Legal Knowledge and Information Systems

JURIX 2020: ***The Thirty-third Annual Conference***
Brno, Czech Republic, December 9-11, 2020

Editors:
Serena Villata
Jakub Harašta
Petr Křemen

JURIX 2020

Foundation for Legal Knowledge Systems
JURIX

Legal Knowledge and Information Systems

Foundation for Legal Knowledge Systems
JURIX

JURIX 2020:
The Thirty-third Annual Conference

Editors:
Serena Villata
Jakub Harašta
Petr Křemen

The field of legal knowledge and information systems has traditionally been concerned with the subjects of legal knowledge representation and engineering, computational models of legal reasoning, and the analysis of legal data, but recent years have also seen an increasing interest in the application of machine learning methods to ease and empower the everyday activities of legal experts.

This book presents the proceedings of the 33rd International Conference on Legal Knowledge and Information Systems (JURIX 2020), organised this year as a virtual event on 9–11 December 2020 due to restrictions resulting from the Covid-19 pandemic. For more than three decades, the annual JURIX international conference, which now also includes demo papers, has provided a platform for academics and practitioners to exchange knowledge about theoretical research and applications in concrete legal use cases. A total of 85 submissions by 255 authors from 28 countries were received for the conference, and after a rigorous review process, 20 were selected for publication as full papers, 14 as short papers, and 5 as demo papers. This selection process resulted in a total acceptance rate of 40% (full and short papers) and a competitive 23.5% acceptance rate for full papers. Topics span from computational models of legal argumentation, case-based reasoning, legal ontologies, smart contracts, privacy management and evidential reasoning to information extraction from different types of text in legal documents, and ethical dilemmas.

Providing a state-of-the-art overview of developments in the field, this book will be of interest to all those working with legal knowledge and information systems.

JURIX 2020

ISBN 978-1-64368-150-4

9 781643 681504

ISBN 978-1-64368-150-4 (print)
ISBN 978-1-64368-151-1 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

LEGAL KNOWLEDGE AND INFORMATION SYSTEMS

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 334

Recently published in this series

Vol. 333. M. Tropmann-Frick, B. Thalheim, H. Jaakkola, Y. Kiyokir and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXXII

Vol. 332. A.J. Tallón-Ballesteros and C.-H. Chen (Eds.), Machine Learning and Artificial Intelligence – Proceedings of MLIS 2020

Vol. 331. A.J. Tallón-Ballesteros (Ed.), Fuzzy Systems and Data Mining VI – Proceedings of FSDM 2020

Vol. 330. B. Brodaric and F. Neuhaus (Eds.), Formal Ontology in Information Systems – Proceedings of the 11th International Conference (FOIS 2020)

Vol. 329. A.J. Tallón-Ballesteros (Eds.), Modern Management based on Big Data I – Proceedings of MMBD 2020

Vol. 328. A. Utka, J. Vaičenonienė, J. Kovalevskaitė and D. Kalinauskaitė (Eds.), Human Language Technologies – The Baltic Perspective – Proceedings of the Ninth International Conference Baltic HLT 2020

Vol. 327. H. Fujita, A. Selamat and S. Omatu (Eds.), Knowledge Innovation Through Intelligent Software Methodologies, Tools and Techniques – Proceedings of the 19th International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_20)

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Legal Knowledge and Information Systems

JURIX 2020: The Thirty-third Annual Conference,
Brno, Czech Republic, December 9–11, 2020

Edited by

Serena Villata
Université Côte d'Azur, CNRS, Inria, I3S, France

Jakub Harašta
Masaryk University, Brno, Czechia

and

Petr Křemen
Czech Technical University, Prague, Czechia

IOS
Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-150-4 (print)
ISBN 978-1-64368-151-1 (online)
doi: 10.3233/FAIA334

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

We are delighted to present the proceedings volume of the 33rd International Conference on Legal Knowledge and Information Systems (JURIX 2020). For more than three decades, JURIX has organized an annual international conference for academics and practitioners, recently also including demos. The intention is to create a virtuous exchange of knowledge between theoretical research and applications in concrete legal use cases. Traditionally, this field has been concerned with legal knowledge representation and engineering, computational models of legal reasoning, and analyses of legal data. However, recent years have witnessed an increasing interest in the application of machine learning tools to relevant tasks to ease and empower legal experts everyday activities. JURIX is also a community where different skills work together to advance research by way of cross-fertilisation between law and computing technologies.

The JURIX conferences have been held under the auspices of the Dutch Foundation for Legal Knowledge Based Systems (www.jurix.nl). It has been hosted in a variety of European locations, extending the borders of its action and becoming an international conference in virtue of the the various nationalities of its participants and attendees.

The 2020 edition of JURIX, which runs from December 9 to 11, is co-hosted by the Institute of Law and Technology (Faculty of Law, Masaryk University, Brno) and the Knowledge-based Software Systems Group (Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague). Due to the Covid-19 health crisis, the conference is organised in a virtual format.

For this edition we have received 85 submissions by 255 authors from 28 countries; 20 of these submissions were selected for publication as full papers (ten pages each), 14 as short papers (four pages each) for a total of 34 presentations. In addition, 5 submissions were selected for publication as demo papers (four pages each). We were inclusive in making our selection, but the competition stiff and the submissions were put through a rigorous review process with a total acceptance rate (full and short papers) of 40%, and a competitive 23.5% acceptance rate for full papers. Borderline submissions, including those that received widely divergent marks, were accepted as short papers or demo papers only. The accepted papers cover a broad array of topics, from computational models of legal argumentation, case-based reasoning, legal ontologies, smart contracts, privacy management and evidential reasoning, through information extraction from different types of text in legal documents, to ethical dilemmas.

Two invited speakers have honored JURIX 2020 by kindly agreeing to deliver two keynote lectures: Katie Atkinson and Raja Chatila. Katie Atkinson is full professor of Computer Science and the Dean of the School of Electrical Engineering, Electronics and Computer Science at the University of Liverpool. She has also been the President of the International Association for Artificial Intelligence and Law in 2016–2017. She is one of the most significant representatives of the computational argumentation research community, and of AI and Law, where she focused on case-based reasoning and implementation of models of this in real world applications. Raja Chatila is Professor emeritus at Sorbonne Université. He is the former Director of the Institute of Intelligent Systems and Robotics (ISIR) and of the Laboratory of Excellence “SMART” on human-machine interaction. He is co-chair of the Responsible AI Working group in the Global Patnership on AI (GPAI), and he was member of the High Level Expert Group in AI with the European Commission (HLEG-AI). He is one of the main research scientists studying the ethical issues around Artificial Intelligence applications. We are very grateful to them for having accepted our invitation and for their interesting and inspiring talks.

Traditionally, the main JURIX conference is accompanied by co-located events comprising workshops and tutorials. This year's edition welcomes five workshops: EXplainable & Responsible AI in Law (XAILA 2020), Artificial Intelligence and Patent Data, Artificial Intelligence in JUrisdictional Logistics (JULIA 2020), the Fourth Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts (ASAIL 2020), and the Workshop on Artificial Intelligence and the Complexity of Legal Systems (AICOL 2020). One tutorial, titled *Defeasible Logic for Legal Reasoning*, is also planned in this edition of JURIX. The continuation of well-established events and the organization of entirely new ones provide a great added value to the JURIX conference, enhancing its thematic and methodological diversity and attracting members of the broader community. Since 2013, JURIX has also hosted the Doctoral Consortium, now in its eighth edition. This initiative aims to attract and promote Ph.D. researchers in the area of AI & Law so as to enrich the community with original and fresh contributions.

Organizing this edition of the conference would not have been possible without the support of many people and institutions. Special thanks are due to the local organizing team chaired by Jakub Harašta and Petr Křemen. We would like to thank the workshops' and tutorials' organizers for their excellent proposals and for the effort involved in organizing the events. We owe our gratitude to Monica Palmirani, who kindly assumed the function of the Doctoral Consortium Chair.

This year, we are particularly grateful to the 74 members of the Program Committee for their excellent work in the rigorous review process and for their participation in the discussions concerning borderline papers. Their work has been even more appreciated provided the complex situation we are experiencing due to the pandemic. Finally, we would like to thank the former and current JURIX executive committee and steering committee members not only for their support and advice but also generally for taking care of all the JURIX initiatives.

Last but not least, this year's conference was supported by AK Janoušek, law firm based in Prague, Czechia (www.janousekadvokat.cz) and by Artificial Intelligence Center ARIC based in Hamburg, Germany (www.aric-hamburg.de).

Serena Villata, JURIX 2020 Program Chair
Jakub Harašta, JURIX 2020 Organization Co-Chair
Petr Křemen, JURIX 2020 Organization Co-Chair

Sponsors

This page intentionally left blank

Contents

Full Papers

Short Papers

Demo Papers

Full Papers

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200844

Traffic Rules Encoding Using Defeasible Deontic Logic

Hanif Bhuiyan [a,b], Guido Governatori [a,1], Andy Bond [b], Sebastien Demmel [b], Mohammad Badiul Islam [a], Andry Rakotonirainy [b]
[a] *Data61, CSIRO*
[b] *Queensland University of Technology (QUT), Centre for Accident Research and Road Safety (CARRS-Q), Queensland, Australia*

Abstract. Automatically assessing driving behaviour against traffic rules is a challenging task for improving the safety of Automated Vehicles (AVs). There are no AV specific traffic rules against which AV behaviour can be assessed. Moreover current traffic rules can be imprecisely expressed and are sometimes conflicting making it hard to validate AV driving behaviour. Therefore, in this paper, we propose a Defeasible Deontic Logic (DDL) based driving behaviour assessment methodology for AVs. DDL is used to effectively handle rule exceptions and resolve conflicts in rule norms. A data-driven experiment is conducted to prove the effectiveness of the proposed methodology.

Keywords. Automated Vehicle, Traffic Rules, Defeasible Deontic Logic, Assessment.

1. Introduction

Automated Vehicles (AVs) are one of the most remarkable and highly anticipated technological developments of this century. This technology where AVs are programmed to drive according to traffic rules [1] can be seen as a solution to improve road safety and prevent traffic violation [2]. Thus one of the challenges is how to assess AV behaviour with respect to traffic rules.

The main problem is that, currently, there is no separate and comprehensive regulatory framework for AVs [3]; thus there is no specific (traffic) regulation to specifically assess the AVs behaviour. Although researchers have speculated that the current regulatory framework may handle AVs in existing transport system situations, it remains unclear whether all existing traffic rules are (directly) applicable to AVs. Leens and Lucivero mentioned that the current traffic rule model might be incomplete for the AV for some driving scenarios [1]. For example, in the current Queensland traffic rules[2], there are some vague expressions (e.g., “can safely overtake”, “overtake when there is a clear view”, etc.), which are almost impossible for an AV to follow [4] without additional parameters clarifying the meaning for the context and environment in which an AV is situated. Also, it may not be possible for AVs to properly follow rules which are related to conflicting situations [5] and exceptions.

[1] * Corresponding Author: Guido Governatori, Data61, CSIRO, Brisbane, Australia; E-mail:Guido.Governatori@data61.csiro.au

[2] https://www.legislation.qld.gov.au/view/html/inforce/current/sl-2009-0194

Therefore there is the need to develop a methodology to assess the AV behaviour by bridging the gap between traffic rules and AV knowledge processing. In this paper, we propose such a methodology by first encoding traffic rules in a machine-computable (MC) format that can be used to address the above-mentioned issues to assess AV driving behaviour.

Traffic rules include thousands of provisions and complex norms. This makes the encoding task challenging. Therefore, in this research, we use Defeasible Deontic Logic (DDL) to encode traffic rules. DDL is the combination of defeasible logic and deontic logic. DDL has been successfully used in legal reasoning to handle norms and exceptions, and it does not suffer from problems affecting other logics used for reasoning about compliance and norms [6]. DDL is an effective logical approach to solve the conflicting situation in norms as it works based on defeasible logic using a suitable variant.

In this paper, the discussion on the methodology for assessing AV driving behaviour is based on Queensland overtaking traffic rules[3]. We choose overtaking traffic rules as it is one of the most challenging traffic rules which has several complicated conditions with multiple facets.

2. Related Work

In general, traffic rules are expressed in natural language and are created for human drivers. Traffic rules are often very detailed and complex and, therefore, it is a big challenge to encode them. Other research has addressed the challenges of traffic rule encoding for different purposes such as driving assistance systems [7], driving context modelling [8], traffic situation representation [9], etc. Some significant related research work about traffic rules encoding for assessing AV behaviour are given below.

In [4], Isabelle logic theorem is proposed to encode traffic rules to monitor the AV behaviour. This research aims to use monitoring to ensure that AV obeys traffic rules. To do that, traffic rules are codified into Linear Temporal Logic (LTL) using High Order Logic (HOL). A verified checker is used to check the compliance of the AV behaviour with the encoded traffic rules. To analyze the data, the recorded information is modelled as discrete-time runs.

In [10], an expert system to encode traffic rules for controlling the autonomous vehicle in certain situations is proposed. This expert system consists of data processing algorithms, multidimensional databases, and a cognitive model of traffic objects and their relationships. To encode traffic rules, data are grouped into two sets. One set consists of traffic lights, road markings, road signs, road types, etc. Another dataset consists of around 800 traffic rules.

In [11], an encoding method for traffic rules was proposed to keep the autonomous vehicle accountable. Three major steps consolidate this methodology. First, legal analysis alleviates the implicit redundancy from the legal text. Next, it explicitly sorts out the responsibility of the AV and the user and then breaks the rules into logical predicate precursors. One of the major aims of this work is to give the opportunity to further develop in the expressivity of rules (translated traffic rules) by using Higher Order Language (HOL).

In [12], a system, Mivar, is introduced that can monitor vehicle activities in real-time and can also inform the driver about the violations of traffic rules. The Mivar system

[3] https://www.legislation.qld.gov.au/view/html/inforce/current/sl-2009-0194#pt.11-div.3

consists of three main modules: trajectory control system (lane position, a safe distance from other vehicles, etc.), a simplified technical vision system (road situation in real-time), and a decision support system (DSS).

Although a few studies work on monitoring mechanisms on the AV activities to verify the AV behaviour against traffic rules [12,4]. However, none of them solve the issues of handling exceptions and resolving conflicting situation of traffic rules. However, these are important variant features and can create challenges while assessing the AV behaviour against traffic rules. In comparison to both of these works and other above-mentioned works, we have proposed a DDL based methodology that can validate the AV behaviour against traffic rules more effectively by efficiently handling the rule exceptions and resolving conflicts in the traffic rules.

3. Driving Behaviour Assessment

The flow diagram of driving behaviour assessment methodology is shown in Figure 1. The proposed methodology consists of three modules. In the first module, traffic rules are encoded into a machine computable (MC) format. In the second module, AV information is formulated into the MC format to comply with the encoded traffic rules. Finally, in module three the mapping and reasoning of traffic rules and AV information are combined to assess the AV behaviour. A brief description of each module is given below.

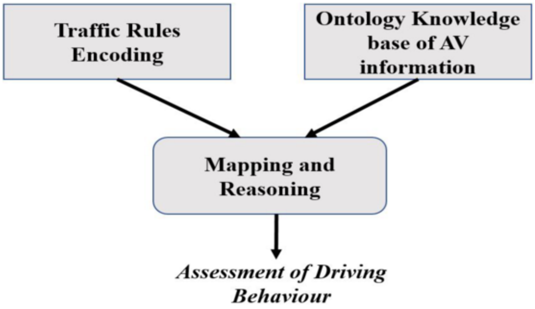

Traffic Rules Encoding
Ontology Knowledge base of AV information
Mapping and Reasoning
Assessment of Driving Behaviour

Figure 1. ***Flow diagram of driving behaviour assessment methodology.***

3.1. Traffic Rules Encoding

Defeasible Deontic Logic (DDL) is used as a formal foundation of this encoding methodology [13]. The proposed methodology works in four steps, as shown in Figure 2, which are define atoms, identify norms, generate if-then structure, and rules encoding.

In the first step, atoms are defined based on the terms appearing in the traffic rules. An atom corresponds to a statement (combining terms in the traffic rules) that can be evaluated as true or false. A term is a variable or an individual constant in the sentence. The proposed encoding method considers these variables and constants in the rule sentences. Norms are identified in the second step. In the traffic rule, norms are conditions to perform specific actions. Every norm is represented by one or more rules, which could either be constitutive or prescriptive rules. Both constitutive and prescriptive forms of rules are used to identify norms. In the third step, if-then structures are generated from rules using atoms and norms. This structure comprises two parts: if (antecedent or premise) and then (consequent or conclusion). If the premise becomes true, then the consequent

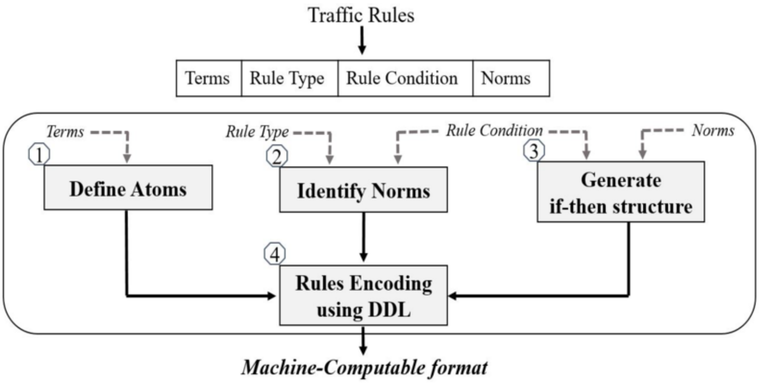
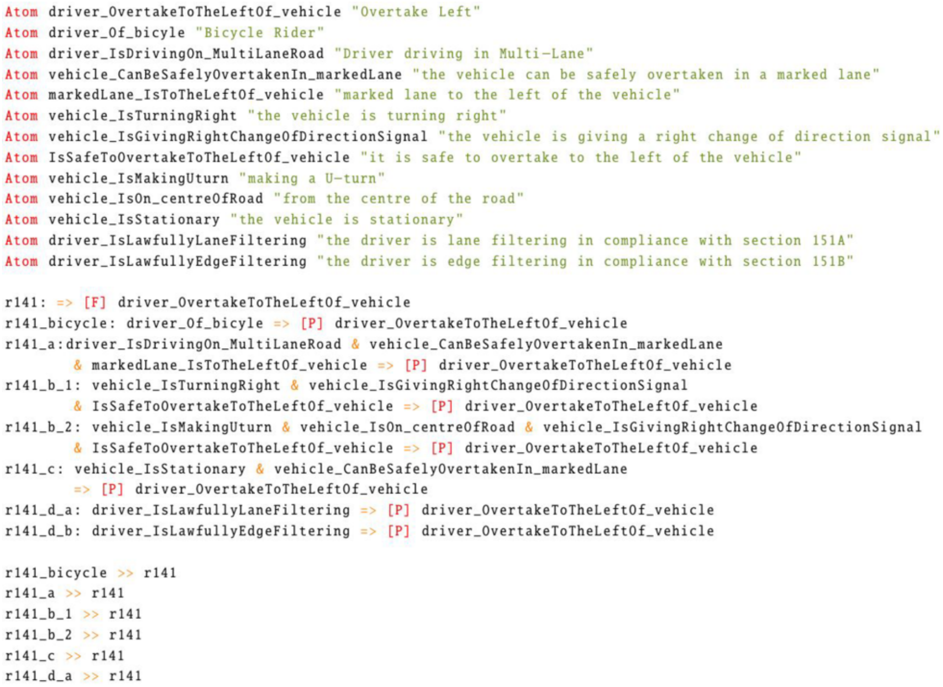


Figure 2. ***Traffic rules encoding.***

part of the rules is triggered. In the fourth step, rules are encoded into the MC format. After identifying and combining atoms, norms, and if-then structures, DDL is applied to them to create the MC format of the rule. The normative effects of (prescriptive) rules are modelled by Obligation (O), Prohibition (F), and Permission (P).

We now provide (Figure 3) an example of traffic rules encoding using DDL. For this example, we use Queensland Overtaking Traffic Rules 141[4]. In the bottom of Figure 3, the priority between the encoded rule is shown.



```
Atom driver_OvertakeToTheLeftOf_vehicle "Overtake Left"
Atom driver_Of_bicyle "Bicycle Rider"
Atom driver_IsDrivingOn_MultiLaneRoad "Driver driving in Multi-Lane"
Atom vehicle_CanBeSafelyOvertakenIn_markedLane "the vehicle can be safely overtaken in a marked lane"
Atom markedLane_IsToTheLeftOf_vehicle "marked lane to the left of the vehicle"
Atom vehicle_IsTurningRight "the vehicle is turning right"
Atom vehicle_IsGivingRightChangeOfDirectionSignal "the vehicle is giving a right change of direction signal"
Atom IsSafeToOvertakeToTheLeftOf_vehicle "it is safe to overtake to the left of the vehicle"
Atom vehicle_IsMakingUturn "making a U-turn"
Atom vehicle_IsOn_centreOfRoad "from the centre of the road"
Atom vehicle_IsStationary "the vehicle is stationary"
Atom driver_IsLawfullyLaneFiltering "the driver is lane filtering in compliance with section 151A"
Atom driver_IsLawfullyEdgeFiltering "the driver is edge filtering in compliance with section 151B"

r141: => [F] driver_OvertakeToTheLeftOf_vehicle
r141_bicycle: driver_Of_bicyle => [P] driver_OvertakeToTheLeftOf_vehicle
r141_a:driver_IsDrivingOn_MultiLaneRoad & vehicle_CanBeSafelyOvertakenIn_markedLane
        & markedLane_IsToTheLeftOf_vehicle => [P] driver_OvertakeToTheLeftOf_vehicle
r141_b_1: vehicle_IsTurningRight & vehicle_IsGivingRightChangeOfDirectionSignal
        & IsSafeToOvertakeToTheLeftOf_vehicle => [P] driver_OvertakeToTheLeftOf_vehicle
r141_b_2: vehicle_IsMakingUturn & vehicle_IsOn_centreOfRoad & vehicle_IsGivingRightChangeOfDirectionSignal
        & IsSafeToOvertakeToTheLeftOf_vehicle => [P] driver_OvertakeToTheLeftOf_vehicle
r141_c: vehicle_IsStationary & vehicle_CanBeSafelyOvertakenIn_markedLane
        => [P] driver_OvertakeToTheLeftOf_vehicle
r141_d_a: driver_IsLawfullyLaneFiltering => [P] driver_OvertakeToTheLeftOf_vehicle
r141_d_b: driver_IsLawfullyEdgeFiltering => [P] driver_OvertakeToTheLeftOf_vehicle

r141_bicycle >> r141
r141_a >> r141
r141_b_1 >> r141
r141_b_2 >> r141
r141_c >> r141
r141_d_a >> r141
```

Figure 3. ***Encoding of Queensland Overtaking Trafic Rule 141***

[4] https://www.legislation.qld.gov.au/view/html/inforce/current/sl-2009-0194#sec.141

3.2. Ontology Knowledge Base

Ontology is a way of representing knowledge in a structured framework that consists of concepts (classes) and relationships (properties). It allows communication and information sharing between software and hardware agents by facilitating the design of rigorous and exhaustive conceptual schema. An important characteristic of ontology is that it represents knowledge in a machine-computable (MC) format as RDF (Resource Description Framework) data [14]. RDF[5] provides a conceptual statement to give a clear specification for modelling data. This MC knowledge (RDF) representation can bridge the gap between AV perception and knowledge processing. Therefore, in this work, we create ontologies of AV information. Moreover, it is also proved by [15] that an ontology can effectively represent road information and driving behaviour of the vehicle, which is helpful for AV knowledge processing. Here, the MC knowledge base is used by the encoded traffic rules to provide the input for the reasoning engine about what are the legal requirements for the AV in the particular situation identified by the data available to the AV.

