword and junyou(article to be applied mutatis mutandis), actually, case010010, case110010, case010011, and case110011.

4 Conclusion and Future Works

In this paper, we propose an IR system that supports to find relevant articles of civil code for Query in Japanese Bar Exam based on legal terminologies and civil code article structure. From an evaluation experiment using the Japanese Bar Exam from 2006 to 2013, we confirm that our system can be used for retrieving relevant article in civil code. In many cases, our system results better than baseline system, and we can find that civil code article structure and legal terminology have important role in IR for Bar Exam. However, we did not adjust the parameter, such as penalty value and weight ratio, and we cannot capture the relation between concrete example and its attribute(such as mansion and real estate). Therefore, there are rooms for improvement for these problem.

References

1. Yoshioka, Masaharu: On a combination of probabilistic and boolean ir models for question answering. In: Information Retrieval Technology: 6th Asia Information Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 1-3, 2010. Proceedings, Berlin, Heidelberg, Springer Berlin Heidelberg (2010) 588–598
2. Mi-Young Kim, Randy Goebel, and Ken Satoh: COLIEE-2015 : Evaluation of legal question answering. In: Ninth International Workshop on Juris-informatics(JURISIN 2015). (2015)
3. S. E. Robertson, S. Walker: Okapi/keenbow at trec–8. In: Proceedings of TREC-8. (2000) 151–162
4. Nakagawa, Hiroshi and Mori, Tatsunori: Automatic term recognition based on statistics of compound nouns and their components. Terminology **9**(2) (2003) 201–219
5. Francis Bond, Timothy Baldwin, Richard Fothergill and Kiyotaka Uchimoto: Japanese semcor: A sense-tagged corpus of japanese. In: The 6th International Conference of the Global WordNet Association (GWC-2012), Matsue (2010)
6. Taku Kudo, Yuji Matsumoto: Japanese dependency analysis using cascaded chunking. In: CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops). (2002) 63–69

Author Index

ISBN 978-4-915905-74-2 C3004(JSAI)