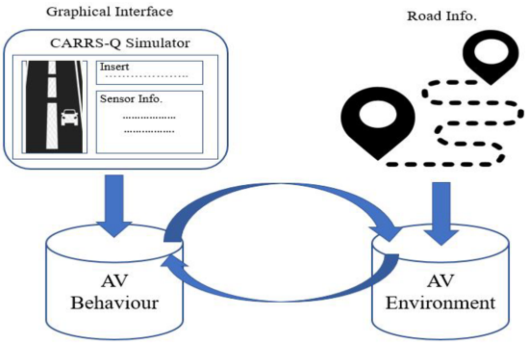


Figure 4. ***Structure of Knowledge Base.***

The structure of the knowledge base is shown in Figure 4. Protégé[6] is used to build these ontologies. The knowledge base consists of two ontologies: AV behaviour and AV environment ontology. AV behaviour ontology is created by using the behaviour information (i.e speed, direction, lane number, etc.) of the AV. The environment ontology is created by using road information (i.e road marking, road type, etc.) and information about AV surroundings (i.e other vehicles speeds, other vehicles lane numbers, etc.). We collect all this information from the CARRS-Q advanced driving simulator[7]. Moreover, based on the requirements, these ontologies can be reused and easily extended by adding another concept. To design the road in the simulator, we collect road information (Queensland, Australia) from Wikipedia and other web blogs[8].

3.3. Reasoning

This section will introduce the reasoning engine to make the assessment of the AV driving behaviour against traffic rules. Figure 5 shows the work flow diagram of the reasoning

[5] https://www.w3.org/RDF/

[6] https://protege.stanford.edu/

[7] https://research.qut.edu.au/carrsq/services/advanced-driving-simulator/

[8] https://www.ozroads.com.au/QLD/classifications.htm

engine. The input to this reasoning engine are atoms (from encoded traffic rules), encoded traffic rules, and knowledge base. The proposed reasoning engine works in four steps. Brief descriptions of these four steps are given below.

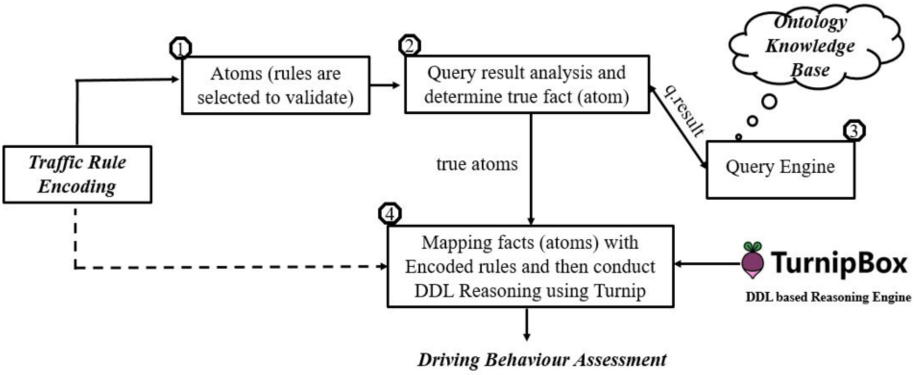

1
Atoms (rules are selected to validate)
2
Query result analysis and determine true fact (atom)
Ontology Knowledge Base
q.result
3
Query Engine
Traffic Rule Encoding
true atoms
4
Mapping facts (atoms) with Encoded rules and then conduct DDL Reasoning using Turnip
TurnipBox
DDL based Reasoning Engine
Driving Behaviour Assessment

Figure 5. ***Work flow diagram of the Reasoning Engine.***

3.3.1. Atoms:

The generated atoms of corresponding traffic rules are stored in this step for further processing.

3.3.2. Determine True Fact

This step determines true facts (atoms) for the driving action of the AV. In this step, for each query, we set some predefined answers. The query result is compared with those answers and if it matches then the system identifies that it is a true fact. For example, to verify the atom (*driver_Of_bicyle*), the SPARQL Query 1_1 is triggered. The answer of the query shows that it is AV & Automated_Vehicle. Therefore, it can be concluded that, this atom is not true as the atom is about a bicycle.

```
Atom driver_Of_bicyle.
Query 1_1: What type of vehicle it is?
```

```
prefix ab:<http://www.semanticweb.org/bhuiyanh/
ontologies/2019/8/untitled-ontology-50#>
SELECT ?Vehicle ?Type
WHERE   {
ab:time_1  ab:driving ?Vehicle.
?Vehicle ab:is_a ?Type.
}
Query_Result:Automated_Vehicle
```

3.3.3. Query Engine

The query engine contains predefined SPARQL queries for each atom. These queries are made based on the empirical study of the overtaking traffic rules of Queensland. Based on the atom, the number of queries vary. SPARQL is one of the most powerful and effective query languages to access the ontology-based knowledge base. Here, we use SPARQL queries to retrieve AV behaviour and environment information from the knowledge base. An algorithm is designed to trigger these queries. If the query result is NULL, then the process breaks and uses the next query. An example of an atom (*driver_Of_bicyle*) and its corresponding query and its results is shown above.

3.3.4. Mapping and Reasoning in Turnip

Turnip[9] is a Defeasible Deontic Logic-based reasoning tool. It is a tool which accepts facts (atoms), strict rules, defeasible rules, defeaters, superiority relation, and modality of DL. It supports non-monotonic and monotonic reasoning with incomplete and inconsistent information. A full illustration of Turnip is out of the scope of this paper. In this research, Turnip receives the encoded rules and atoms and thus does the mapping and reasoning.

For example (see Table 1), regarding overtaking traffic rule 141 (Figure 3), if for any timestamp, true facts for the AV are as Table 1(a), then the reasoning result shows that, AV has permission ($[P]$) to do left-side overtaking. However, if any of the facts among them (Table 1(a)) become false like (Table 1(b)), then permisssion for left overtaking is declined ($[F]$) according to traffic rule 141.

Table 1. ***Example of mapping and reasoning in Turnip***

Rules	
Encoding of Rule 141 (Figure 3)	
True Facts	**True Facts**
driver_IsDrivingOn_MultiLaneRoad vehicle_CanBeSafelyOvertakenIn_markedLane markedLane_IsToTheLeftOf_vehicle IsSafeToOvertakeToTheLeftOf_vehicle vehicle_IsOn_centreOfRoad	driver_IsDrivingOn_MultiLaneRoad vehicle_CanBeSafelyOvertakenIn_markedLane IsSafeToOvertakeToTheLeftOf_vehicle vehicle_IsOn_centreOfRoad
Results	**Results**
[P] driver_OvertakeToThe LeftOf_vehicle	[F] driver_OvertakeToThe LeftOf_vehicle
(a)	(b)

4. Experiment

This chapter shows the experiment results of the proposed Automated Vehicle (AV) driving behaviour assessment approach. We firstly present the experiment scenarios and data. Each scenario is a specific maneuver of the AV. The experiment is conducted to find the legal and illegal driving behaviour of the AV during the maneuver. The evaluation is performed with the help of domain experts.

4.1. Experiment Scenarios

The CARRS-Q advanced driving simulator is used to make experiment scenarios. We do some empirical study on overtaking cases of Queensland traffic and hence composed scenarios. This study helps us to cover (see Figure 6) almost all aspects of overtaking cases generally occurring in Queensland. Four scenarios are designed to investigate the proposed approach. A depiction of each scenario is shown in Figure 6.

[9] https://turnipbox.netlify.com/

- In Figure 6(a), the AV is approaching to overtake the TV-1 in a multi-lane road.
- AV is approaching to overtake TV-2 although it is displaying a “do not overtake turning vehicle” sign (Figure 6(b)).
- In Figure 6(c), the AV is approaching to overtake TV-1 as it is in a stationary position.
- In a non-marked two-way road, the AV is approaching to overtake TV-1 (Figure 6(d)).

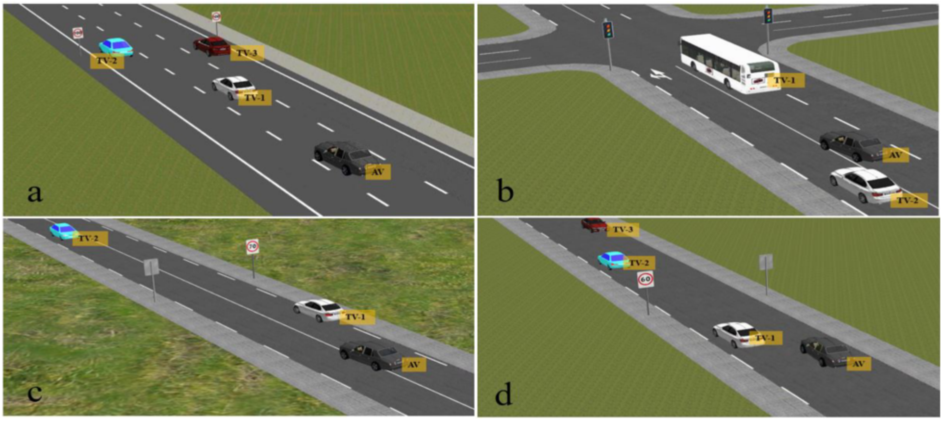


Figure 6. ***Experiment Scenarios.***

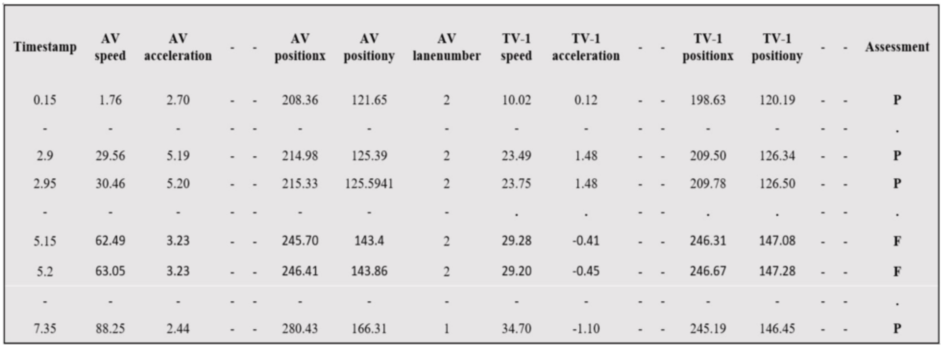
These types of overtaking cases are very common in Queensland traffic. In some aspects, these types of maneuver are risky and challenging. We experiment on these four scenarios for both Left Overtaking (LO) and Right Overtaking (RO). Based on overtaking type (LO / RO), the scenario changes. For each experiment, we consider three different maneuvers to evaluate the proposed methodology effectiveness. Among these maneuvers, two of them are a clear case of legal and illegal action. The third maneuver is about the border-line maneuver, which cannot directly define whether it is legal or illegal.

4.2. Experiment Data

Experiment data is generated using the CARRS-Q simulator. The simulator can provide the data under managed and repeatable conditions and also make the data more useful and meaningful for analysing. A snippet of experiment data is shown in Figure 7. Here, we generate behaviour and environment information of vehicles every 0.05s.

4.3. Experiment Result

We conducted 24 experiments based on the above-represented scenarios (Figure 6). 12 experiments were conducted individually for Left Overtaking (LO) and Right Overtaking (RO). Each experiment is divided into n timestamps. Each timestamp is 0.05s (Figure 7). In each experiment, every timestamp is validated against the corresponding traffic rule. After completing the validation of all timestamps of an experiment, the result is determined. For example, experiment result of all timestamps of the LO, experiment 2, maneuver type -3 is shown in Figure 7. As in this maneuver, in some timestamps the driving action is prohibited (Prohibition: F), therefore this maneuver is illegal according



Timestamp	AV speed	AV acceleration	-	-	AV positionx	AV positiony	AV lanenumber	TV-1 speed	TV-1 acceleration	-	-	TV-1 positionx	TV-1 positiony	-	-	Assessment
0.15	1.76	2.70	-	-	208.36	121.65	2	10.02	0.12	-	-	198.63	120.19	-	-	P
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	.
2.9	29.56	5.19	-	-	214.98	125.39	2	23.49	1.48	-	-	209.50	126.34	-	-	P
2.95	30.46	5.20	-	-	215.33	125.5941	2	23.75	1.48	-	-	209.78	126.50	-	-	P
-	-	-	-	-	-	-	-	.	.	-	-	.	.	-	-	.
5.15	62.49	3.23	-	-	245.70	143.4	2	29.28	-0.41	-	-	246.31	147.08	-	-	F
5.2	63.05	3.23	-	-	246.41	143.86	2	29.20	-0.45	-	-	246.67	147.28	-	-	F
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	.
7.35	88.25	2.44	-	-	280.43	166.31	1	34.70	-1.10	-	-	245.19	146.45	-	-	P

Figure 7. ***An snippet of experiment data and assessment result (LO, Ex -2, Maneuver Type-3)***.

to the LO-141 (QLD Traffic Rules). However, if all timestamps of this maneuver are permitted (Permission: *P*), then it would become a legal maneuver.

Table 2 shows the effectiveness of the proposed methodology in terms of assessing AV behaviour against overtaking traffic rules. To evaluate the experiment result, we took help from three domain experts (who have 25 years experience of driving in Queensland and never have any allegation of illegal overtaking). We use the knowledge of experienced drivers to validate the interpretation of local overtaking maneuvers. For the maneuver, we consider domain expert judgement as the ground truth. If the experts regard any behaviour as illegal then the result is considered negative.

According to the experiment result (Table 2), the proposed methodology successfully works for both LO and RO cases for the experiment 2. For experiment-3 & 4, the proposed method could correctly assess all LO cases, but is unsuccessful for all RO cases. On the

Table 2. ***Experiment Result of the proposed assessment method***.

Ex-No.	Situations Covered	Overtaking Type					
		Left Overtaking (LO)			Right Overtaking (RO)		
		Maneuver Type	Proposed Methodology	Domain Expert	Maneuver Type	Proposed Methodology	Domain Expert
Ex-1 .	Vehicles position, multiple vehicles, multiple lanes, lane type (marked lane), lane marking.	Type -1	✓	✓	Type -1	✓	✓
		Type -2	×	×	Type -2	×	×
		Type -3	×	✓	Type -3	×	×
Ex-2 .	Vehicles position, multiple vehicles, multiple lanes, lane type (marked lane), lane marking, do not overtake turning vehicle sign, Intersections.	Type -1	✓	✓	Type -1	✓	✓
		Type -2	×	×	Type -2	×	×
		Type -3	×	×	Type -3	×	×
Ex-3 .	Vehicles position multiple vehicles, stationary vehicle, two-way lane, lane type (marked lane), lane marking.	Type -1	✓	✓	Type -1	✓	✓
		Type -2	×	×	Type -2	×	×
		Type -3	×	✓	Type -3	×	✓
Ex-4.	Vehicles position multiple vehicles, multiple lanes, lane type (non-marked lane), two-way lane.	Type -1	✓	✓	Type -1	✓	✓
		Type -2	×	×	Type -2	×	×
		Type -3	×	×	Type -3	✓	✓

other side, for experiment-1, the proposed method is not successful to correctly assess all LO cases, while it is successful for all RO cases.

5. Conclusion and Future Work

The experiment result shows that the proposed assessment method can assess the AV driving behaviour against traffic rules by effectively handling exceptions and resolving conflicts in rule norms. Therefore, it can be said that, this assessment methodology would be useful for the traffic authority to automatically identify AVs that drive illegally.

In future, we will enhance the scope of this proposed assessment mechanism by covering other traffic environments such as lane change, roundabout, intersection crossing, and etc. Furthermore, from this assessment mechanism we will determine which traffic rules need additional interpretation in terms of the information available by an AV.

References

[1] Leenes R, Lucivero F. Laws on robots, laws by robots, laws in robots: regulating robot behaviour by design. Law, Innovation and Technology. 2014 Dec 31;6(2):193-220.

[2] Khorasani G, Tatari A, Yadollahi A, Rahimi M. Evaluation of intelligent transport system in road safety. International Journal of Chemical, Environmental & Biological Sciences (IJCEBS). 2013;1(1):110-8.

[3] Fulbright NR. Autonomous vehicles: The legal landscape of dedicated short range communication in the US, UK and Germany. Accessed: Dec. 2017;11:2018.

[4] Rizaldi A, Keinholz J, Huber M, Feldle J, Immler F, Althoff M, Hilgendorf E, Nipkow T. Formalising and monitoring traffic rules for autonomous vehicles in Isabelle/HOL. In International Conference on Integrated Formal Methods 2017 Sep 20 (pp. 50-66). Springer, Cham.

[5] Prakken H. On the problem of making autonomous vehicles conform to traffic law. Artificial Intelligence and Law. 2017 Sep 1;25(3):341-63.

[6] Governatori G. The Regorous approach to process compliance. In2015 IEEE 19th International Enterprise Distributed Object Computing Workshop 2015 Sep 21 (pp. 33-40). IEEE.

[7] Zhao L, Ichise R, Liu Z, Mita S, Sasaki Y. Ontology-based driving decision making: A feasibility study at uncontrolled intersections. IEICE TRANSACTIONS on Information and Systems. 2017 Jul 1;100(7):1425-39.

[8] Xiong Z, Dixit VV, Waller ST. The development of an Ontology for driving Context Modelling and reasoning. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) 2016 Nov 1 (pp. 13-18). IEEE.

[9] Buechel M, Hinz G, Ruehl F, Schroth H, Gyoeri C, Knoll A. Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles. In 2017 IEEE Intelligent Vehicles Symposium (IV) 2017 Jun 11 (pp. 1471-1476). IEEE.

[10] Shadrin SS, Varlamov OO, Ivanov AM. Experimental autonomous road vehicle with logical artificial intelligence. Journal of advanced transportation. 2017 Jan 1;2017.

[11] Costescu DM. Keeping the autonomous vehicles accountable: Legal and Logic Analysis on Traffic Code. In Conference Vision Zero for Sustainable Road Safety in Baltic Sea Region 2018 Dec 5 (pp. 21-33). Springer, Cham.

[12] Aladin DV, Varlamov OO, Chuvikov DA, Chernenkiy VM, Smelkova EA, Baldin AV. Logic-based artificial intelligence in systems for monitoring the enforcing traffic regulations. InIOP Conference Series: Materials Science and Engineering 2019 May (Vol. 534, No. 1, p. 012025). IOP Publishing.

[13] Bhuiyan H, Olivieri F, Governatori G, Badiul M Islam, Bond A, Rakotonirainy A. A Methodology for Encoding Regulatory Rules. In 2019 4th International Workshop on MIning and REasoning on Legal texts (MIREL) 2019 Dec 11 (pp. 1-13). CUER-WS.

[14] Najmi E, Malik Z, Hashmi K, Rezgui A. ConceptRDF: An RDF presentation of ConceptNet knowledge base. In 2016 7th International Conference on Information and Communication Systems (ICICS) 2016 Apr 5 (pp. 145-150). IEEE.

[15] Zhao L, Ichise R, Mita S, Sasaki Y. Core Ontologies for Safe Autonomous Driving. InInternational Semantic Web Conference (Posters & Demos) 2015.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200845

A Model for the Burden of Persuasion in Argumentation

Roberta CALEGARI [a,1] and Giovanni SARTOR [a,b]
[a] *CIRSFID - Alma AI, University of Bologna, Italy*
[b] *European University Institute, Florence, Italy*

Abstract. This work provides a formal model for the burden of persuasion in legal proceedings. The model shows how the allocation of the burden of persuasion may induce a satisfactory outcome in contexts in which the assessment of conflicting arguments would, without such an allocation, remain undecided. The proposed model is based on an argumentation setting in which arguments may be accepted or rejected according to whether the burden of persuasion falls on the conclusion of such arguments or on its complements. Our model merges two ideas that have emerged in the debate on the burden of persuasion: the idea that allocation of the burden of persuasion makes it possible to resolve conflicts between arguments, and the idea that its satisfaction depends on the dialectical statuses of the arguments involved. Our model also addresses cases in which the burden of persuasion is inverted, and cases in which burdens of persuasion are inferred through arguments.

Keywords. burden of persuasion, argumentation, legal reasoning

1. Introduction

The burden of proof is a central feature in legal decision-making and yet no agreed theory of it exists [1,2]. Generally speaking, we can say that burdens of proof distribute dialectical responsibilities to the parties: when a party has a burden of proof of type b relative to a claim ϕ, then, unless the party provides the kinds of arguments or evidence required by b, the party will lose on claim ϕ, i.e., that party will fail to establish ϕ. Burdens of proof can complement the analysis of dialectical frameworks that are provided by argumentation systems. In particular, they are important in adversarial contexts: they are meant to facilitate the process of reaching a single outcome in contexts of doubt and lack of information. In the legal domain, two types of burdens are distinguished: the *burden of production* (also called burden of providing evidence, or 'evidential' burden), and the *burden of persuasion*. The focus of this paper is on the burden of persuasion, and its purpose is to show how an allocation of the burden of persuasion may induce single outcomes in contexts in which the assessment of conflicting arguments would, without such an allocation, remain undecided. Our approach is based on providing specific criteria for accepting and rejecting propositions upon which there is a burden of persuasion.

[1] R. Calegari and G. Sartor have been supported by the H2020 ERC Project "CompuLaw" (G.A. 833647).

2. Burdens of production and burdens of persuasion

Following the account in [3], we distinguish the burden of production from the burden of persuasion. A party burdened with production needs to provide some support for the claim he or she is advancing. More exactly, we can say that the party has the burden of production for ϕ if the following is the case: unless relevant support for ϕ is provided – i.e., unless an argument for ϕ is presented that deserves to be taken into consideration – then ϕ will not be established (even in the absence of arguments against ϕ). When knowledge is represented through a set of rules and exceptions, the party interested in establishing the conclusion of a rule has the burden of production relative to the elements in the rule's antecedent condition, while the other party (who is interested in preventing the conclusion from being derived from the rule) has the burden of production relative to the exceptions to the rule (as provided in a separate exception clause or in an unless-exception within the rule). Note that meeting the burden of production for a claim ϕ is only a *necessary* condition, and not a sufficient one, for establishing ϕ, since the produced arguments may be defeated by counterarguments. This aspect is addressed by the burden of persuasion, under which the burdened party looking to establish a claim needs to provide a 'convincing' argument for it—that is, an argument that prevails over arguments to the contrary to an extent that is determined by the applicable standard of proof. If there is a burden of persuasion on a proposition ϕ, and no prevailing argument for ϕ is provided, then the party concerned will lose on ϕ. In this paper, we focus on the burden of persuasion. We shall discuss it by way of three running examples: one from criminal law, one from civil law, and one from antidiscrimination law.

In criminal law, the burden of production is distributed between prosecution and defence, while the burden of persuasion (in most legal systems) is always on prosecution. More exactly, in criminal law, the burden of production falls on the prosecution relative to the two constitutive elements of crime, namely, the criminal act (*actus reus*) and the required mental state (*mens rea*, be it intention/recklessness or negligence), while it falls to the defendant relative to justifications or exculpatory defences (e.g., self-defence, state of necessity, etc.). In other words, if both actus reus and mens rea are established, but no exculpatory evidence is provided, the decision should be a criminal conviction. On the other hand, the burden of persuasion falls on the prosecution for all determinants of criminal responsibility, including not only for the constitutive elements of a crime but also for the absence of an exculpatory defence.

Example 1 (Criminal law example) *Let us consider a case in which a woman has shot and killed an intruder in her own home. The applicable law consists of the rule according to which intentional killing constitutes murder, and in the exception according to which there is no murder if the victim was killed in self-defence. Assume that it has been established with certainty that the woman shot the intruder and that she did so intentionally. However, it remains uncertain whether the intruder was threatening the woman with a gun, as claimed by the defence, or had turned back and was running away on having been discovered, as claimed by the prosecution. The burden of persuasion is on prosecution, who needs to provide a convincing argument for murder. Since it remains uncertain whether there was self-defence, prosecution has failed to provide such an argument. Therefore the legally correct solution is that there should be no conviction: the woman needs to be acquitted.*

In civil law, both the burden of production and the burden of persuasion may be allocated in different ways in the law, depending on various factors, such as the ability of a party to provide evidence in favour of his or her claim. In matters of civil liability, for example, it is usually the case that the plaintiff, who asks for compensation, has to prove both that the defendant caused him harm, and that this was done intentionally or negligently. However, in certain cases, there is an inversion of the burden of proof for negligence (both the burden of production and the burden of persuasion). This means that in order to obtain compensation, the plaintiff only has to prove that he was harmed by the defendant. This will be sufficient to win the case unless the defendant provides a convincing argument that she was not negligent.

Example 2 (Civil law example) *Let us consider a case in which a doctor caused harm to a patient by misdiagnosing his case. There is no doubt that the doctor harmed the patient: she failed to diagnose a cancer, which consequently spread and became incurable. However, it is uncertain whether or not the doctor followed the guidelines governing this case: it is unclear whether she prescribed all the tests that were required by the guidelines in such a case, or whether she failed to prescribe some tests that would have enabled the cancer to be detected. Assume that, under the applicable law, doctors are liable for any harm suffered by their patients, but they can avoid liability if they show that they were diligent (not negligent) in treating the patient, i.e., that they exercised due care. Thus, doctors have both a burden of production and a burden of persuasion concerning their diligence. Let us assume that law also says that doctors are considered to be diligent if they followed the medical guidelines that govern the case. In this case, given that the doctor has the burden of persuasion on her diligence, and that she failed to provide a convincing argument for it, the legally correct solution is that she should be ordered to compensate the patient.*

These two examples share a common feature. In both, uncertainty remains concerning a decisive issue, namely, the existence of self-defence in the first example and the doctor's diligence in the second. However, this uncertainty does not preclude the law from prescribing a single legal outcome in each case. This outcome can be achieved by discarding the arguments that fail to meet the required burden of persuasion, i.e., the prosecution's argument for murder and the doctor's argument for her diligence, respectively. Our third example addresses anti-discrimination law. According to the European law against discrimination – or at least according to an interpretation of some of its controversial provisions – where there is evidence for discrimination in employment, it is on the employer to prove that there was no discrimination.

Example 3 (Anti-discrimination law example) *Let us consider a case in which a woman claims to have been discriminated against in her career on the basis of her sex, as she was passed over by male colleagues when promotions came available, and brings evidence showing that in her company all managerial positions are held by men, even though the company's personnel includes many equally qualified women, having worked for a long time in the company, and with equal or better performance. Assume that this practice is deemed to indicate the existence of gender-based discrimination, and that the employer fails to provide prevailing evidence that the woman was not discriminated against. It seems that it may be concluded that the woman was indeed discriminated against on the basis of her sex.*

In this paper, we put forward a formal model for the burden of persuasion which captures the patterns of reasoning that are exemplified above. Our model originates from legal considerations and is applied to legal examples. However, the issue of the burden of proof carries a significance that goes beyond the legal domain and involves other domains – public discourse, risk management, etc. – in which evidence and arguments are needed and corresponding responsibilities are allocated according to dialectical or organisational roles.

3. Argumentation Framework

We introduce a structured argumentation framework relying on a lightweight ASPIC$^+$-like argumentation system [4]. In a nutshell, arguments are produced from a set of defeasible rules, and attack relationships between arguments are drawn into argumentation graphs. Then arguments from the graph are labelled by following an acceptance labelling semantics that takes burdens of persuasion into account.

3.1. Defeasible theories, argumentation graphs and burden of persuasion

Let a literal be an atomic proposition or the negation of one.

Notation 3.1 *For any literal ϕ, its complement is denoted by $\bar{\phi}$. That is, if ϕ is a proposition p, then $\bar{\phi} = \neg p$, while if ϕ is $\neg p$, then $\bar{\phi}$ is p.*

Literals are brought into relation through defeasible rules.

Definition 3.1 *A **defeasible rule** r has the form: $\rho : \quad \phi_1, ..., \phi_n, \sim \phi'_1, ..., \sim \phi'_m \Rightarrow \psi$ with $0 \leq n$, and where*

- *ρ is the unique identifier for r, denoted by $N(r)$;*
- *each $\phi_1, \ldots \phi_n, \phi'_1, ..., \phi'_m, \psi$ is a literal;*
- *$\phi_1, \ldots \phi_n, \sim \phi'_1, ..., \sim \phi'_m$ are denoted by $Antecedent(r)$ and ψ by $Consequent(r)$;*
- *$\sim \phi$ denotes the weak negation (negation by failure) of ϕ: ϕ is an exception that would block the application of the rule whose antecedent includes $\sim \phi$.*

The name of a rule can be used as a literal to specify that the named rule is applicable, and its negation correspondingly to specify that the rule is inapplicable [5].
A superiority relation $\succ$ is defined over rules: $s \succ r$ states that rule s prevails over rule r.

Definition 3.2 *A **superiority relation** $\succ$ over a set of rules Rules is an antireflexive and antisymmetric binary relation over Rules, i.e., $\succ \subseteq Rules \times Rules$.*

A defeasible theory consists of a set of rules and a superiority relation over the rules.

Definition 3.3 *A **defeasible theory** is a tuple $\langle Rules, \succ \rangle$ where Rules is a set of rules, and $\succ$ is a superiority relation over Rules.*

Given a defeasible theory, by chaining rules from the theory we can construct arguments, as specified in the following definition; cf. [5,6,7].

Definition 3.4 *An* ***argument*** A *constructed from a defeasible theory* $\langle Rules, \succ \rangle$ *is a finite construct of the form:* $\mathsf{A}: \mathsf{A}_1, \dots \mathsf{A}_n \Rightarrow_r \phi$ *with* $0 \leq n$*, where*

- A *is the argument's unique identifier;*
- $\mathsf{A}_1, \dots, \mathsf{A}_n$ *are arguments constructed from the defeasible theory* $\langle Rules, \succ \rangle$*;*
- ϕ *is the* conclusion *of the argument, denoted by* $\mathsf{Conc}(\mathsf{A})$*;*
- $r : \mathsf{Conc}(\mathsf{A}_1), \dots, \mathsf{Conc}(\mathsf{A}_n) \Rightarrow \phi$ *is the top rule of* A*, denoted by* $\mathsf{TopRule}(\mathsf{A})$.

Notation 3.2 *Given an argument* $\mathsf{A} : \mathsf{A}_1, \dots \mathsf{A}_n \Rightarrow_r \phi$ *as in definition 3.4,* $\mathsf{Sub}(\mathsf{A})$ *denotes the* ***set of subarguments*** *of* A*, i.e.,* $\mathsf{Sub}(\mathsf{A}) = \mathsf{Sub}(\mathsf{A}_1) \cup \dots \cup \mathsf{Sub}(\mathsf{A}_n) \cup \{\mathsf{A}\}$. $\mathsf{DirectSub}(\mathsf{A})$ *denotes the* ***direct subarguments*** *of* A*, i.e.,* $\mathsf{DirectSub}(\mathsf{A}) = \{\mathsf{A}_1, \dots, \mathsf{A}_n\}$.

Preferences over arguments are defined via a last-link ordering: an argument A is preferred over another argument B if the top rule of A is stronger than the top rule of B.

Definition 3.5 *A* ***preference relation*** $\succ$ *is a binary relation over a set of arguments* $\mathscr{A}$*: an argument* A *is preferred to argument B, denoted by* $\mathsf{A} \succ \mathsf{B}$*, iff* $\mathsf{TopRule}(\mathsf{A}) \succ \mathsf{TopRule}(\mathsf{B})$.

We now provide definitions of possible collisions between arguments. Our definition focuses on cases in which an argument: (a) contradicts the conclusion of another argument (top-rebutting), or (b) denies the (applications of the) latter's top rule or contradicts a weak negation in the latter's body (top-undercutting).

Definition 3.6 A ***top-rebuts*** B *iff* $\mathsf{Conc}(\mathsf{A}) = \overline{\mathsf{Conc}(\mathsf{B})}$*, and* $\mathsf{B} \not\succ \mathsf{A}$*;* A ***strictly top-rebuts*** B *iff* $\mathsf{A} \succ \mathsf{B}$.

Definition 3.7 A ***top-undercuts*** B *iff*

- $\mathsf{Conc}(A) = \neg \mathsf{N}(r)$ *and* $TopRule(B) = r$*; or*
- $\mathsf{Conc}(A) = \phi$ *and* $\sim \phi \in Antecedent(\mathsf{TopRule}(B))$

Definition 3.8

- A ***top-attacks*** B *iff* A ***top-rebuts*** B *or* A ***top-undercuts*** B
- A ***strictly top-attacks*** B *iff* A ***strictly-top-rebuts*** B *or* A ***top-undercuts*** B

3.2. Labelling semantics

We use $\{\textsc{in}, \textsc{out}, \textsc{und}\}$-labellings, where each argument is labelled IN, OUT,or UND, depending on whether it is accepted, rejected, or undecided,respectively.

Definition 3.9 *Let G be an argumentation graph. An* $\{\textsc{in}, \textsc{out}, \textsc{und}\}$*-**labelling** L of G is a total* ***function*** $\mathscr{A}_G \rightarrow \{\textsc{in}, \textsc{out}, \textsc{und}\}$.

Notation 3.3 *Given a labelling L, we write* $\textsc{in}(L)$ *for* $\{\mathsf{A} | L(\mathsf{A}) = \textsc{in}\}$*,* $\textsc{out}(L)$ *for* $\{\mathsf{A} | L(\mathsf{A}) = \textsc{out}\}$ *and* $\textsc{und}(L)$ *for* $\{\mathsf{A} | L(\mathsf{A}) = \textsc{und}\}$.

Definition 3.10 ***Axargumentation graph*** *constructed from a defeasible theory T is a tuple* $\langle \mathscr{A}, \rightsquigarrow \rangle$*, where* $\mathscr{A}$ *is the set of all arguments constructed from T, and* $\rightsquigarrow$ *is an* attack *relation over* $\mathscr{A}$*.*

Notation 3.4 *Given an argumentation graph $G = \langle \mathscr{A}, \rightsquigarrow \rangle$, we write $\mathscr{A}_G$, and $\rightsquigarrow_G$ to denote the graph's arguments, and attacks respectively.*

Now, let us introduce the notion of a **BP-labelling**, namely a semantics which takes into account a set of burden of persuasion $\mathsf{BurdPers}$, where $\mathsf{BurdPers}$ is a set of literals, in determining the status of arguments.

Definition 3.11 *A **BP-labelling** of an argumentation graph G, relative to a set of burdens of persuasion $\mathsf{BurdPers}$, is a $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling s.t. $\forall \mathsf{A} \in \mathscr{A}_G$ with $\mathsf{Conc}(A) = \phi$*

1. $\mathsf{A} \in L(\mathsf{IN})$ *iff*
   (a) $\bar{\phi} \in \mathsf{BurdPers}$ *and*
      i. $\forall \mathsf{B} \in \mathscr{A}_G$ *such that* B *strictly top-attacks* $\mathsf{A} : \mathsf{B} \in L(\mathsf{OUT})$ *and*
      ii. $\forall\ \mathsf{A}' \in \mathsf{DirectSub}(\mathsf{A})$*:* $\mathsf{A}' \in L(\mathsf{IN})$ *or*
   (b) $\bar{\phi} \notin \mathsf{BurdPers}$ *and*
      i. $\forall\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *top-attacks* A*:* $\mathsf{B} \in L(\mathsf{OUT})$ *and*
      ii. $\forall\ \mathsf{A}' \in \mathsf{DirectSub}(\mathsf{A}) : \mathsf{A}' \in L(\mathsf{IN})$
2. $\mathsf{A} \in L(\mathsf{OUT})$ *iff*
   (a) $\phi \in \mathsf{BurdPers}$ *and*
      i. $\exists\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *top-attacks* A *and* $\mathsf{B} \notin L(\mathsf{OUT})$ *or*
      ii. $\exists\ \mathsf{A}' \in \mathsf{DirectSub}(\mathsf{A})$ *such that* $\mathsf{A}' \notin L(\mathsf{IN})$ *or*
   (b) $\phi \notin \mathsf{BurdPers}$ *and*
      i. $\exists\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *strictly top-attacks* A *and* $\mathsf{B} \in L(\mathsf{IN})$ *or*
      ii. $\exists\ \mathsf{A}' \in \mathsf{DirectSub}(\mathsf{A}) : \mathsf{A}' \in L(\mathsf{OUT})$*;*
3. $\mathsf{A} \in L(\mathsf{UND})$ *otherwise.*

In Definition 3.11, items *1)* and *2)* concern conditions for acceptance and rejection, respectively, based on burdens of persuasion.

Condition for acceptance. Item *1.(a)* concerns the case in which a burden of persuasion in on the complement $\bar{\phi}$ of the conclusion ϕ of argument A. A counterargument B for $\bar{\phi}$ is disfavoured by the burden of persuasion, while A is favoured. Thus, acceptance of A is not affected by a top-attacker B unless B is a strict top-attacker. Acceptance also require that all strict subarguments of A are IN. Item *1.(b)* concerns the case in which the conclusion of argument A is contradicted by a counterargument B on which there is no burden of persuasion. Here, there is no favour for A. Thus, acceptance of A may also be affected whn B is a non-strict top-attacker. Acceptance also require that all direct subarguments of A are IN.

Condition for rejection. Item *2.(a)* concerns the case in which the burden of persuasion is on the conclusion of argument A, so that A is disfavoured by the burden of persuasion. Here, the rejection of A may be determined by a counterargument B that is uncertain (UND), and also by any uncertainty on one of A's direct subarguments. Item *2.(b)* concerns the case in which there is no burden of persuasion on the conclusion of argument A.

Here, the rejection of A is only determined by a counterargument B of A that is IN or by a direct subargument of A that is OUT.
Note that the semantic just described does not always deliver a single labelling. This happens in particular in cases involving “team defeat”, or “team strict defeat”, i.e., in cases where argument A strictly attacks C, while being attacked by D, and B strictly attacks D, while being attacked by C. In such a case, both a labelling where A and B are IN and C and D are OUT and a labelling where all such arguments are UND fits the semantics. In all of the following examples, we will focus on the IN-minimal labelling, i.e., on the labelling where such arguments are labelled UND.

Example 4 (Civil law example) *According to the description of Example 2, let us consider the following rules (note that we assume that evidence is provided to establish the factual claims at issue, i.e., that the corresponding burdens of production are satisfied).*

$$\begin{array}{lll}
e1: ev_1 & e2: ev_2 & e3: ev_3 \\
er1: ev_1 \Rightarrow \neg guidelines & er2: ev_2 \Rightarrow guidelines & er3: ev_3 \Rightarrow harm \\
r1: guidelines \Rightarrow dueDiligence & r2: harm, \sim dueDiligence \Rightarrow liable &
\end{array}$$

We can then build the following arguments:

$$\begin{array}{lll}
\mathsf{A1} :\Rightarrow ev1 & \mathsf{A2} :\Rightarrow ev2 & \mathsf{A3} :\Rightarrow ev3 \\
\mathsf{A4} : \mathsf{A1} \Rightarrow \neg guidelines & \mathsf{A5} : \mathsf{A2} \Rightarrow guidelines & \mathsf{A6} : \mathsf{A3} \Rightarrow harm \\
\mathsf{A7} : \mathsf{A5} \Rightarrow dueDiligence & \mathsf{A8} : \mathsf{A6} \Rightarrow liable &
\end{array}$$

The argumentation graph and its grounded {IN, OUT, UND}*-labelling are depicted in Figure 1 (left), in which all arguments are* UND *except arguments for undisputed facts. The result is not satisfactory, according to the law, since it does not take into account the applicable burdens of persuasion. The doctor should have lost the case – i.e., be found liable – since she failed to discharge her burden of proving that she was diligent (non-negligent). The doctor’s failure results from the fact that it remains uncertain whether she followed the guidelines. To capture this aspect of the argument, we need to specify burdens of persuasion. Let us assume that (as under Italian law) we have* BurdPers = {*dueDiligence*, *liable*} *(i.e., the doctor has to provide a convincing argument that she was diligent, the patient has to provide a convincing argument for the doctor’s liability). As the burdened doctor’s argument for dueDiligence is* OUT*, her liability can be established even though it remains uncertain whether the guidelines were followed.* □

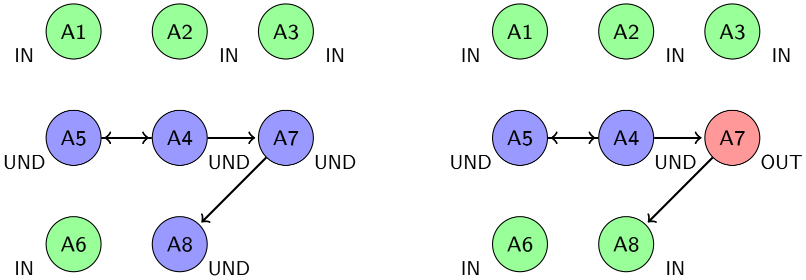


Figure 1. Grounded {IN, OUT, UND}-labelling of Example 2 in the absence of burdens of persuasion (left) and its BP-labelling with BurdPers = {*dueDiligence*, *liable*} (right).

This example shows how the model here presented allows us to deal with the *inversion of the burden of proof*, i.e., a situation in which one argument A is presented for a claim ϕ burdened with persuasion, and A (or a subargument of it) is attacked by a counterargument B whose conclusion ψ is also burdened with persuasion. If no convincing argument for ψ can be found, then the attack fails, and the uncertainty on ψ does not affect the status A.

Example 5 (Criminal law example) *According to the description in Example 1, let us consider the following rules (for simplicity's sake, we will not specify the evidence here, but we assume that all factual claims are supported by evidence):*

$f1\colon \Rightarrow killed$ $\qquad$ $f2\colon \Rightarrow intention$
$f3\colon \Rightarrow threatWithWeapon$ $\qquad$ $f4\colon \Rightarrow \neg threatWithWeapon$
$r1\colon threatWithWeapon \Rightarrow selfDefence$ $\qquad$ $r2\colon \neg threatWithWeapon \Rightarrow \neg selfDefence$
$r3\colon selfDefence \Rightarrow \neg murder$ $\qquad$ $r4\colon killed, intention \Rightarrow murder$

with $r3 \succ r4$. *We can build the following arguments:*

$\mathsf{A1} :\Rightarrow killed$ $\qquad$ $\mathsf{B1} :\Rightarrow threatWithWeapon$ $\qquad$ $\mathsf{C1} :\Rightarrow \neg threatWithWeapon$
$\mathsf{A2} :\Rightarrow intention$ $\qquad$ $\mathsf{B2} : \mathsf{B1} \Rightarrow selfDefence$ $\qquad$ $\mathsf{C2} : \mathsf{C1} \Rightarrow \neg selfDefence$
$\mathsf{A3} : \mathsf{A1}, \mathsf{A2} \Rightarrow murder$ $\qquad$ $\mathsf{B3} : \mathsf{B2} \Rightarrow \neg murder$

In the $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$*-labelling of Figure 2 (left), all arguments are* UND *except for the undisputed facts. Thus, in the absence of burdens of persuasion, we do not obtain the legally correct answer, namely, acquittal. To obtain acquittal we need to introduce burdens of persuasion. The prosecution has the burden of persuasion on murder: it therefore falls to the prosecution to persuade the judge that there was killing, that it was intentional, and that the killer did not act in self-defence. The BP-labelling is depicted in Figure 2 (right). The prosecution failed to meet its burden of proving murder, i.e., its argument is not convincing, since it remains undetermined whether there was self-defence. Therefore, murder is* OUT *and the presumed killer is to be acquitted.* □

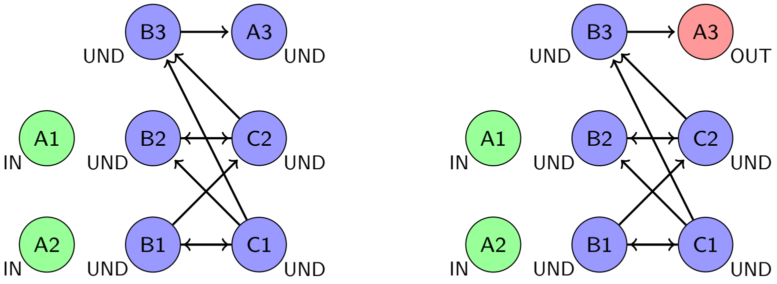


Figure 2. Grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of Example 1 in the absence of burdens of persuasion (left) and BP-labelling with the burden of persuasion $\mathsf{BurdPers} = \{murder\}$ (right).

3.3. Adversarial BP

Adversarial BP expands a BP-labelling approach with the idea that failure to meet a burden of persuasion on ϕ does not only mean that any argument for ϕ which fails to be

IN will be OUT. This also means that failure to provide an IN argument for ϕ will lead to $\neg\phi$ being established. For instance, failure to show that the accused is guilty will entail that he should be found innocent. Similarly, the plaintiff's failure to provide a convincing argument that he has a right to compensation for a certain event will entail that he has no right to be compensated. Or the burden of providing a convincing argument that a genetically modified crop is not harmful will entail that the crop is deemed to be harmful. Thus an adversarial burden of persuasion on a claim ϕ entails not only that arguments for ϕ will be OUT if they are not IN, but also that failure to establish ϕ entails ϕ's complement: "$\sim \phi \Rightarrow \neg\phi$". For instance, by adding a rule "$abp1 :\sim murder \Rightarrow \neg murder$" we would conclude in the criminal law example above that there is no murder. This is indeed what happens in criminal and other legal cases: failure to establish the prosecution's claim that a murder was committed or the plaintiff's claim that a compensation is due leads to the conclusion that there is no crime or that no compensation is due.

3.4. Reasoning with BPs

In the model described above, BPs are defined outside the legal knowledge base used. What if BPs become part of that rule base, so that we can reason to establish whether or not there is a BP on a literal ϕ.

Notation 3.5 *To specify, within our rule language, that there is a burden if persuasion on a literal ϕ, we write $bp(\phi)$.*

We propose the following definition.

Definition 3.12 *A **BP-labelling** of an argumentation graph G, relative to burdens of persuasion $BurdPers$, is a $\{\text{IN}, \text{OUT}, \text{UND}\}$-labelling such that $\forall \mathsf{A} \in \mathscr{A}_G$*

1. $\mathsf{A} \in L(\text{IN})$ *iff*
   (a) *there is an* IN *argument for $bp(\bar{\phi})$ and*
      i. $\forall \mathsf{B} \in \mathscr{A}_G$ *such that* B *strictly top attacks* $\mathsf{A} : \mathsf{B} \in L(\text{OUT})$ *and*
      ii. $\forall\ \mathsf{A}' \in DirectSub(\mathsf{A})$*:* $\mathsf{A}' \in L(\text{IN})$ *or*
   (b) *there no* IN *argument for $bp(\bar{\phi})$ and*
      i. $\forall\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *top attacks* A*:* $\mathsf{B} \in L(\text{OUT})$ *and*
      ii. $\forall\ \mathsf{A}' \in DirectSub(\mathsf{A}) : \mathsf{A}' \in L(\text{IN})$*;*
2. $\mathsf{A} \in L(\text{OUT})$ *iff*
   (a) *there is an* IN *argument for $bp(\phi)$ and*
      i. $\exists\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *top attacks* A *and* $\mathsf{B} \notin L(\text{OUT})$ *or*
      ii. $\exists\ \mathsf{A}' \in DirectSub(\mathsf{A})$ *such that* $\mathsf{A}' \notin L(\text{IN})$ *or*
   (b) *there is no* IN *argument for $bp(\phi)$ and*
      i. $\exists\ \mathsf{B} \in \mathscr{A}_G$ *such that* B *strictly top attacks* A *and* $\mathsf{B} \in L(\text{IN})$ *or*
      ii. $\exists\ \mathsf{A}' \in DirectSub(\mathsf{A}) : \mathsf{A}' \in L(\text{OUT})$*;*
3. $\mathsf{A} \in L(\text{UND})$ *otherwise.*

Accordingly, bp-statements can be part of the knowledge base or be inferred from it.

Example 6 (Antidiscrimination la example) *Consider, for instance, the following formalisation of the European nondiscrimination law in Example 3:*

$$\begin{array}{lll} e1 : ev1 & e2 : ev2 & e3 : ev3 \\ er1 : ev1 \Rightarrow indiciaDiscrim & er2 : ev2 \Rightarrow \neg discrim & er3 : ev3 \Rightarrow discrim \\ r1 : indiciaDscrim \Rightarrow \mathsf{bp}(\neg discrim) & & \end{array}$$

In this case, since there are indicia of discrimination, we can infer that there is the burden of proving nondiscrimination. Then, given that there is uncertainty about whether there was discrimination, the argument for nondiscrimination fails (it is OUT*), which means that the argument for discrimination is* IN. □

4. Conclusion

In this paper we provide and discussed a formal model for the burden of persuasion. The model shows how an allocation of the burden of persuasion may lead to a single outcome (IN arguments) in contexts in which the assessment of conflicting arguments would otherwise remain undecided. Our model explores the intersection between the burden of persuasion and argumentation labelling frameworks and provides a starting point for further research. In particular, it combines the insight of [8,9], where the burden of persuasion provides a criterion for adjudicating conflicts of arguments, and the insight of [10,11], where the satisfaction of burdens of argumentation depends on the dialectical status of the arguments at issue. The proposed model also deals with situations in which we have to combine a general burden of persuasion for one party (concerning the top conclusion to be reached), with inversions of the burden relative to specific propositions.

References

[1] D. Walton, *Burden of proof, presumption and argumentation*, Cambridge University Press, USA, 2014.

[2] R. Calegari and G. Sartor, Burden of Persuasion in Argumentation, in: *Proceedings 36th International Conference on Logic Programming (Technical Communications), ICLP 2020*, Vol. 325, Open Publishing Association, 2020, pp. 151–163. doi:10.4204/EPTCS.325.21.

[3] H. Prakken and G. Sartor, A Logical Analysis of Burdens of Proof, *Legal Evidence and Proof: Statistics, Stories, Logic* **1** (2010), 223–253.

[4] H. Prakken, An Abstract Framework for Argumentation with Structured Arguments, *Argument and Computation* **1** (2010), 93–124.

[5] S. Modgil and H. Prakken, The *ASPIC*$^+$ framework for structured argumentation: a tutorial, *Argument & Computation* **5**(1) (2014), 31–62.

[6] M. Caminada and L. Amgoud, On the Evaluation of Argumentation Formalisms, *Artificial Intelligence* **171**(5—6) (2007), 286–310.

[7] G. Vreeswijk, Abstract Argumentation Systems, *Artificial Intelligence* **90**(1–2) (1997), 225–279.

[8] H. Prakken and G. Sartor, More on Presumptions and Burdens of Proof, in: *21th Annual Conference on Legal Knowledge and Information Systems*, IOS, Groningen, The Netherlands, 2008, pp. 176–85.

[9] H. Prakken and G. Sartor, On Modelling Burdens and Standards of Proof in Structured Argumentation, in: *24th Annual Conference on Legal Knowledge and Information Systems*, IOS, 2011, pp. 83–92.

[10] T.F. Gordon, H. Prakken and D. Walton, The Carneades model of argument and burden of proof, *Artificial Intelligence* **171**(10) (2007), 875–896.

[11] T.F. Gordon and D.N. Walton, Proof Burdens and Standards, in: *Argumentation in Artificial Intelligence*, I. Rahwan and G.R. Simari, eds, Springer, 2009, pp. 239–60.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200846

A Taxonomy for the Representation of Privacy and Data Control Signals

Kartik CHAWLA [a,1], Joris HULSTIJN [a]
[a] *Department of Information Management, TiSEM, Tilburg University*

Abstract. In interacting with digital apps and services, users create digital identities and generate massive amounts of associated personal data. The relationship between the user and the service provider in such cases is, *inter alia*, a principal-agent relationship governed by a 'contract'. This contract is provided mostly in natural language text, however, and remains opaque to users. The need of the hour is multi-faceted documentation represented in machine-readable, natural language and graphical formats, to enable tools such as smart contracts and privacy assistants which could assist users in negotiating and monitoring agreements.

In this paper, we develop a Taxonomy for the Representation of Privacy and Data Control Signals. We focus on 'signals' because they play a crucial role in communicating how a service provider distinguishes itself in a market. We follow the methodology for developing taxonomies proposed by Nickerson et al. We start with a grounded analysis of the documentation of four smartphone-based fitness activity trackers, and compare these to insights from literature. We present the results of the first two iterations of the design cycle. Validation shows that the Taxonomy answers (10/14) relevant questions from Perera et al.'s requirements for the knowledge-modelling of privacy policies fully, (2/14) partially, and fails to answer (2/14). It also covers signals not identified by the checklist. We also validate the Taxonomy by applying it to extracts from documentation, and argue that it shows potential for the annotation and evaluation of privacy policies as well.

1. Introduction

In interacting with digital apps and services in what Hildebrandt [1] terms the modern '*on*life', users create digital identities and generate massive amounts of associated personal data. The interaction between the users of these devices and services, and their service providers is characterised by a variety of roles and relationships (e.g., user-service provider, consumer-trader, data subject-data controller).

Crucially, one of these relationships is that of a principal (the user) and an agent (the service provider) [2], as the user must rely on the service provider performing its task of protecting and enabling her 'privacy' with care and effort. The linchpin of any such relationship is a contract between the parties; in this case, this is quintessentially represented by the service provider's 'documentation', which hereinafter refers to the terms and conditions, privacy policy, and linked legal or technical documents [3]. Ideally,

[1] PhD Candidate, Department of Information Management, TiSEM, Tilburg Universty, Warandelaan 2, 5037 AB Tilburg, The Netherlands; E-mail: k.chawla@uvt.nl. My thanks to Bert-Jan Butijn, Mariana Evram, Rajesh Chawla, and Ruud van Cruchten for their comments and feedback.

these digital contracts should be negotiated and their implementation monitored to the benefit of both parties. In practice, however, these contracts are often ignored, and even if they are not, are difficult to comprehend, note or manage [4,5]. Consequently, one of the biggest issues in the contemporary privacy debate is enabling users to negotiate the default conditions and monitor the actions of all of the apps, services and devices that collect their data. This is a problem for users, but also for service providers, data protection authorities, and the market as a whole [6].

In research and in practice, we find a variety of initiatives to address this issue. One stream of research focuses on negotiation protocols such as the P3P [7], or the creation of privacy assistants [8], analogous to the idea of a 'butler' [9]. Others investigate the automatic annotation or evaluation of privacy policies [10,11,12,13]. A third stream focuses on the development of 'Personal Data Stores' (PDSs),[2], which are systems that provide an architecture allowing users retain and manage their own data. There are almost certainly other initiatives as well.

Each of these proposed solutions needs to work with a representation of the 'Documentation', or at least the privacy policy, a role currently fulfilled largely by natural language text. Morel and Pardo [14] survey the means of representation of privacy policies and find three main dimensions: natural language, graphical and machine-readable, each fulfilling some particular needs of the communities they originate from. However, none can single-handedly fulfill the requirements of all communities (e.g., legal compliance, understandability and enforceability). Morel and Pardo argue that what is needed instead is a *multi-faceted privacy policy*, one that covers all three dimensions simultaneously [14]. We agree. Multi-faceted documentation would allow users to process and manage their interactions with digital services according to their privacy preferences better than natural language documentation alone. A secondary benefit of machine-readable taxonomies is easier enforcement [14]. We would add that machine-readable policies also allow for the creation of privacy management tools for the management of all the policies a user 'agrees' to, similar to current password managers like LastPass[3], and for the customisation of 'notice'.

For the creation of machine-readable and graphical documentation, a pre-requisite is a categorisation and coherent representation of a service provider's data practices. This is not an easy task [11]. The objective of this research is to develop a *Taxonomy for the Representation of Privacy and Data Control Signals*. The term 'signal' comes from contract theory (law and economics), and refers to credible information conveyed by the agent to the principal, in a market with asymmetric information [15]. A signal is meant to reveal certain information about the agent's behaviour (here: data handling and control practices) to the principal, so that they can react accordingly. The natural language documentation from the service provider is supposed to convey some of these signals to the user, and in its final version it represents a 'meeting of the minds' as to what happens to a user's data.

In this paper, we report on our efforts to identify a taxonomy to represent such 'privacy and data control signals', as communicated in the Documentation. We employ design science for this task [16], specifically Nickerson et al.'s [17] methodology for the development of taxonomies. This research answers the knowledge question 'What infor-

[2] SOLID (`https://inrupt.com/solid`), Hub of All Things, (`https://www.hubofallthings.com/`)
[3] `https://www.lastpass.com`

mation should a multi-faceted documentation be able to represent?'. The design question 'How should this information be represented?' will be the focus of further research.

We present the results from the first two iterations of the method. We identify a complex and multi-layered Taxonomy, based on four empirical samples. We evaluate the Taxonomy at this stage using Perera et al's [18] checklist of questions that a knowledge-based modelling of privacy policies should be able to answer. In the long run, however, as Nickerson et al. [17] note, a taxonomy is only useful if it is used. We aim to bring this Taxonomy to a level where it can actually be used in practice, *inter alia*, for multi-faceted privacy documentation, and annotation schemes, and subsequently for the creation of smart contracts and privacy assistants. The next steps in this project consist of two inter-linked stages: running further iterations of the Taxonomy's design cycle, and using the Taxonomy for the design and implementation of Multi-faceted Privacy Policies.

2. Theory

We live today in what Mirelle Hildebrant [1] calls a 'new animism': a transformative '*on*life' situated *"beyond the increasingly artificial distinction between online and offline."* In interacting with this '*on*life', we create digital identities and generate massive amounts of associated data from the increasing number of devices and services that we use in the course of our daily lives, from the banal to the exceptional. This creates significant challenges for privacy-conscious users.

2.1. Notice and Choice

Each digital service comes with its own documentation, its own 'contract'. The user's consent to this agreement rests, precariously, on the infamous principle of 'notice and choice' [19]. This is currently implemented largely by a service provider's natural language documentation, though various parts thereof can be scattered throughout a user's experience with a service.

This mechanism simply cannot keep up with the evolution of technology and the cornucopia of information that needs to be conveyed. A significant amount of literature that the failure of the 'notice and choice' mechanism in general [20,19,5], and the failure of privacy policies in specific [21]. The issue is not limited to privacy – 'online' or 'digital' contracts are generally associated with significant information and negotiation power asymmetries [4].

As Calo [19] notes, however, the problem doesn't lie in the idea of 'notice and choice' but in its implementation. There will necessarily always be a component of 'notice' and 'choice' in such interactions. Even the GDPR requires transparency about processing operations and their purposes (Recital 60), and requires consent to be *"freely given, specific, informed and unambiguous"* (Art. 4(11)) where it is necessary. At the normative level, this is necessary. At the practical level, such transparency and active choice is difficult to implement for service providers, and difficult to comprehend and use for end-users. It results in documentation being written for the purpose of legal compliance [22], rather than for the communication of privacy signals to users [14]. It also creates a potential opportunity for exploitation by rent-seekers. For effective notice and choice, the documentation, and the privacy and data control signals it contains, needs to evolve into a more manageable format.

2.2. Privacy and Privacy Signals

In the market for digital services, a privacy-conscious user, (seen here as principal [2]) must identify the service providers that match her privacy preferences and communicate such preferences to them. As the agent, the service provider's must communicate 'signals' about the quality of their service and about how they accommodate and respect their users' privacy preferences. Neither of these tasks are easy.

Before we can identify them, we must define what we mean by 'privacy and data control signals'. Vila et al. [6] analysed the market for privacy in websites and described it as a market with asymmetric information – exactly the type of market where 'signals' become relevant. They define 'signals' as *"a means by which privacy-respecting sites can differentiate themselves from their defecting competitors"* [6].

Vila et al. [6] mention a 'strong' privacy policy as an example of such a signal. But what is a 'strong' privacy policy? The fact of the matter is that a strong privacy policy is often the one that matches the user's preferences. A policy that one user may consider 'weak' might be entirely acceptable by the standards of another user. The GDPR provides a set of 'default rules', a minimum standard that every policy must comply with. Beyond that, however, there is still a lot of room left for negotiation, or rather selection. A practical example of this is the collection of user data on websites via cookies, and the options given to users for their consent to different types of data collection.

A signal can therefore be defined as information that allows users to identify the whether the data management practices of a service provider matches their expectations or preferences or not, allowing them to adjust their behaviour accordingly. So any type of information that reveals how the service provider handles its user data, and which 'choices' it allows users, will count as such a signal. This is a rather broad definition, but we will limit our scope by focusing only on signals in the service provider's documentation. We also focus exclusively on the user-service provider relationship, even though third-party integration of services and devices creates important consequences for a user's privacy.

We focus specifically on the signals that correlate with Westin's [23] definition of privacy as an individual's right *"to control, edit, manage, and delete information about them*[selves] *and decide when, how, and to what extent information is communicated to others."* 'Signals' here include legal and technical information. An illustration of such signals is Naeini et al. [24]'s work on standardised 'labels' for privacy in IoT devices.

2.3. Open Texture

Legal documents, often suffer from the 'open texture' of language; i.e., they employ open-ended terms (sometimes intentionally) in order to account for as many potential eventualities as possible [25] and to comply with legal requirements [22]. These are not written, primarily, for the communication of these 'signals'. For instance, privacy policies often indicate that an action 'may' be conducted without specifying whether it is actually conducted or not, or use inclusive rather than exhaustive lists at many places (e.g. *"...and other information you might share with us"*).

This has two important consequences. First, there is necessarily some loss of information between 'open-textured' legal documentation and the specificity of a taxonomy-based representation. Second, this along with the dynamic nature of the context means that any taxonomy would need to be flexible, frequently updated, and carefully applied.

3. Related Work

Our focus is on privacy and data control signals, their representations, taxonomies, and evaluations. There is related work in legal (e.g. [26]) and technical [27] research. Directly relevant is research on the annotation or evaluation of privacy policies [11], and the development of knowledge-based languages for their representation [18]. A search of the keywords 'privacy policy' & 'annotation' on SCOPUS, Web of Science and IEEE identifies 21 papers that seem relevant based on their titles and abstracts. Of these, 10 contain or utilise annotation schemes or other categorisations of privacy policies content. Most use a novel scheme, though two reuse Wilson et al.'s corpus and scheme [12].

There are two main limitations of these categorisations. First, many are limited to *the* privacy policy, ignoring additional documentation and the settings and choices offered in the software itself. Second, many tend to focus on the readability, comprehensibility or compliance of the text of the privacy policies. Few provide an analysis of the content of privacy policies, with exceptions such as Antón & Earp [27] (requirements engineering perspective) for and Wilson et al. [12] (computational linguistics).

4. Methodology

We adopt Nickerson et al.'s [17] method for the development of taxonomies in a domain of interest, due to its suitability for the task at hand. The method consists of two cycles, Empirical-to-Conceptual (E2C) and Conceptual-to-Empirical (C2E), in which either dimensions, or characteristics are added, on the basis of empirical material and literature respectively. Table 1 summarises the the application of the methodology in this project.

Table 1. Nickerson et al.'s [17] Methodology, As Applied

Methodology Components	Methodology Application
1. Meta Characteristic	**Signals**: Information regarding a service provider's data handling practices and the control allowed to users, relevant for ascertaining compatibility with a user's privacy preferences. **Expected Use**: The design of a multi-faceted privacy policy, privacy-policy annotation. **Purpose**: The categorisation of the information conveyed via a natural language privacy policy and associated documentation
2. Ending Conditions	**Objective Condition 1**: No new dimensions or characteristics added in the last iteration **Objective Condition 2**: No dimensions or characteristics merged or split in the last iteration **Subjective conditions**: Conciseness, Robustness, Comprehensiveness, Extendibility, and Explanability
3. Empirical to Conceptual Approach	3.1 The sampling of four diverse examples of 'smartphone-based fitness activity tracking' applications and their documentation. 3.2 The coding and organisation of 'signals' from the documentation with Atlas.ti. 3.3 The categorisation of the signals, providing a first iteration of the Taxonomy.
4. Conceptual to Empirical Approach	4.1 The identification of 5 selected papers relevant to the taxonomy. 4.2 Identifying additional dimensions and characteristics from the selected literature. 4.3 The addition or reorganisation of the Taxonomy taking into account the insights from the literature.

4.1. Sample Selection

For the first iteration of this project we selected four ‘fitness activity tracking’ applications on the Google Play Store: Strava (socially-oriented), Runkeeper (was subject to a complaint for privacy violations by the Norwegian Consumer Council), Adidas Runtastic (Affiliated entity), and OpenTracks (privacy-oriented). The documentation of these services was obtained via the links provided on their Google Play Store pages, other linked pages as needed, and the relevant ‘settings’ page in the applications. Where the policies linked to policies of third-party service providers, the latter were not analysed.

In the first cycle, the documentation collected from the samples was analysed and coded with Atlas.ti using a grounded approach. The identified codes were then structured to identify the relevant dimensions and attributes, taking into account definitions and concepts from the GDPR. The second cycle took into account the categorisations used in literature on the topic, specifically: Wilson et al. [12,11], Morel and Pardo [14], Bhatia et al. [28] and Contissa et al. [10]. These papers were selected for their relevance. The Taxonomy resulting from the E2C cycle most closely resembles Wilson et al. [11]’s work though it is structured differently and adds a few dimensions, which can be taken as at least a partial validation of it. Similarly, categorising the ‘types of data collected’ is tricky, and Bhatia et al. [28] offer an alternative to the results of the E2C cycle. After these two cycles, further iterations will follow. In the next iteration, we aim to include a larger sample of applications and documentation and literature.

5. Results: A Taxonomy for Privacy and Data Control Signals

This exercise results in a complex and multi-layered Taxonomy for the Representation of Privacy and Data Control Signals. Due to space constraints, we can only present the first two levels of the dimensions of the resulting Taxonomy in (Fig 1) and explain some of the important dimensions below. The full Taxonomy is available in graphical and tabular representations on Github [4].

The taxonomy contains three levels of dimensions. The first level includes: ‘Policy Meta-Data’, ‘Data and Control’, and ‘Processing and Usage’. Data and Control includes, *inter alia*, signals about what data is collected and what controls users are allowed; Processing and Usage relates to processing activities, entities and purposes; and Policy Meta-data covers some more abstract, contextual information, and information regarding the policy itself. To allow for documentation’s open texture, we keep the ‘Types of data collected’ agnostic to the specific characteristics of the data or the sensors from which it is collected. These are covered under a separate dimension (‘Data Characteristics’). Bhatia et al. [28] provide an alternative lexicon for this. We also identify signals relating to User Control, further divided into control ‘Options’, ‘Channels’ and ‘Limitations’. This is similar to the factors in the design space for privacy notices noted by Schaub et al.[22]. We separate the anonymity of a user’s profile on a platform (‘Active Audience’) from the risk that the service provider will share their data with a third party (‘Data Shared with’).

We identify three distinct but related types of signals in the Documentation: legal basis, purpose and functionality. Each conveys some information about why a user’s data is collected, but at different levels of abstraction. The ‘basis’ is the most abstract, relating

[4] https://github.com/KartikChawla-droid/Taxonomy_Privacy_Data_Control_Signals

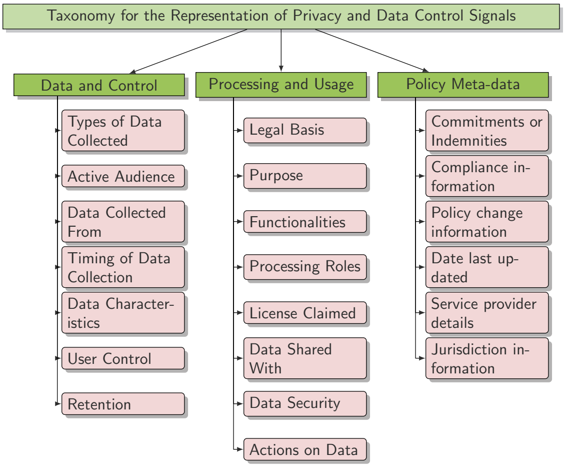


Figure 1. The First Two Layers Of The Dimensions Of The Taxonomy

to Art. 6(1) of the GDPR. The ’purpose’ is slightly more specific but still vague, while ‘functionality’ is the most specific and relates closely to the technical aspects of the service. For instance, the purpose for the collection of a user’s account data and location data, both, is ‘provision of service’. The distinction lies in the functionality: ‘account creation’ and ‘fitness activity tracking’ respectively. Functionality, furthermore, allows for a comparison between services: if two unrelated services both offer a ’social network’ functionality, even if they cannot be compared as a whole, the implementation of this functionality and the data it collects can be compared between the two services.

6. Evaluation and Discussion

The Taxonomy identifies a variety of factors not identified by previous research: the distinction between user control options, limitations and channels; the commitments and indemnities; whether the data is licenseable and whether such a license is claimed or not; and, crucially, the distinction between the ‘legal basis’ and ‘purpose’ of data collection and the ‘functionality’ it links to. The variety of dimensions identified by it verifies the depth and complexity of the information conveyed via the documentation.

The purpose of the Taxonomy is to enable representation of privacy signals in a multi-faceted format. The individual elements of the Taxonomy already identify some relevant signals, but combinations thereof identify even more, making explicit the links between different types of information. For instance, ‘Types of data collected’ is a signal in and of itself but its combination with ‘functionalities’ or ‘data shared with’ communicates a different, but still crucial, type of a signal altogether. Clustering ‘types of data collected’ with ‘functionalities’ tells the reader what data is funneled into which functionalities. Keeping in mind that the same data may go into multiple functionalities, this allows a user to evaluate the ‘exchange’; i.e., it allows users to see which functionalities require which types of data, and evaluate whether they are willing to forgo with such data to receive these functionalities. An analogue to this in practice is the separation of ‘cookies’ by functionality (such as: ‘necessary’, ‘marketing’ or ‘analytics’) available in cookie consent managers, and the separation of microservices in cloud computing.

The results of the Taxonomy also justify the extension of its scope beyond the privacy policy, and beyond compliance with regulations such as the GDPR. The dimensions relating to the licenseability of the data would not have been identified from the privacy policy alone, and the ‘functionalities’ and ‘applicable jurisdiction’ are more evident in the Terms and Conditions in some cases. A variety of signals go beyond pure compliance with the GDPR. For instance, Table 2 illustrates a possible application of the Taxonomy, using the OpenTracks privacy policy as an example[5]. Note that this policy would fail a test for GDPR compliance, but then it doesn’t need to comply because there is no third-party processing of data! However, even this two-sentence documentation contains important information that is captured by the Taxonomy. A more extended list of examples for evaluation is available in the Github repository.

Given the diversity and dynamic nature of the information conveyed by the sample space, we would argue that rather than specifying all the information that could potentially be conveyed, it would be more efficient to specify a flexible code ‘library’ or ‘package’, based on this taxonomy, that can enable the writers of the documentation to add new information on the fly.

Table 2. Coding From Text and Application Of Taxonomy To Opentracks Sample

Sample Text	**Taxonomy Coding**
OpenTracks does only store data on the local device	retention_location: local
that is relevant for tracking your sport exercise .	functionality: activity tracking
Stored data is not transmitted from the app itself to a third party.	type_of_data_collected: [none]

We further evaluate the Taxonomy against Perera et al.’s [18] checklist about the information a representation language for privacy policies should be able to convey. The checklist contains a total of 17 questions, 14 of which are relevant for the ‘content’ of the representation. The Taxonomy presented here can answer 12 questions completely, 2 partially, and fails to answer 2. For further details, please refer to Github. One unanswered question tells us that we need to add ‘Methods of data collection’, in the next cycle. The second missing question asks what information is data controllers expect to discover from the user’s data, but this information is not present in the sample documentation.

7. Limitations and Further research

This research has certain limitations. First, an application’s effect on a user’s privacy must take into account its technical context. Applications are necessarily deployed on a hardware and software stack (‘vertical stack’) and may be integrated with third-party applications and services working in parallel (‘horizontal integration’). Both affect the functioning of the application and the user’s privacy. We have not taken these vertical and horizontal interfaces into account, but a useful taxonomy needs to be ‘modular’ to accommodate this layering. Second, the open texture of legal documents means that a certain loss of information or ambiguity in the the taxonomy is perhaps inevitable (e.g., ‘open-ended’ as an attribute for ‘purpose’). Third, our analysis is limited to privacy signals contained in the documentation and technical implementation. There are further

[5] https://opentrack.run/about/privacy.html

market-oriented signals which have not been included here, such as reputation, size, business model, and of course the code (as much as is observable). These and further signals regarding the context [29] and consumer rights [30] should be included in further research. Particularly, information regarding the API calls made or enabled by an application, if available, should also be included in the Taxonomy. Fourth, the Taxonomy is limited to the *representation* of 'signals'. The natural follow-up question is whether the communicated signals are legitimate or not. This would require a more elaborate system for monitoring a service provider's behaviour and testing compliance with the agreements [6]. That makes a good topic for future research.

8. Conclusion

For online services, we look at the relationship between users and service providers from the perspective of principal-agent theory [2]. This relationship exists in a market with asymmetric information, which means that 'signals' about the digital service are crucial for users. From the empirical analysis it is evident that a service provider's documentation provides a lot of privacy and data control signals in a relatively unstructured form. However, currently, signals about privacy and data control tend to get lost in natural language documentation. The negotiation and monitoring costs the user must bear to ensure an optimal contract are too high without support tools. Even if a user retains technical control over her data with a PDS system, she would still need legal support tools for negotiating and monitoring access to her data. The depth and complexity of the information, even when viewed through the lens of the Taxonomy, makes the need for machine readable or annotated privacy policies self-evident even without taking into account the behavioural issues pointed out by Acquisti et al. [31].

This paper presents the results of the first two iterations of a design science project for the development of a 'Taxonomy for the Representation of Privacy and Control Signals' that allows for a machine-readable representation of these signals. We identify crucial dimensions not covered by previous taxonomies, based on an empirical analysis of four sample documentations. This answers the knowledge question 'What information should be represented in a multi-faceted documentation on privacy and data controls?'. The Taxonomy still requires further iterations, which are planned. At the same time, we will attempt to use this knowledge model for the annotation and evaluation of privacy policies and the development and design of smart contracts and privacy assistants. That is, we will attempt to use this to answer the design question 'How should this information be represented?' in further research. We will conduct a survey of relevant tools for the latter (e.g. with protégé, as XML, or as a library) as well.

References

[1] M. Hildebrandt. Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology. Cheltenham Edward Elgar Publishing; 2015.

[2] EA Posner. Agency Models in Law and Economics. John M. Olin Program in Law and Economics Working Paper No. 92. 2000.

[3] M. Pavis. Paris Tribunal Guts Twitter'S T&Cs… Including The Copyright Clause For User-Generated Content. [online] The IPKat. Available at: https://ipkitten.blogspot.com/2018/09/paris-tribunal-guts-twitters-t.html [Accessed 23 October 2020].

[4] R. Momberg. Standard Terms and Transparency in Online Contracts. In A. De Francheschi, editors, *Standard Terms and Transparency in Online Contracts*. Intersentia, 2016:189-206.

[5] K. Martin, Transaction costs, privacy, and trust: The laudable goals and ultimate failure of notice and choice to respect privacy online. FM [Internet]. 2013 Dec. 15 [cited 2020 Oct. 23]; 18(12). Available from: https://firstmonday.org/ojs/index.php/fm/article/view/4838.

[6] T. Vila, R. Greenstadt, and D. Molnar. Why We Can't Be Bothered to Read Privacy Policies - Models of Privacy Economics as a Lemons Market.*Proceedings of the 5th ICEC*. 2003.

[7] I. Reay, S. Dick, and J. Miller. An analysis of privacy signals on the World Wide Web: Past, present and future. Inf Sci. 2009 Mar 29; 179:1102–15.

[8] A. Das, M. Degeling, D. Smullen, N.M. and Sadeh. Personalized Privacy Assistants for the Internet of Things: Providing Users with Notice and Choice. IEEE Pervasive Computing. 2018; 17:35–46.

[9] Lawrence Lessig. Code and Other Laws of Cyberspace. Basic Books; 1999.

[10] G. Contissa, K. Docter, F. Lagioia, M. Lippi, H.W. Micklitz, P. Palka, G. Sartor, P. and Torroni. Claudette Meets GDPR : Automating the Evaluation of Privacy Policies Using Artificial Intelligence. SSRN Electronic Journal. 2018 Jan 1.

[11] S. Wilson, F. Schaub, F. Liu, K.M. Sathyendra, D. Smullen, S. Zimmeck, R. Ramanath, P. Story, F. Liu,N. Sadeh, and N.A. Smith. Analyzing Privacy Policies at Scale: From Crowdsourcing to Automated Annotations. ACM Trans Web. 2018 Dec 13.

[12] S. Wilson, F. Schaub, A.A. Dara, F. Liu, S. Cherivirala, P. Giovanni Leon, M. Schaarup Andersen, S. Zimmeck, K.M. Sathyendra, N.C. Russell, T.B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh. The Creation and Analysis of a Website Privacy Policy Corpus.*Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*. Berlin, Germany, 2016 Aug :1330–40.

[13] D. Audich, R. Dara, and B. Nonnecke. Privacy Policy Annotation for Semi-automated Analysis: A Cost-Effective Approach. 2018 Jun 30;29–44.

[14] V. Morel and R. Pardo. Three Dimensions of Privacy Policies. Research Report 9287. Inria, Project-Teams Privatics, 2019 Nov.

[15] M. Spence. Job Market Signaling In P. Diamond and M. Rothschild. Uncertainty in Economics. Academic Press, 1978 Jan 1:281–306.

[16] R.J. Wieringa. Design Science Methodology for Information Systems and Software Engineering. Springer, 2014.

[17] R.C. Nickerson, U. Varshney, and J. Muntermann. A method for taxonomy development and its application in information systems. European Journal of Information Systems. 2013;22:336–59.

[18] C. Perera, C. Liu, R. Ranjan, L. Wang, and A. Zomaya. Privacy Knowledge Modelling for Internet of Things: A Look Back. Computer. 2016; 49.

[19] MR Calo. Against Notice Skepticism in Privacy (and Elsewhere). Notre Dame Law Review. 2011;87:1027.

[20] FH Cate and V. Mayer-Schönberger. Notice and consent in a world of Big Data. International Data Privacy Law. 2013 May 1; 3:67–73.

[21] R. W. Proctor, M. Athar Ali, and L. Kim-Phoung L. Vu. Examining Usability of Web Privacy Policies. International Journal of Human-Computer Interaction. 2008; 24:307–28.

[22] F. Schaub, R. Balebako, L.F. and Cranor LF. Designing Effective Privacy Notices and Controls. IEEE Internet Computing. 2017 May; 21:70–7.

[23] A.F. Westin. Privacy and Freedom. Bodley Head, 1967.

[24] P.E. Naeini, Y. Agarwal, L. Cranor, and H. Hibshi. Ask the Experts: What Should Be on an IoT Privacy and Security Label? 2020 Feb 11.

[25] H.L.A. Hart. The Concept of Law. Third. Oxford University Press, 2012.

[26] D.J. Solove. A Taxonomy of Privacy. Univ Pa Law Rev. 2006 Jan; 154:477–564.

[27] AI Antón and JB Earp. A requirements taxonomy for reducing Web site privacy vulnerabilities. Requirements Engineering. 2004 Aug 1; 9:169–85.

[28] J. Bhatia and T.D. Breaux. Towards an information type lexicon for privacy policies. 2015 :19–24.

[29] H. Nissenbaum. A Contextual Approach to Privacy Online. Daedalus. 2011 Oct 1; 140:32–48.

[30] M. Loos and J. Luzak. Wanted: a Bigger Stick. On Unfair Terms in Consumer Contracts with Online Service Providers. Journal of Consumer Policy. 2016 Mar; 39:63–90.

[31] A. Acquisti, L. Brandimarte, and G. Loewenstein. Privacy and human behavior in the age of information. Science. 2015; 347:509–14.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200847

Events Matter: Extraction of Events from Court Decisions

Erwin FILTZ [a,b], María NAVAS-LORO [c], Cristiana SANTOS [d], Axel POLLERES [a] and Sabrina KIRRANE [a]

[a] *Vienna University of Economics and Business*
[b] *Siemens AG Österreich*
[c] *Universidad Politécnica de Madrid – Ontology Engineering Group, Madrid, Spain*
[d] *Utrecht University*

Abstract. The analysis of court decisions and associated events is part of the daily life of many legal practitioners. Unfortunately, since court decision texts can often be long and complex, bringing all events relating to a case in order, to understand their connections and durations is a time-consuming task. Automated court decision timeline generation could provide a visual overview of what happened throughout a case by representing the main legal events, together with relevant temporal information. Tools and technologies to extract events from court decisions however are still underdeveloped. To this end, in the current paper we compare the effectiveness of three different extraction mechanisms, namely deep learning, conditional random fields, and rule-based method, to facilitate automated extraction of events and their components (i.e., the event type, who was involved, and when it happened). In addition, we provide a corpus of manually annotated decisions of the European Court of Human Rights, which shall serve as a gold standard not only for our own evaluation, but also for the research community for comparison and further experiments.

Keywords. event extraction, named entity recognition, court decisions

1. Introduction

Court decisions are an important source of law information for legal practitioners: they elaborate on the facts of a case, involved parties, interpretations of the circumstances, the applicable law and legal principles, and finally the legal assessment leading to the decision. Legal professionals constantly extract, interpret and reason with and about prior cases whilst arguing for a decision in a current, undecided case. However, court decisions texts can be long and complex and thus time-consuming to read. Therefore it would be beneficial to find a means to provide a quick overview of a case, thereby helping to turn decisions into operational, consumable and actionable legal knowledge.

In this work we focus specifically on using Natural Language Processing (NLP) techniques to automatically extract the essence of a court case. Besides extracting general legal rules from individual cases, we aim at providing a quick overview of what happened, who was involved and when the event took place. In the terminology of NLP, event extraction can be treated as a *text classification task* aiming at assigning text fragments (typically, paragraphs, sentences or smaller parts of documents) to predefined (event) classes [1]. Another, related NLP task is Named Entity Recognition (NER) which extracts entities referred to in texts and classifies them into categories [2], for instance people, places and organizations; moreover, named entities can also be domain-specific, for instance, courts or laws. Event extraction can benefit from NER, since it can be used to enrich events with relevant information, such as the parties involved. This paper focuses on the extraction of events and their components from court decisions of the European Court of Human Rights (ECHR)[1] based on a sample thereof.

Summarizing our *contributions*, we: (i) provide a corpus of manually annotated ECHR decisions; (ii) perform a comparison of different approaches to automatically extract events and their components – implementations as well as our evaluation results are made available on GitHub; and (iii) introduce a prototypical web interface that can be used to display court decisions along with their extracted timelines.

The remainder of this paper is structured as follows. We outline related works in Section 2. Our corpus as well as the annotation methodology is described in Section 3. Section 4 contains information about the compared event classification and NER approaches, followed by Section 5 discussing evaluation results. Section 6 provides conclusions.

2. Related Work

Recent advances in NLP are often based on embedding text in multidimensional vector space, with neural network architectures being trained on such numeric representations. Such methods yield in re-usable, publicly available language models trained on large corpora of texts, where embeddings can be created on different levels, for instance words, sentences and documents. While pre-training models on large corpora of generic texts is a very time-consuming process [3], adapting (aka fine-tuning) such generic models to domain-specific language is often less demanding.

Overviews on diverse automated event extraction approaches in the general domain can be found in literature [4,5]. Specifically in the legal domain [6], existing work usually involves searching for *ad hoc* definitions of events, ignoring general event annotation schemas such as the ACE 2005 model [7]. Several approaches tend to be supported by patterns, using manually crafted rules or semantic role labeling techniques [8,9,10,11]. Other approaches do not search for events specifically, but target legal case factors [12].

The automated generation of timelines out of annotated documents could help to get a better and faster understanding of the content of a document. Existing work focusing on this task include Linea [13], a system that is able to build and navigate timelines from unstructured text, and TimeLineCurator [14] a system that is primarily designed to allow journalists to generate temporal stories, however can be used to produce a timeline from any free text or url. Furthermore, the creation of timelines has also been investigated in other domains, such as medicine [15,16] and journalism [17]. We refer to [14] for further details on the respective approaches.

Regarding corpora in the legal domain, court decisions of the ECHR have also been used in literature for different tasks [18,19]. Nevertheless, very few annotated corpora from the legal domain have been made available, and to the best of our knowledge none of them considers events.

[1] https://echr.coe.int/

3. Corpus and annotation methodology

This section describes the ECHR corpus as well as our annotation methodology.

Description of the corpus. The corpus consists of 30 decisions of the ECHR. The ECHR decisions were chosen because they contain: i) different types of time-related events concerning different actors in comparison with the decisions of the Court of Justice of the EU [6]; and ii) a standard structure in which different legal events are embedded. ECHR decisions are divided into several sections containing specific information according to Rule 74 of the Rules of the Court [20]: the *Preamble* and the *Introduction* are followed by *Facts* which contain information about the formal procedure and the circumstances of the case providing details about what happened. The following *Law* section describes the legal situations and states the alleged violation(s). The document concludes with the *Decision* section. For the purposes of this paper, we use the mentioned document structure excluding the *Law* section and focus on the procedure, circumstances and decision.

Annotation methodology. The corpus was annotated by two legal experts in several iterations. The experts annotated independently and then met with a third person to reach a consensus on the disagreements. In this work, as we focus on event extraction aimed to automated court decision timeline generation, we were interested in information that is relevant to searching for or extracting time-related information, such as events, processes, temporal information, and the parties involved. As time-related events of cases are linguistically expressed, we annotated the most salient candidate passages thereof. The decisions were manually annotated following the frame "who-when-what". To illustrate the applicability thereof, we make use of an annotated paragraph of the case Altay v. Turkey (no. 2), no. 11236/09, 9 April 2019 (a case referring to respect of private life):

"On 29 May 2008 the applicant lodged an application with the Edirne Enforcement Court for the restriction on the conversations between him and his lawyer to be lifted."

"Who" corresponds to the subject of the event, which can either be a subject, but also an object (i.e., an application); in the example, the subject is "(the) applicant". *"When"* refers to the date of the event, or to any temporal reference thereto; in the paragraph considered, the "when" is the "29 May 2008". *"What"* usually corresponds to the main verb reflecting the baseline of all the paragraph (which in this case is "lodged"); additionally, we include thereto a *complementing* verb or object whenever the core verb is not self-explicit or requires an extension to attain a sufficient meaning; in the paragraph considered, the "what" is "lodged an application". Another e.g. is "dismiss an action". *"Event"* relates to the extent of text containing contextual event-related information. The *type* of such annotations can be either *circumstance* – meaning that the event correspond to the facts under judgment; or *procedure*– wherein the event belongs to the procedural dimension of the case. This includes court procedures (legal proceedings stricto sensu), but also actions that trigger procedural effects. A further analysis of this distinction can be found in previous literature [6,18]. In the paragraph at stake, we annotated as *event* the whole sentence, being its type *procedure*. Further, we have annotated events and their temporal dimension (related-time events) with concrete guidelines:

Extension of what event element. One *what* event element can also include two or more close-related verbs, e.g. "divorced" and "agree on custody", instead of annotating two connected verbs autonomously. Moreover, whenever there is some causal relationship between events, we annotate merely one, e.g. “they drink water and they felt unwell".

Repeated events. When there is reference to events happening in several dates (e.g. “the dates of birthday of three applicants, respectively“), we annotate solely one event as the *what*, and add just one annotation that covers all the related dates.

Non-dated events. Events that are not dated, though semantically expressing an implicit time reference, are then annotated under “when”, for example, the time expressions as “the same date”, “this afternoon”, “on unspecified dates”, “in a number of occasions”.

Non-annotated events. Some events were not considered relevant to be depicted in a timeline, and therefore not annotated, e.g. the fact that *X was born in X* seemed irrelevant.

Factuality. Events that are mentioned in the text but do not occur, are yet annotated with the feature “factuality”, but not included in the timeline. When events are negated, this feature equals to “NOT”, for instance, a party does not appeal against a decision.

Difficult and blurred annotations. During the annotation process, some events were difficult to tag, and others sparked discussion about how to do it, challenging the stipulated guidelines and evidencing how complex and subjective annotating tasks can be. Due to space constrains, we only show one sample annotation that triggered discussion on the type of events between procedure/circumstance. Further examples can be found in the corpus webpage. Regarding the paragraph *"On 26 February 2014 the Deputy Town Prosecutor carried out an inspection of remand prison SIZO-6"*, the issue relates to the semantics attributed to the role "Deputy Town Prosecutor" which renders the idea of being a court magistrate, and as such, it would be deemed as a procedural event. Herein, the function instead refers to an inspection task, without procedural effect.

4. Event extraction and named entity recognition

Herein we describe different methods used in our experiments for the extraction of events and their components in the ECHR court decisions. The applied approaches include deep-learning- and embeddings- based, conditional random fields and rule-based methods. The corpus and the code used in this paper is available on Github[2].

4.1. Deep learning

The task of assigning one or multiple classes from a set of classes to a text fragment is called text classification [1]. Fragments in our context are typically sentences that are classified into the types *procedure*, *circumstance* or neither. Hence we deal with a multi-class classification problem. The extraction of the event components is similar to a Named Entity Recognition Problem. We use a state-of-the-art model as a baseline and compare it further with additional approaches selected upon their results on legal texts (cf. [21,22, 23]). As there is no pre-trained legal model available, we apply the common approach

[2] `https://mnavasloro.github.io/EventsMatter/`

of *fine-tuning* a Universal Language Model for Text Classification (ULMFiT) [3] which takes a generic model and tunes it with a domain-specific corpus (called transfer learning). In terms of preprocesing, we remove very short sentences from the dataset, for instance headings such as *II THE LAW*. The models are:

Flair and Flair-finetuned. We first selected the generic *news-forward-fast* language model from the Flair embedding approach [24], which is pre-trained on a corpus with one billion words as our baseline model. We also fine-tune the pre-trained model with the documents from our corpus for one epoch (which took more than seven hours).

Flair ECHR. There are no specific legal pre-trained models available that we could use for our experiments. On a different classification task, we made good experiences in prior work with using a domain specific model trained on a small corpus of EU legal documents outperforming generic models in a multi-label text classification task [25]. Therefore, we also train a model on a corpus of 13,000 ECHR court decisions acquired from the European Court of Human Rights OpenData project [26] for four epochs.

BERT and BERT-finetuned. The Bidirectional Encoder Representations from Transformers (BERT) [27] is a language model learning the context of words in a bidirectional way and is applicable to many tasks. We use a BERT model (*bert-base-cased*) pre-trained on Wikipedia and a book corpus, plus further add a layer on top fine-tuning the model with the ECHR corpus for two epochs.

DistilBERT and DistilBERT-finetuned. DistilBERT [28] is a lightweight version of BERT that makes use of a teacher-student setup to distill the knowledge of the large model (BERT) to the student (DistilBERT). Our fine-tuned model (two epochs) is based on the pre-trained *distilbert-base-cased* model with an additional ECHR corpus layer.

4.2. Conditional Random Fields

Conditional Random Fields (CRF) are used for the mapping of sequences based on probabilistic models to label sequences [29]. CRF have already been applied in similar tasks in the legal domain for extracting specific legal entities, such as lawyers, courts and legal literature [30]. A CRF model uses features of a token, for instance casing, position of the token and subsequences, to calculate the probability that it is preceded or followed by a particular other token. It also takes the probabilities into account that a specific named entity, for instance a temporal information is followed by a subject.

4.3. Rule-based

Unlike the previous approaches, implemented as a classification task, the rule-based approach is an annotation task based on a search for specific patterns of events in the form of frames. Our approach has two steps: i. the collection of frames (done before the annotation), and ii. the event extraction that uses the frames in order to annotate a text.

1. Frame collection. We listed all *what* event components in the training set, and then identified the main verb, its type and the dependency relations (using the CoreNLP dependency parser [31]), within the *what*, and towards the subject (*who*), including the object for both possible active and passive voices since they are very different. When there are several mentions of the same main verb, all information is gathered and combined into a single frame. Once all the *what* elements are processed, they are stored for later use by the extraction algorithm.

2. Event extraction. Using the previously obtained frames, we look for the relevant events in the text. Since there are events that can appear many times in a text, we just consider events that have a date attached. To find dates and their normalized value (in order to be able to build a timeline), we adapted the Añotador software [32]. Then we used the information from the frames to look for the main verb of the event and for the previously identified dependency relations, as well as some Part-of-Speech considerations (using also CoreNLP). Additionally, some specific events that tend to appear always in the same form in the text (such as the final decision) are identified using regular expressions.

4.4. Use case: Timeline generation

In order to enable an intuitive way to overview a case, we decided to generate timelines from the case. We developed a demonstrator [3] that takes the *id* of a ECHR case and returns its rule-based annotation and generates a timeline. Through this timeline, we can navigate a case going directly to the event mention in the text just by clicking on it in the timeline. The fact that it directly refers to the text allows the user to retrieve the context of the event, as well as surrounding information that might not be reflected in the timeline.

5. Evaluation and Discussion

In this section we present results of our experiments. For experiments based on deep learning approaches, we used the state-of-the-art NLP library Flair[4] which uses contextualized string embeddings (called FlairEmbeddings) that captures the semantics and the context, and therefore, produce different context dependent embeddings for the same words [24]. The pre-trained transformer models (BERT, DistilBERT) are provided by the Huggingface library [33] and can be easily imported into Flair. The Flair ECHR model is created using the Flair library, and fine-tuning of the BERT and DistilBERT models is also based on the transformers library by Huggingface. All models have been trained with the same settings of a maximum of 150 epochs, patience of 3 and an anneal factor set to 0.5 and the training is automatically stopped when the learning rate is too small. We use common evaluation metrics: *Precision (P)*, *Recall (R)* and *F-score (F)*.

The documents have an average size of 2,302 tokens without the legal section (legal framework). Each document includes on average 21 different events, divided into 10 *procedure* and 11 *circumstance* events on average. The number of *who* occurrences amounts to 13.9 on average, while the number of temporal information annotations (*when*) to 17.6, and the number of *core* annotations to 24. We split the dataset into training, testing and validation set on a document level applying 5-fold cross-validation (in the deep learning based approach) such that the training set consists of 24, and the test and validation set of three documents each. The results represent the average of all splits. The results for all approaches are presented in Table 1. When comparing different approaches on event (component) extraction, we can observe that more advanced language models based on

[3] `https://whenthefact.oeg-upm.net/`

[4] `https://github.com/flairNLP/flair`

Table 1. Evaluation results for event classification and event components. (*P=Precision, R=Recall, F=F-score. Best results highlighted in boldface.*)

		Event Types		Event Components		
		Procedure	Circumstance	What	When	Who
CRF	P	82.39	68.78	85.10	89.30	89.09
	R	80.26	47.88	76.91	84.46	70.38
	F	80.80	54.78	**80.50**	86.58	78.34
Flair pretrained	P	83.32	57.21	56.41	90.50	89.93
	R	78.95	32.64	45.50	79.65	76.49
	F	80.31	40.57	50.10	84.35	82.30
Flair finetuned	P	87.07	58.88	60.12	90.87	91.63
	R	81.57	51.12	51.79	80.02	83.71
	F	84.13	53.33	55.58	84.87	87.44
Flair ECHR	P	76.78	41.93	57.94	82.00	40.48
	R	71.21	13.12	15.69	57.88	11.87
	F	73.86	17.92	23.28	66.88	18.23
BERT pretrained	P	81.95	66.70	60.45	85.88	86.37
	R	80.79	49.23	61.17	88.22	89.90
	F	80.56	54.31	60.78	86.98	88.05
BERT finetuned	P	91.44	76.81	65.58	89.45	88.88
	R	90.20	78.94	66.26	91.01	92.22
	F	90.55	77.59	65.83	**90.22**	**90.44**
DistilBERT pretrained	P	83.91	56.53	59.58	81.87	86.67
	R	83.57	51.63	57.45	86.35	85.73
	F	83.26	53.26	58.41	83.95	86.09
DistilBERT finetuned	P	91.64	81.61	62.79	87.31	89.92
	R	93.27	78.65	62.06	89.33	90.12
	F	**92.38**	**79.75**	62.37	88.23	89.98

		Event				Event Components					
		Identification		Type		What		When		Who	
		Len	Str	Len	Str	Len	Str	Len	Str	Len	Str
Rules	P	85.71	80.00	47.14	42.86	80.26	23.68	77.59	72.41	75.00	68.75
	R	77.92	72.73	42.86	38.96	69.32	20.45	63.38	59.15	63.16	57.89
	F	81.63	76.19	44.90	40.82	74.39	21.95	69.77	65.12	68.57	62.86

the transformer architecture [34] (BERT and DistilBERT), in general, provide a better result compared to the standard embedding models (Flair). Furthermore, we can see that the application of the ULMFiT approach to finetune generic language models, with a domain-specific corpus, leads to improved results between less than 1% (Flair pretrained to Flair finetuned for *who*) and 25% (DistilBERT for *circumstance*). The average increase in performance with fine-tuning is 8% for recognizing *procedure* and 21% for *circumstance* events, resp. The results of the CRF approach for the *what* component is unexpected, as it outperforms the more advanced methods by approximately 20%. The results for the extraction of the event components show that recognizing temporal information (*when*) of an event yields better results than the *what* of an event by 27% and the subject (*who*) by 21% (averaged over all approaches). The performance increase for the extraction of the event components of fine-tuned models, compared to generic models, is with 5% (what), 3% (when) and 4% (who) lower compared to the results for event types.

We see that results within the event type detection are within approx. 20% over all approaches, with the worst result being achieved by the Flair ECHR approach (F 73.86%), and the best result by the DistilBERT finetuned approach with an F-score of 92.38%. The results for the *circumstance* event type show a bigger spread between the worst result of the Flair ECHR approach with an F-score of only 17.92%, while the best result is achieved by DistilBERT finetuned (F 79.75%). For the *circumstance* event types we see generally lower results than for *procedure* type detection. We attribute this to the fact that the linguistic variety of the *procedure* events is narrower as they refer to a restricted set of ways of how to express them. The performance of the Flair ECHR model showed the least performance, due to being trained only on 13,000 ECHR documents, while it is common to train language models on much larger corpora to capture the basics of a language.

The performance differences between the *procedure* and *circumstance* event classes are evident with the latter results being worse by 29% on average. *Procedure* events capture formal processes throughout a legal trail and the ways to formulate the same events is somewhat restricted, for instance, *the court upheld the judgment*; in the description of the *circumstance*s of a case, however, the English language is potentially used in its entirety. Similarly, we observe the same behavior with the results for the event components with the results for *when* and *who* being better than the results for *what*. We attribute this to the fact that absolute temporal information (e.g. a date) contained in the court decisions under investigation always follows the structure of *Day Month Year*, and the number of acting subjects is also limited to a certain range of persons (eg. applicant, judge, prosecutor), authorities (eg. Supreme Court, housing authority) or things (eg. application, appeal). Relative temporal information (eg. *X days later*, *between X and Y* or *until X*) is also expressed in a few ways only.

Overall, we can say that fine-tuning an existing language model trained on a large corpus that captures the basic features of a language with a domain-specific corpus performs better than training a new language model with a rather small domain-specific corpus. Moreover, the more restricted the variety of class candidates for classification is, the better the results. The same applies to the information following a specific format, i.e. temporal information in the form of dates.

Regarding the rule-based approach, the evaluation is slightly different. While in the deep learning approach (first table) the number of named entities reflect the results of finding the event arguments *only* in those sentences where there is an event. On the contrary, the rule-based approach (second table) finds the events and the arguments in the same algorithm, so the results of the argument are contingent upon the event results. Additionally, we provide both *strict* and the *lenient* results, meaning that either the extent of our annotation match exactly to the one by the annotators or that it only overlaps (adding or omitting some words), resp. Also, the event evaluation includes finding the extent of the event, and then, over this finding, decide its type. The annotation and evaluations for the rule-based approach were done with the software GATE [35].

From the results of the rule-based approach we see that in the event finding task we got good results, both in the strict and lenient case, meaning that most of the events are correctly found and with the correct extent. Generally speaking, we identify about 4 out of every 5 relevant events, and additionally some that were not marked as relevant (although this does not mean they are not events). Regarding event types, the results for rule-based approaches are not very promising, mainly due to the fact that the same verb can often represent both circumstantial or procedural events, depending on surrounding information that the current rule-based implementation is not able to identify.

Results for detecting event arguments with the rule-based approach, on the other hand, are very different. While the *what* event component has very bad strict results, mainly due to the difficulty to determine the extent of the relevant modifiers of a verb, the *who* and the *when* show very good results, finding correctly most of them (e.g., 68.57% of the *who* taken into account that the limit was less than the 81.63% of the events) and almost always with the correct extent. The lenient results of the core, similar to the ones from the other arguments, demonstrates that besides the extent, the identification is correct.

6. Conclusions and Future Work

This paper presented a new corpus of legal decisions annotated with relevant events, along with a comparison of different approaches for the extraction of events and their components. Moreover, we tested state of the art methods to accomplish this annotation task automatically with promising results. To illustrate the utility of this task, we implemented an online timeline generation service which could be used by lawyers to get a quick overview of a case, thereby helping to turn decisions into operational, consumable and accessible legal knowledge.

To the best of our knowledge there is no previous comparison of event extraction techniques over legal texts in literature, and neither an available legal corpus annotated with events. In future work it would be interesting to validate the results with decisions from other courts such as the European Court of Justice or the United States Supreme Court, which are structured differently.

Acknowledgements. María Navas-Loro[5] work was partially supported by a Predoctoral grant from the I+D+i program of the Universidad Politécnica de Madrid. Sabrina Kirrane is funded by the FWF Austrian Science Fund and the Internet Foundation Austria netidee SCIENCE programme.

References

[1] Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv. 2002;34(1):1–47.

[2] Grishman R, Sundheim BM. Message understanding conference-6: A brief history. In: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics; 1996. p. 466–471.

[3] Howard J, Ruder S. Fine-tuned Language Models for Text Classification. CoRR. 2018;abs/1801.06146.

[4] Hogenboom F, Frasincar F, Kaymak U, De Jong F. An overview of event extraction from text. In: DeRiVE@ ISWC. Citeseer; 2011. p. 48–57.

[5] Xiang W, Wang B. A Survey of Event Extraction From Text. IEEE Access. 2019;7:173111–173137.

[6] Navas-Loro M, Santos C. Events in the legal domain: first impressions. In: TERECOM@JURIX; 2018. p. 45–57.

[7] The ACE 2005 Evaluation Plan.;. `https://api.semanticscholar.org/CorpusID:10821576`.

[8] Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, et al. Automating the extraction of rights and obligations for regulatory compliance. In: International Conference on Conceptual Modeling. Springer; 2008. p. 154–168.

[9] Maxwell KT, Oberlander J, Lavrenko V. Evaluation of semantic events for legal case retrieval. In: Proceedings of the WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval. ACM; 2009. p. 39–41.

[5] ORCID 0000-0003-1011-5023

[10] Lagos N, Segond F, Castellani S, O'Neill J. Event extraction for legal case building and reasoning. In: International Conference on Intelligent Information Processing. Springer; 2010. p. 92–101.

[11] Navas-Loro M, Satoh K, Rodríguez-Doncel V. Contractframes: Bridging the gap between natural language and logics in contract law. In: JSAI International Symposium on Artificial Intelligence. Springer; 2018. p. 101–114.

[12] Wyner AZ, Peters W. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. In: JURIX. vol. 10; 2010. p. 127–136.

[13] Etiene T, et al. Linea: Building Timelines from Unstructured Text. In: 28th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2015. IEEE Computer Society; 2015. p. 234–241.

[14] Fulda J, et al. TimeLineCurator: Interactive Authoring of Visual Timelines from Unstructured Text. IEEE Trans Vis Comput Graph. 2016;22(1):300–309.

[15] Styler IV W, et al. Temporal Annotation in the Clinical Domain. Transactions of ACL. 2014;2:143–154.

[16] Jung H, et al. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. In: Proceedings of BioNLP 2011 workshop. ACL; 2011. p. 146–154.

[17] Tannier X, Vernier F. Creation, Visualization and Edition of Timelines for Journalistic Use. In: Proceedings of Natural Language meets Journalism Workshop at IJCAI; 2016. .

[18] Navas-Loro M, Filtz E, Rodríguez-Doncel V, Polleres A, Kirrane S. TempCourt: evaluation of temporal taggers on a new corpus of court decisions. The Knowledge Engineering Review. 2019;34:e24.

[19] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. 2020;28(2):237–266.

[20] Registry of the Court. European Court of Human Rights; 2020. Accessed 2020-09-14. https://www.echr.coe.int/documents/rules_court_eng.pdf.

[21] Chalkidis I, Fergadiotis M, Malakasiotis P, Androutsopoulos I. Large-Scale Multi-Label Text Classification on EU Legislation. CoRR. 2019;abs/1906.02192.

[22] Shaheen Z, Wohlgenannt G, Filtz E. Large-scale legal text classification using transformer models. In: Semapro 2020; to appear.. .

[23] Tuggener D, von Däniken P, Peetz T, Cieliebak M. LEDGAR: a large-scale multi-label corpus for text classification of legal provisions in contracts. In: 12th Language Resources and Evaluation Conference (LREC) 2020. European Language Resources Association; 2020. p. 1228–1234.

[24] Akbik A, Blythe D, Vollgraf R. Contextual String Embeddings for Sequence Labeling. In: COLING 2018, 27th International Conference on Computational Linguistics; 2018. p. 1638–1649.

[25] Filtz E, Kirrane S, Polleres A, Wohlgenannt G. Exploiting EuroVoc's Hierarchical Structure for Classifying Legal Documents. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer; 2019. p. 164–181.

[26] Quemy A. European Court of Human Right Open Data project. CoRR. 2018;abs/1810.03115.

[27] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT (1); 2019. p. 4171–4186.

[28] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR. 2019;abs/1910.01108.

[29] Lafferty JD, McCallum A, Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Brodley CE, Danyluk AP, editors. Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001). Morgan Kaufmann; 2001. p. 282–289.

[30] Leitner E, Rehm G, Moreno-Schneider J. Fine-grained Named Entity Recognition in Legal Documents. In: International Conference on Semantic Systems. Springer; 2019. p. 272–287.

[31] Chen D, Manning CD. A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 740–750.

[32] Navas-Loro M, Rodríguez-Doncel V. Annotador: a temporal tagger for Spanish. Journal of Intelligent & Fuzzy Systems. 2020;39:1979–1991. 2.

[33] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv e-prints. 2019 Oct;p. arXiv:1910.03771.

[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.

[35] Cunningham H, et al. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLOS Computational Biology. 2013 02;9(2):1–16.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200848

Retrieval of Prior Court Cases Using Witness Testimonies

Kripabandhu GHOSH [b,1], Sachin PAWAR [a], Girish PALSHIKAR [a],
Pushpak BHATTACHARYYA [c] and Vasudeva VARMA [d]

[a] TCS Research, Tata Consultancy Services, Pune, India
[b] Indian Institute of Science Education and Research - Kolkata
[c] Indian Institute of Technology Bombay
[d] International Institute of Information Technology Hyderabad

Abstract. Witness testimonies are important constituents of a court case description and play a significant role in the final decision. We propose two techniques to identify sentences representing witness testimonies. The first technique employs linguistic rules whereas the second technique applies distant supervision where training set is constructed automatically using the output of the first technique. We then represent the identified witness testimonies in a more meaningful structure – event verb (predicate) along with its arguments corresponding to semantic roles A0 and A1 [1]. We demonstrate effectiveness of such representation in retrieving semantically similar prior relevant cases. To the best of our knowledge, this is the first paper to apply NLP techniques to extract witness information from court judgements and use it for retrieving prior court cases.

Keywords. Prior Case Retrieval, Witness Testimonies, Natural Language Processing, Semantic Roles

1. Introduction

Witnesses – whether prosecution or defence, lay or expert – are important in all types of court cases. Witness testimonies and their cross-examinations by the counsels have a significant effect on the judges' decision. Large corpora of court judgements (e.g., the Indian Supreme and High Court judgements), often contain the judges' summaries of the witness testimonies presented during the proceedings. In addition, judges often comment in the judgement on (a) the correctness, quality, completeness and reliability of the testimonies of a witness; (b) the interrelationships between the testimonies of various witnesses (e.g., consistency or contradictions); and (c) the impact ("weighing in") of various witness testimonies on their final decision. The specific contents of witness testimonies and such high-level analyses are valuable for preparing a case, retrieving relevant past cases, understanding strengths and weaknesses of a case, predicting court decisions and extracting legal argumentation.

In this paper, first, we propose two NLP techniques (linguistic knowledge-based and distantly supervised) to identify sentences of class Testimony, i.e., sentences containing

[1] Work was carried out when the author was in TCS Research, Tata Consultancy Services, Pune

testimonies of witnesses; example: `The body of Gian Kaur was sent to Dr. Singh (PW 6) for post-mortem who noticed five minor injuries on the body of the deceased.` Further, we extract details of events mentioned in such witness Testimony sentences. A witness testimony provides factual or subjective details about various events, objects and persons. We extract information provided by witnesses about various events, and not about the persons or objects, though the approach can be easily extended. We focus on two types of events: *crime events* or *legal events* such as filing a police complaint, or arrest. We restrict an *event* to mean a physical action or communication. We focus on events expressed as verbs, although nouns (e.g., `attack`) can also denote an event. We represent event information provided by a witness as an *event frame*, consisting of (i) the action verb, (ii) the agent who initiated the action, and (iii) the patient (or beneficiary) who experienced the action. Other event details (e.g., time, location), can be easily extracted. We use MatePlus [2], a semantic role labeling (SRL) tool, to identify the predicate and associated A0, A1 arguments (described in Section 2.2.1) and fill up event frames.

Finally, after extracting the event details from witness testimonies, we demonstrate its use for improving retrieval of relevant past cases (*prior cases*) based on high-level English queries which might be asked by a lawyer or a lay person. *Prior cases* form the backbone of judicial systems following Common Law; e.g., in India. For *prior case retrieval*, we propose two techniques. The first is based on exact matching of the event frames (one from the query and another from a past court judgement). The other is based on learning a representation for event frames and then using a similarity measure over event frames. We demonstrate that our approaches perform better in retrieving past court judgements as compared to three baseline methods: BoW retrieval function BM25, similarity over document representation vectors given by Doc2Vec, and Sentence-BERT. Doc2Vec has demonstrated efficacy in prior case retrieval [3] when the whole case document is considered. However, explainability of a prior case retrieval result remains an open question, which we attempt to address in this paper. Recently, [4] used supervised techniques for answering basic questions in legal domain using numerous features. Our proposed technique for prior case retrieval is completely unsupervised which handles fine-grained questions pertinent to a given case situation. To the best of our knowledge, this is the first paper to apply NLP techniques to extract witness information from court judgements and use it for assisting lawyers in an explainable manner.

2. Methodology

Our proposed technique works in two phases. In the first phase, we identify witness testimony sentences from prior court case documents using a set of linguistic rules and a distantly supervised LSTM-based sentence classifier. Then in the second phase, we retrieve prior court cases relevant to a query using two different matching techniques. Here, the queries are matched with only witness testimony sentences from the prior court cases and other sentences are ignored. We use a corpus of 30,034 Indian Supreme Court judgements from years 1952 to 2012, for all our experiments.

2.1. Identifying Witness Testimony Sentences

As there are no readily available annotated datasets for Testimony sentences, we use linguistic rules to create training data automatically. This technique works in two steps where in the first step, the *linguistic rules* are used to identify Testimony sentences as well as certain non-Testimony sentences with high confidence. In the second step, we employ *distant supervision*, where training data is automatically created using output of the first step. Here, we train a Bi-LSTM based sentence classifier to identify Testimony sentences.

2.1.1. Linguistic rules

Our linguistic rules are designed to ascertain that any sentence identified as Testimony sentence satisfies the following linguistic properties. Here, we use the spaCy [5] dependency parser to obtain dependency tree structure for each sentence.

R_1: Presence of explicit (e.g., `eye-witness`, `P.W.2`) or implicit witness mentions. Implicit mentions can be pronouns (`he`, `she`), person-indicating common nouns (`landlord`, `doctor`), or actual person names (`S.I. Patil`).

R_2: Presence of at least one statement-indicating verb like `stated`, `testified`, `narrated`.

R_3: Within its dependency subtree, the statement verb should contain at least one of the following: a clausal complement (*ccomp*) or open clausal complement (*xcomp*).

R_4: The statement verb should NOT have a child which negates it like `not`.

R_5: The statement verb should have at least one witness mention within its *nsubj* or *agent* dependency subtree (to ensure that the witness mention is subject/agent of the statement verb) but should NOT have any *legal role* (e.g. `lawyer`, `counsel`, `judge`) mention within its *nsubj* or *agent* dependency subtree (to exclude the statements by lawyers or judges).

The same set of rules are also used to identify non-Testimony sentences which are quite *similar* to Testimony sentences. Such sentences are those which satisfy the rules R_1 to R_4 but don't satisfy the rule R_5. Using the above rules, we identified 37572 Testimony sentences and 14382 non-Testimony sentences (see Table 1 for examples). In order to estimate the precision of our linguistic rules, we manually verified 200 random sentences identified as Testimony and the precision turned out to be 85%.

2.1.2. Distantly supervised Bi-LSTM based sentence classifier

As our linguistic rules are dependent on achieving correct dependency parsing, we observed that the rules fail to identify several Testimony sentences due to incorrect parsing. To overcome this, we trained a Bi-LSTM based sentence classifier which does not use

Table 1. S_1,S_2: Witness Testimony sentences identified by rules; S_3: Negative instance identified by rules for Testimony; S_4: Testimony sentence NOT identified by rules but identified by the Bi-LSTM based classifier.

S_1	`It must be noticed that P.W.-1 in his deposition stated that the appellant had taken him away in an ambassador car driven by P.W.-4 Rajib Bhuyan.`
S_2	`He further stated that the portion of the ground on which the grass was cut was shown to the Police Inspector.`
S_3	`The learned counsel stated that PWs 1, 2 and 3 must have come there to attack the appellants.`
S_4	`PW-15 further deposed that she knew Bharosa Colour Lab as she had been there several times to meet Mahesh.`

any dependency information but uses only the sequence information of the words. For training the classifier, we create the training dataset automatically by using our linguistic rules. 37572 Testimony sentences and 14382 non-Testimony sentences identified by the rules are treated as positive and negative instances, respectively. In addition, 23190 sentences are randomly selected from the rest of the corpus and treated as negative instances. Once the classifier is trained, we classify all the remaining sentences in the corpus and select 10000 sentences with highest confidence as Testimony sentences. In order to estimate the precision of our distantly supervised Bi-LSTM classifier, we manually verified 200 random sentences out of these 10000 and the precision turned out to be 75%. This classifier clearly learns more patterns over the rule based method (see Table 1). In Table 1, our rules fail to identify S_4 as a Testimony sentence because the dependency parsing fails to identify PW-15 as the subject of the verb deposed. However, our Bi-LSTM based sentence classifier correctly identifies this sentences as Testimony with high confidence.

2.2. Retrieving Relevant Prior Cases

In this section, we describe two matching techniques which are used to compute *semantic* similarity between a query and a witness Testimony sentence from a prior court case.

2.2.1. Background: Semantic Roles

A syntactic or grammatical structure (such as dependency or constituency parse tree) of a sentence does not always capture full *meaning* of a sentence. E.g., consider the two sentences "John broke the window." and "The window broke." Here, even if the syntactic role of "the window" is different in both these sentences (*object* in the first sentence and *subject* in the second sentence), the underlying *semantic role* of "the window" is same in both of these sentences. Semantic Role Labelling (SRL) of a sentence identifies *predicate-argument* structures in the sentence such as the examples shown in Table 2. These predicate-argument structures are shown in a format adopted by PropBank [1]. In PropBank, the arguments of a predicate are numbered as A0, A1, A2 depending on the semantic role it plays. For a particular predicate, A0 is generally an *Agent* (someone who initiates the action), while A1 is a *Patient* or a *Theme* (someone who undergoes the action). No such consistent generalizations can be made across different verbs for the higher-numbered arguments. Hence, in this paper, we only consider A0 and A1 arguments of verbal predicates in witness Testimony sentences and queries. Also, as we are only focussing on predicates corresponding to event verbs, we also refer to these predicate-argument structures as *event frames*. We use MatePlus [2], a semantic role labeling (SRL) tool, for obtaining predicate-argument structures present in each sentence.

Table 2. Examples of predicate-argument structures in PropBank[1] style. The A0 (Arg0) argument plays an agent semantic role and A1 (Arg1) plays a patient/theme semantic role.

S_1: P.W. 1 to 5 have stated that the appellant assaulted the deceased with a crow bar on his head.
Predicate: assaulted, **A0 (agent)**: the appellant, **A1 (patient/theme)**: the deceased
Q_1: Which are the cases where the appellant has attacked the deceased?
Predicate: attacked, **A0 (agent)**: the appellant, **A1 (patient/theme)**: the deceased

2.2.2. Exact Semantic Match (M1)

We leverage the predicate-argument structure (as elucidated in Table 2) in a query or a sentence in a candidate prior case in retrieval. We find the similarity of a query, Q (e.g., Q_1 in Table 2) with each sentence, S (e.g., S_1 in Table 2) in a candidate prior case document **D**. To this end, we match the predicate-argument structure of Q with that of S, where the corresponding predicate and arguments are matched. That is, the *Predicate* in Q is matched with the *Predicate* in S, the *A0* in Q is matched with the *A0* in S and so on. The similarity between Q and S is defined as:

$$SIM_s(Q,S) = \frac{\sum_r match(Q_r,S_r)}{|Q|} \tag{1}$$

Here, $r \in \{Predicate,\ A0,\ A1\}$ (semantic roles), $match(.) = 1$ if there is an *exact match*, 0 otherwise. $|Q|$ is the number of *not null* arguments in the query (some of the argument values may be *null* if not detected by the SRL tool). This is done to normalise the score over the query length so that incomplete matches are penalized. In our example, the $SIM_s(Q,S)$ is $\frac{2}{3} = 0.67$. In case of a complete match (if S contained `attacked` instead of `assaulted` as *Predicate*), the value of $SIM_s(Q,S)$ will be 1. We compute the similarity at the document level, i.e. between Q and D using: $SIM(Q,D) = max_S(SIM_s(Q,S))$ (maximum of all $SIM_s(Q,S)$ values for all sentences $S \in D$).

2.2.3. Semantic Match using Event Frame Representation (M2)

Finding exact match of predicate and arguments in a query event frame (predicate-argument structure) may not be possible always due to the usage of different but semantically similar words in relevant documents. In our aforementioned example, the high semantic similarity between Q and D is not realized even though `attacked` and `assaulted` share the same semantic context. Hence, we propose to learn an embedded representation for the complete event frame structure, i.e. $\langle predicate, A0, A1\rangle$. We train a de-noising autoencoder [6] by masking either predicate, A0 or A1 of an event frame at a time and trying to reconstruct the complete frame. We employ a simple architecture where the input layer accepts a vector (of 900 dimensions) which is a concatenation of 300-dimensional pre-trained word vectors corresponding to predicate, A0 and A1, where any one of these is masked by using a zero vector. The next layer is a fully connected dense layer of 300 dimensions. Finally, the output layer is again a 900-dimensional layer reconstructing the original concatenated vector corresponding to the complete frame. Once this autoencoder is trained, its encoder part (i.e. first two layers) is used to obtain embedded 300-dim representation of any event frame. The similarity in this case is calculated as:

$$SIM(Q,D) = max_S(cosine\_sim(Repr(Q), Repr(S))) \tag{2}$$

Here, $Repr(x)$ is the representation of a frame x. That is, we take $SIM(Q,D)$ as the maximum value of *cosine similarity* between the representations of Q and S over all the sentences $S \in D$.

Table 3. Comparative performance of all techniques. B1:BM25, B2:Doc2Vec, B3: Sentence-BERT, M1:Exact Semantic Match, M2:Semantic Match using Event Frame Representation; best values shown in bold.

Query	**R-Precision (R-Prec)**				
	B1	**B2**	**B3**	**M1**	**M2**
q1: Which are the cases where a husband has set his wife on fire?	0.13	0.00	0.50	0.63	0.63
q2: Which are the cases where the appellant has attacked the deceased?	0.21	0.10	0.24	0.28	0.45
q3: Which are the cases where the respondent killed the deceased?	0.00	0.00	0.0	1.00	1.00
q4: Which are the cases where the appellant demanded money?	0.06	0.13	0.0	0.56	0.75
q5: Which are the cases where the respondent has forged signatures?	0.00	0.00	0.25	0.75	0.75
q6: Which are the cases where the appellant accepted bribe?	0.00	0.00	0.17	0.33	0.50
q7: Which are the cases where an appointment was challenged?	0.14	0.14	0.00	0.43	0.57
q8: Which are the cases where an election was challenged?	0.08	0.31	0.08	0.38	0.46
q9: Which are the cases where the complainant was beaten by wife?	0.00	0.00	1.00	1.00	1.00
q10: Which are the cases where the respondent has admitted the charge?	0.00	0.00	0.00	1.00	1.00
Average over all queries	0.06	0.07	0.22	0.64	**0.71**

Query	**Average Precision (AP)**				
	B1	**B2**	**B3**	**M1**	**M2**
q1: Which are the cases where a husband has set his wife on fire?	0.13	0.00	0.54	0.70	0.89
q2: Which are the cases where the appellant has attacked the deceased?	0.10	0.06	0.09	0.28	0.51
q3: Which are the cases where the respondent killed the deceased?	0.00	0.00	0.17	1.00	1.00
q4: Which are the cases where the appellant demanded money?	0.03	0.07	0.02	0.56	0.76
q5: Which are the cases where the respondent has forged signatures?	0.05	0.00	0.17	0.95	0.62
q6: Which are the cases where the appellant accepted bribe?	0.02	0.00	0.10	0.33	0.43
q7: Which are the cases where an appointment was challenged?	0.04	0.05	0.00	0.43	0.63
q8: Which are the cases where an election was challenged?	0.01	0.15	0.04	0.38	0.50
q9: Which are the cases where the complainant was beaten by wife?	0.00	0.00	1.00	1.00	1.00
q10: Which are the cases where the respondent has admitted the charge?	0.00	0.00	0.00	1.00	1.00
Average over all queries	0.04	0.03	0.21	0.66	**0.73**

3. Experimental Evaluation

3.1. Dataset

Corpus: We use the Indian Supreme Court judgements from years 1952 to 2012 freely available at `http://liiofindia.org/in/cases/cen/INSC/`. There are 30,034 files containing 4,634,075 sentences and 134,329,128 tokens.

Queries: We selected 10 queries (shown in Table 3) each from different topic viz. domestic violence, homicide, forgery, corruption etc.

Ground Truth: As there is no publicly available ground truth for our queries, we use the standard *pooling technique* [7] for selection of candidate documents for annotation. We run several ranking models (including our own techniques) and select top 20 documents for each model to form a pool which we annotate manually.

3.2. Baselines

We compare our technique with the following baselines:

1. **BM25** [8] ($B1$): A popular term scoring model based on the "bag-of-words" assumption, i.e. it *does not* consider the relative ordering of the words in the query and documents. We use the default parameter setting of the model, viz. $k_1 \in [1.2, 2]$ and $d = 0.75$.
2. **Doc2Vec** [9] ($B2$): A popular neural model that offers representation ("embeddings") of a piece of text. This is a popular neural model that offers representations ("embeddings") of a piece of text (sentence, paragraph and document). It overcomes the drawbacks of the bag-of-words models by incorporating the relative ordering of words in a text in the embeddings. We use hierarchical sampling with skip-gram model for window length=5, min-count=1.
3. **Sentence-BERT** [10] (B_3): A recent technique for obtaining sentence embeddings using Siamese-BERT networks. We used their state-of-the-art pre-trained model `bert-base-nli-stsb-mean-tokens` to obtain sentence embeddings for sentences in both query and documents. We could not fine-tune this model because of unavailability of annotated sentence pairs (with labels indicating whether they are semantically similar or not) for our experiments. Finally, the documents are ranked as per the maximum cosine similarity obtained by any of its sentence with a query sentence.

FIRE 2019 AILA track [11] contained one of the tasks which focussed on identifying relevant prior cases for a given situation. However, although the task is similar to ours, the queries are quite verbose. This is in contrast with our task where the queries are simple sentences with single verbal predicates. Hence, our techniques are not readily applicable to the task in the FIRE 2019 AILA Track. However, we use baseline techniques BM25 and Doc2Vec which are used by the most of the participating teams in this track.

3.3. Results

We evaluate the baselines and the proposed method in standard IR evaluation setup consisting of the corpus, the queries and the ground truth.

We use the following evaluation measures to evaluate the performance of our techniques as well as the baseline techniques:

1. **Average Precision (AP)**: This incorporates the relative ranking order of relevant documents; combines the joint effect of Precision and Recall.

2. **R-Precision (R-Prec)**: Precision at R, the number of relevant documents [7]

The retrieval performance of the proposed methodologies, as compared with the baselines, are shown in Table 3. Only witness Testimony sentences of the court cases in the corpus are considered for all the retrieval experiments. The proposed methods, viz. Exact Semantic Match ($M1$) and Semantic Match using Event Frame Representation

($M2$) outperform the baselines for all the queries and in both the evaluation measures, by a considerable margin. $M2$ outscores $M1$ on most queries.

To evaluate the contribution of witness Testimony sentences, we considered complete documents for BM25 as against only witness Testimony sentences. BM25 could not find even a single relevant document within top 10 for all the queries, highlighting the need for focussing only on witness Testimony sentences. Hence, we run all the experiments considering only the witness Testimony sentences. To evaluate the contribution of our distantly supervised Bi-LSTM based classifier, we applied our technique using only those Testimony sentences identified by the linguistic rules. We observed that the AP of M2 reduced from 0.73 to 0.69, stressing the importance of additional Testimony sentences identified by the distantly supervised classifier.

3.3.1. Analysis of results

Explainability of results is of paramount importance for the credibility of the system in an application area like legal domain, if the system is to be used by experts. In our proposed solution, we use semantic roles that capture an event expressed in a query. E.g., in the query q1 (`Which are the cases where a husband has set his wife on fire?`) (in Table 3), the predicate-arguments are: Predicate: `set`, A0: `husband`, A1: `wife` which semantically captures an event and matches it with a prior case where a similar event has occurred e.g., `a husband has poured kerosene on his wife and set her on fire`, based on the similarity of the semantic argument structure. We believe, this imparts more transparency and interpretability of the results in addition to the accuracy of the same. The baselines are unable to capture such nuanced semantic representations of the underlying events in a query. Our technique M2 helps in retrieving documents even if there is no exact match of the argument values in a query. E.g., for the query q2 (`Which are the cases where the appellant has attacked the deceased?`), M2 is able to retrieve the document containing the sentence `P.W. 1 to 5 have stated that the appellant assaulted the deceased with a crow bar on his head.` Although we have not used any formal notion of explainability, the proposed predicate-argument structure (semantic role) based matching schemes are able to implicitly explain the semantic similarity of a query with a prior case. However, we look to induct explainability in a more principled way in future.

It was observed that some of the queries result in much lower AP and R-Prec scores than others. This is because some queries are more general (e.g., q8) i.e., having higher number of relevant documents than some other queries which are specific (e.g., q9). The evaluation scores are dependent on the number of relevant documents retrieved at top ranks and hence are affected by this general or specific nature of the queries.

4. Conclusions and Future Work

We proposed a novel method which identifies witness Testimony sentences in Indian Supreme Court documents, extracts predicate-argument structures (or event frames) of event verbs and leverages them for prior case retrieval. The proposed method outperforms standard baselines on fine-grained queries. We look to extend our experiments on a bigger dataset with more complex queries in near future.

References

[1] Palmer M, Gildea D, Kingsbury P. The proposition bank: An annotated corpus of semantic roles. Computational linguistics. 2005;31(1):71–106.

[2] Roth M, Woodsend K. Composition of word representations improves semantic role labelling. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 407–413.

[3] Mandal A, Chaki R, Saha S, Ghosh K, Pal A, Ghosh S. Measuring Similarity Among Legal Court Case Documents. In: Proceedings of the 10th Annual ACM India Compute Conference. Compute '17. New York, NY, USA: ACM; 2017. p. 1–9. Available from: http://doi.acm.org/10.1145/3140107.3140119.

[4] McElvain G, Sanchez G, Matthews S, Teo D, Pompili F, Custis T. WestSearch Plus: A Non-factoid Question-Answering System for the Legal Domain. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR'19. New York, NY, USA: ACM; 2019. p. 1361–1364. Available from: http://doi.acm.org/10.1145/3331184.3331397.

[5] Honnibal M, Montani I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://spacyio/. 2017.

[6] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[7] Manning C, Raghavan P, Schutze H. Introduction to information retrieval. Natural Language Engineering. 2010;16(1):100–103.

[8] Robertson SE, Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR94. Springer; 1994. p. 232–241.

[9] Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning; 2014. p. 1188–1196.

[10] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019. p. 3973–3983.

[11] Bhattacharya P, Ghosh K, Ghosh S, Pal A, Mehta P, Bhattacharya A, et al. Overview of the FIRE 2019 AILA Track: Artificial Intelligence for Legal Assistance. In: FIRE (Working Notes); 2019. p. 1–12.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200849

Generalizing Culprit Resolution in Legal Debugging with Background Knowledge

Wachara FUNGWACHARAKORN [a] and Ken SATOH [a]
[a] *National Institute of Informatics, Sokendai University, Tokyo, Japan*

Abstract. Since the legal rules cannot be perfect, we have proposed a work called Legal Debugging for handling counterintuitive consequences caused by imperfection of the law. Legal debugging consists of two steps. Firstly, legal debugging interacts with a judge as an oracle that gives the intended interpretation of the law and collaboratively figures out a legal rule called a culprit, which determines as a root cause of counterintuitive consequences. Secondly, the legal debugging determines possible resolutions for a culprit . The way we have proposed to resolve a culprit is to use extra facts that have not been considered in the legal rules to describe the exceptional situation of the case. Nevertheless, the result of the resolution is usually considered as too specific and no generalizations of the resolution are provided. Therefore, in this paper, we introduce a rule generalization step into Legal Debugging. Specifically, we have reorganized Legal Debugging into four steps, namely a culprit detection, an exception invention, a fact-based induction, and a rule-based induction. During these four steps, a new introduced rule is specific at first then becomes more generalized. This new step allows a user to use existing legal concepts from the background knowledge for revising and generalizing legal rules.

Keywords. legal reasoning, legal representation and algorithmic debugging

1. Introduction

Since we cannot codify every essential condition in the law, the law may still lack essential conditions which may later be revealed in a real-life case. This problem is also known in artificial intelligence as a *qualification problem* [1]. When we apply literal interpretation of such law to an exceptional case, it would lead to counterintuitive consequences, which cause absurdity or harm the public interest.

This paper focuses on Legal Debugging [2], which proposes on the detection of a cause of counterintuitive consequence called a *culprit* by asking users systematically, then it attempts to resolve a culprit. However, the previous work of Legal Debugging has encountered a problem that the result of resolution is too specific since it does not cooperate with background knowledge. Therefore, in this paper, we present the cooperation of Legal Debugging and external knowledge. We reorganized Legal Debugging into four steps. The first step is a culprit detection as described in [2]. The second step is an exception invention based on

Closed World Specification [3]. The third step is a fact-based induction based on V-operator described in [4]. The fourth step is a rule-based induction, in which the system cooperates with background knowledge and generalizes the culprit resolution using the same V-operator as in the previous step.

This paper is structured as follows. Sections 2 illustrates an example case used throughout this paper. Section 3 describes the legal formalization used in this paper. Section 4 explains four steps of Legal Debugging, including a step of rule-based induction that is the main proposal of this paper. Section 5 provides the discussion and future works. Finally, section 6 provides the conclusion.

2. Example

We adapted an example case from [5] as follows.

1. The plaintiff made a lease contract for his room between him and the defendant.
2. When the defendant returned home for a while, he let his son use the room.
3. Then, the plaintiff claimed that the contract was ended by his cancellation for the reason that the defendant subleases without permission.

The related piece of law in this case is Japanese Civil Code Article 612, which is stated as follows.

> **Phrase 1:** A Lessee may not assign the lessee's rights or sublease a leased thing without obtaining the approval of the lessor.
> **Phrase 2:** If the lessee allows any third party to make use of or take profits from a leased thing in violation of the provisions of the preceding paragraph, the lessor may cancel the contract.

From the literal interpretation of this article, the cancellation is valid. The third party in this case is the defendant's son and the defendant allowed his son to use the room without obtaining the approval of the lessor (the plaintiff in this case). However, it seems too harsh in this exceptional situation as the court decided as follows.

> Phrase 2 is not applicable in exceptional situations where the sublease does not harm the confidence between a lessee and a lessor, and therefore the lessor cannot cancel the contract unless they prove the lessee's destructing of confidence [6].

In this court decision, the court introduced the idea of destruction of confidence as an exception of Phrase 2 to prevent the counterintuitive consequence from the literal interpretation of Article 612.

3. Legal Formalization

3.1. Formalizing the law

One primary representation used for formalizing the law is to represent it into a logic program with negation as failure (later referred as a logic program) as in [7]. The logic program is defined as follows.

Definition 1 (Logic Program). *A logic program is a set of rules of the form:*

$$h : -b_1, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n. \tag{1}$$

where h, $b_1, \ldots, b_n$ $(1 \leqslant i \leqslant n)$ are first-order atoms and not presents negation as failure.

Sometimes, the rule is expressed in the form $h : -B$ where $B = \{b_1, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n\}$. For a rule R in the form (1), we denote the head h of the rule by $head(R)$; the positive body of the rule $\{b_1, \ldots, b_m\}$ by $pos(R)$; the negative body of the rule $\{b_{m+1}, \ldots, b_n\}$ by $neg(R)$; and the whole body of the rule B by $body(R)$. We express h. if the body of the rule is empty.

A first-order atom consists of a predicate and a set of arguments. a predicate begins with a lowercase and an argument is a variable (beginning with an uppercase) or a constant (beginning with a lowercase). A ground atom refers to an atom without any variable. A ground rule refers to a rule which contains only ground atoms.

We follow the previous work [2] to divide a predicate into two types: a rule predicate and a fact predicate. A rule predicate means a predicate that occurs at least once in a head of a rule while a fact predicate means a predicate that never occurs in a head of a rule. An atom with a rule predicate, called a rule atom, shall represent a legal consequence while an atom with a fact predicate, called a fact atom, shall represent a legal fact.

Table 1: An example of a logic program representing Article 612

```
cancellation_due_to_sublease(Lessor,Lessee) :-
        effective_lease_contract(Lessor,Lessee,Property),
        effective_sublease_contract(Lessee,Thirdparty,Property),
        using_leased_thing(Thirdparty,Property),
        manifestation_cancellation(Lessor,Lessee),
        not approval_of_sublease(Lessor,Lessee).

effective_lease_contract(Leaser,Lessee,Property):-
        agreement_of_lease_contract(Leaser,Lessee,Property),
        handover_lease_contract(Leaser,Lessee,Property).

effective_sublease_contract(Leaser,Lessee,Property):-
        agreement_of_sublease_contract(Leaser,Lessee,Property),
        handover_sublease_contract(Leaser,Lessee,Property).

approval_of_sublease(Lessor,Lessee):-
        approval_before_the_day(Lessor,Lessee).
```

Table 1 illustrates an example of logic program representing Article 612. It is adapted from the example described in [5]. From the example, we shall count `cancellation_due_to_sublease`, `effective_lease_contract`, and `effective_sublease_contract` as rule predicates where others are fact predicates.

3.2. Formalizing a case

Computational law researchers have been long interested in representing a legal case or a court decision. One popular way is to represent a legal case with a set of facts and a note that the plaintiff or the defendant won in such case [8,9]. However, we represent a legal case as a set of facts and a set of intentions since we focus on the consideration of legal consequence immediately before the judgement. Our case is formally defined as follows.

Definition 2 (Case)**.** *A case is a tuple* (X, V, I) *where* X *is a set of ground fact atoms refer to a fact situation of the case,* V *is a set of ground rule atoms that shall be valid and* I *is a set of ground rule atoms that shall be invalid (*V *and* I *are disjoint).*

Table 2: A set of legal fact representing the example case

```
agreement_of_lease_contract(plaintiff,defendant,room).
handover_lease_contract(plaintiff,defendant,room).
agreement_of_sublease_contract(defendant,son,room).
handover_sublease_contract(defendant,son,room).
using_leased_thing(son,room).
manifestation_cancellation(plaintiff,defendant).
father(defendant,son).

shall_be_invalid(cancellation_due_to_sublease(plaintiff,defendant)).
```

Table 2 illustrates a representation of example case in Section 2. The case around a ground fact atom in which a fact predicate never occurs in the program before. This ground fact atoms as *extra facts* e.g. `father(defendant,son)` in the example. Since the judge intended that cancellation due to sublease shall be invalid in this case, we note `cancellation_due_to_sublease(plaintiff,defendant)` in the set of ground rule atoms that shall be invalid.

4. Four Steps in Legal Debugging

4.1. Culprit Detection

The first step of the legal debugging is to detect a culprit. As discussed in [2] a *culprit* may be determined as a root cause of counterintuitive consequences. Counterintuitive consequences are defined as differences between literal interpretation of the law and the intended interpretation from the user. Since the intention may not be known in the first place, the system will ask the intention from the user until it finds a legal consequence that falls into two criteria of a culprit.

A *false-positive* culprit means a culprit that shall be valid but literally invalid. On the other hand, a *false-negative* culprit means a culprit that shall be invalid but literally valid.

Definition 3 (Intended Interpretation)**.** *An intended interpretation IM is an oracle and possibly infinite set of ground atoms representing an intended interpretation in the user's mind. We denote it by an oracle set since we cannot know the whole intended interpretation but for a case (X, V, I), we know that $X \subset IM$, $V \subset IM$ and I and IM are disjoint.*

Definition 4 (Support)**.** *Let IM be an intended interpretation. We say IM supports a ground rule atom p with respect to a program T if there is a rule in T that can be substituted into a rule in the form (1) such that $\{b_1, \ldots, b_m\} \subset IM$, $\{b_{m+1}, \ldots, b_n\}$ and IM are disjoint, and $p = h$. The substituted rule is called a supporting rule of p w.r.t. IM.*

Definition 5 (Culprit)**.** *Let IM be an intended interpretation. A ground rule atom p is a* culprit *with respect to IM and a program T if*

(i) $p \notin IM$ but IM supports p w.r.t. T (false-positive) or
(ii) $p \in IM$ but IM does not support p w.r.t. T (false-negative).

We follow the previous work [2] to trace down a sequence of counterintuitive consequences and a culprit will be one in the last of the sequence.

Table 3: An example of a culprit detection dialogue

```
Considering
cancellation_due_to_sublease(Lessor,Lessee) :-
        effective_lease_contract(Lessor,Lessee,Property),
        effective_sublease_contract(Lessee,Thirdparty,Property),
        using_leased_thing(Thirdparty,Property),
        manifestation_cancellation(Lessor,Lessee),
        not approval_of_sublease(Lessor,Lessee).

Shall effective_lease_contract(plaintiff,defendant,Property) be valid
    ? yes
Which Property? room.
Shall effective_sublease_contract(defendant,Thirdparty,room) be valid
    ? yes
Which Thirdparty? son.
Shall approval_of_sublease(plaintiff,defendant) be valid? no

Detect a culprit
cancellation_due_to_sublease(plaintiff,defendant).
With a supporting rule(s)
cancellation_due_to_sublease(plaintiff,defendant):-
    effective_lease_contract(plaintiff,defendant,room),
    effective_sublease_contract(defendant,son,room),using_lease_thing
    (son,room),manifestation_cancellation(plaintiff,defendant),not(
    approval_of_sublease(plaintiff,defendant)).
```

Table 3 illustrates an example of a culprit detection dialogue. The system asks a user whether there are some instantiation of rule atoms that shall be valid.

From this dialogue, we realize more members in the intended interpretation. As a result, we know that `cancellation_due_to_sublease(plaintiff, defendant)` is a culprit since it shall be invalid but the intended interpretation supports it.

4.2. Exception Invention

For a false-negative culprit, we may simply resolve by introducing a culprit. On the other hand, for a false-positive culprit, we require to invent a new predicate for an exception to rebut the supporting rule. This section describes how to invent a new predicate when the identified culprit shall be invalid. To this end, we apply Closed World Specification algorithm [3] as in Algorithm 1. It describes how to revise a logic program with negation as failure when we intend a ground atom A to be invalid. The algorithm states that if there is an exception in the supporting rule of A, we should use an instantiation of the exception; otherwise, we should invent a new atom with a new predicate for an exception.

Algorithm 1 An original Closed World Specification (CWS) algorithm

Input a logic program with negation as failure T with the unique stable model M and a ground atom A such that A is valid w.r.t. T but A is intended to be invalid.

```
for all supporting rule $R$ of $A$ w.r.t. $M$ and a substitution $\theta$ do
    if $body(R)$ contains $not\ b$ then
        Let $T' = T \cup \{b\theta\}$
    else
        Let $\{V_1, \ldots, V_n\}$ be the domain of $\theta$
        Let $q$ be a predicate symbol not found in $T$
        Let $b$ be $q(V_1, \ldots, V_n)$
        Let $T' = T\backslash\{R\} \cup \{head(R) : -(body(R) \cup \{not\ b\})\} \cup \{b\theta\}$
return $T'$
```

However, if we apply this algorithm to the example case, `approval_of_sublease(plaintiff,defendant)` is introduced. Such introduction is contradicted to the user intention that the `approval_of_sublease(plaintiff,defendant)` shall be invalid. From this reason, we may solve by forcing the algorithm to merely introduce an exception with a new predicate, as shown in Algorithm 2. Table 4 illustrates the exception invention in the example case. Now the system knows that the example case is an exceptional situation but what is a sufficient condition in the example case that makes the case exceptional would be determined in the next step.

Table 4: Exception invention in the example case

```
Inventing an exception using a closed world specification...
please specify a new exception name: new_exception.

The culprit is revised into

cancellation_due_to_sublease(Lessor,Lessee) :-
        effective_lease_contract(Lessor,Lessee,Property),
        effective_sublease_contract(Lessee,Thirdparty,Property),
        using_leased_thing(Thirdparty,Property),
        manifestation_cancellation(Lessor,Lessee),
        not approval_of_sublease(Lessor,Lessee),
        not new_exception(Lessor,Lessee,Property,Thirdparty).
```

Algorithm 2 An adapted Closed World Specification (CWS) algorithm

Input a logic program with negation as failure T with the unique stable model M and a ground atom A such that A is valid w.r.t. T but A is intended to be invalid.

```
for all supporting rule $R$ of $A$ w.r.t. $M$ and a substitution $\theta$ do
    Let $\{V_1, \ldots, V_n\}$ be the domain of $\theta$
    Let $q$ be a predicate symbol not found in $T$
    Let $b$ be $q(V_1, \ldots, V_n)$
    Let $T' = T \backslash \{R\} \cup \{head(R) : -(body(R) \cup \{not\ b\})\} \cup \{b\theta\}$
return $T'$
```

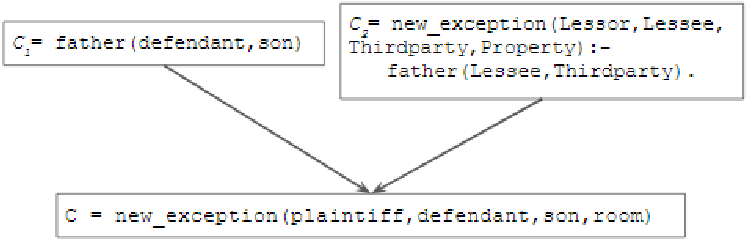
4.3. Fact-based Induction

In this step, we obtain the sufficient condition of why the present case is exceptional by asking from a user. Since we require to form a rule for describing the exceptional situation, the system would apply Inverse Resolution [4], to induce a new rule from known rules. Inverse Resolution is widespread applied for inductive programming, including for refining legal concepts in legal ontology [10]. However, there are some concerns about Inverse Resolution in Logic Program with Negation as Failure [11]. The first concern is that the result of Inverse Resolution is not generally consistent with the input program under the stable model semantics. We can only guarantee for some types of input programs e.g. input programs that are locally stratified and the dependencies of the input program are preserved in the result program. Since logic programs in legal representation are usually locally stratified, we have no problem with the first issue. For the second issue, one practical way is to take some extra facts into a body of a new rule to guarantee that we do not destroy dependencies of the input program. This corresponds to the practice in the law that the extra facts should be identified to distinguish the present exceptional case with the precedent. Another concern is that all variables in a body of a new rule should occur in a head of a new rule. It limits a new rule so that it is not too generalized.

Table 5: An example of fact-based induction dialogue

```
Generating a primary exception rule using Inverse Resolution
Listing possibly relevant facts...
1: agreement_of_lease_contract(plaintiff,defendant,room)
2: handover_lease_contract(plaintiff,defendant,room)
3: agreement_of_sublease_contract(defendant,son,room)
4: handover_sublease_contract(defendant,son,room)
5: using_leased_thing(son,room)
6: manifestation_cancellation(plaintiff,defendant).
7: father(defendant,son)
please specify relevant facts by a list of incremental indices (e.g.
    [1,3,5])
|: [7].

A new exception rule
new_exception(Lessor,Lessee,Property,Thirdparty):-father(Lessee,
    Thirdparty).
```



C_1= father(defendant,son)
C_2= new_exception(Lessor,Lessee,
Thirdparty,Property):-
father(Lessee,Thirdparty).
C = new_exception(plaintiff,defendant,son,room)

Figure 1. Illustration of applying V-operator to induce a new rule

A user would give the sufficient condition of the exceptional situation as the relevant facts and the system may check whether the set of relevant facts meets above criteria as a body of a new rule (e.g. the set must contain at least one extra fact). If the set passes the criteria, the system would apply the V-operator in Inverse Resolution to induce a new rule from a pair of ground atoms.

Definition 6 (Resolution). *Let C_1 and C_2 be two rules with no common variables. Let p be an atom within $pos(C_2)$ such that p is unifiable with $head(C_1)$ by the most general unifier (mgu) of θ. We denote the* resolvent *of C_1 and C_2 by $C = C_1 \cdot C_2$ where $C = head(C_2)\theta : -(body(C_2)\backslash\{p\})\theta \cup body(C_1)\theta$.*

Definition 7 (V-operator). *Given two rules C_1 and C, We call C_2 an induced rule by the V-operator from C_1 and C if $C_1 \cdot C_2$ is substitutable to C .*

Table 5 illustrates an example of fact-based induction dialogue. The systems ask the user to select a set of relevant facts. In the example, a user selects that fact that the defendant is a father of the third party, represented by `father(defendant,son)`, is the reason why this case is exceptional. Since the fact is an extra fact, it passes the criteria. Let C be `new_exception(plaintiff,defendant,room,son)`, a ground exception from the

Table 6: An example of fact-based induction dialogue

```
Would you like to generalize the rule more by using the background
    theory (y./n.) |: y.

Found more general rule
new_exception(Lessor,Lessee,Property,Thirdparty):-
    relatives(Lessee,Thirdparty).

Would you like to generalize the rule more by using the background
    theory (y./n.)
|: y.
Found no more general rule
```

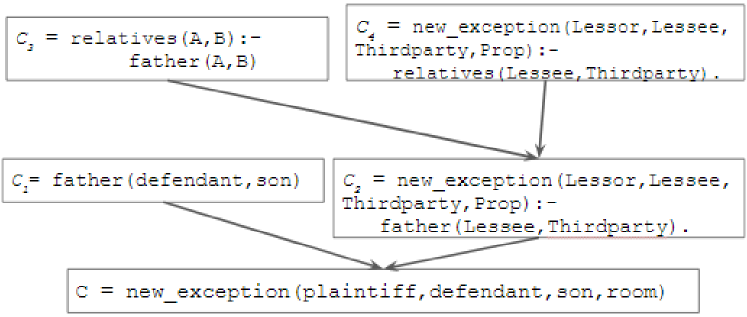

C3 = relatives(A,B):-
father(A,B)
C4 = new_exception(Lessor,Lessee,
Thirdparty,Prop):-
relatives(Lessee,Thirdparty).
C1= father(defendant,son)
C2 = new_exception(Lessor,Lessee,
Thirdparty,Prop):-
father(Lessee,Thirdparty).
C = new_exception(plaintiff,defendant,son,room)

Figure 2. Illustration of applying V-operator to induce a new rule

exception invention step by the adaption of Closed World Specification Algorithm; and let C_1 be `father(defendant,son)`, the reason given by the user, the system induce a new rule by the V-operator as in Fig. 1. An induced rule is more generalized than the ground exception from the previous step since an induced rule by the V-operator does not specifically apply to the example case. From the example, the system knows that the sufficient condition to make a case exceptional is when the lessee is the father of the sublessee.

4.4. Rule-based Induction

Beyond the primary induced rule, in this newly introduced step, the system may apply Inverse Resolution further with background knowledge. For ease of exposition, we assume that the background knowledge is convertible to a logic program called a background theory. This background theory is assumed to contain general knowledge rules as well as legal knowledge rules. For example, the background theory may contain a rule “A father is a kind of relative”, which is represented as `relative(X,Y) :- father(X,Y)`.

Table 6 illustrates an example of rule-based induction dialogue. If a user would like to generalize a rule induced in the previous step, the system would find a rule in a theory such that it can induce more general rule using the V-operator. From the dialogue, the system found a rule C_3 `relative(X,Y) :- father(X,Y)`. The

V-operator induces a new rule C_4 from C_2 (from the previous step) and C_3 as in Figure 2. The result rule C_4 implies that a new exception may be executed if the lessee is a relative of the sublessee.

Since a revision is only an advisory, the user can reject the generalization, accept the generalization, or request the system to generalize a rule further. The system may use other cases with intention to determine whether the generalized rule is acceptable.

Another operation that has not been mentioned in the example is W-operator [4]. W-operator is simply a combination of two V-operators back-to-back. It may be used for grouping similar concepts into the new concept. For example, suppose we know that a new exception should be valid not only for a case such that the lessee is a relative of the sublessee but also for a case such that the lessee is a working colleague of the sublessee. With W-operator, these two concepts may be grouped into a new concept, that covers a case such that the lessee is an acquaintance of the sublessee.

5. Discussion and Future Works

This paper is in line with a previous study [12] suggesting the benefit of background knowledge in computational law. Since we assume the legal rules and cases are formalized using first-order predicates, Legal Debugging has not yet supported open-texture concepts, which shows that a qualification problem still exists in our formalization. Another limitation of the proposed method is that a case which causes counterintuitive consequences is presumed to contain an extra fact describing the exceptional situation of the case. Since the V-operator used in the proposed method supports only one extra fact to induce each rule, we think that potential future works are extending the V-operator to support multiple extra facts, obtaining practical extra facts, or combining the facts already existed in legal rules with extra facts.

6. Conclusion

This paper describes the reorganization of Legal Debugging into four steps, namely a culprit detection, an exception invention, a fact-based induction, and a rule-based induction. These steps generalizes the resolution of a culprit by using Closed World Specification and Inverse Resolution. The rule-based induction, which is firstly introduced in this paper, can obtain more general rules for resolving a culprit by cooperating with background knowledge in a form of background theory. With such cooperation, the resolution can obtain more general normative facts to resolve a culprit in a more practical way. In future, we would like to investigate the acquisition of extra facts, the compliance of multiple extra facts, and the combination of extra facts and facts that already existed in legal rules.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470. We would like to thank Tiago Oliveira and anonymous reviewers for their extensive comments.

References

[1] M. Thielscher, The qualification problem: A solution to the problem of anomalous models, *Artificial Intelligence* **131**(1–2) (2001), 1–37.

[2] W. Fungwacharakorn and K. Satoh, Legal debugging in propositional legal representation, in: *JSAI International Symposium on Artificial Intelligence*, Springer, 2018, pp. 146–159.

[3] M. Bain and S. Muggleton, Non-monotonic learning, *Inductive logic programming* **38** (1992), 145153.

[4] S. Muggleton and W. Buntine, Machine invention of first-order predicates by inverting resolution, in: *Machine Learning Proceedings 1988*, Elsevier, 1988, pp. 339–352.

[5] K. Satoh, M. Kubota, Y. Nishigai and C. Takano, Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming, in: *Proceedings of the 2009 Conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference*, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009, pp. 162–171. ISBN ISBN 978-1-60750-082-7.

[6] 1994 (O) 693, Tokyo High Court No. 9 at 2431, Minshu Vol. 50, 1996.

[7] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory, The British Nationality Act as a logic program, *Communications of the ACM* **29**(5) (1986), 370–386.

[8] V. Aleven, Teaching case-based argumentation through a model and examples, PhD thesis, University of Pittsburgh, 1997.

[9] E.L. Rissland and K.D. Ashley, A case-based system for trade secrets law, in: *Proceedings of the 1st international conference on Artificial intelligence and law*, 1987, pp. 60–66.

[10] M. Kurematsu, M. Tada and T. Yamaguchi, A legal ontology refinement environment using a general ontology, in: *Proceedings of Workshop on Basic Ontology Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence*, Vol. 95, 1995.

[11] C. Sakama, Some properties of inverse resolution in normal logic programs, in: *International Conference on Inductive Logic Programming*, Springer, 1999, pp. 279–290.

[12] V. Aleven, Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment, *Artificial Intelligence* **150**(1–2) (2003), 183–237.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200850

Transformers for Classifying Fourth Amendment Elements and Factors Tests

Evan GRETOK [a] David LANGERMAN [a] and Wesley M. OLIVER [b]

[a] *Department of Electrical and Computer Engineering, Swanson School of Engineering, University of Pittsburgh, Pittsburgh, PA, USA*
[b] *Duquesne University School of Law, Pittsburgh, PA, USA*

Abstract. Determining if a court has applied a bright-line or totality-of-the-circumstances rule for Fourth Amendment cases demonstrates a difficult problem even for human lawyers and justices. Determining the type of test that governs an issue is essential to answering a legal question. Modern natural language processing (NLP) tools, such as transformers, demonstrate the capacity to extract relevant features from unlabelled text. This study demonstrates the effectiveness of the BERT, RoBERTa, and ALBERT transformer models to classify Fourth Amendment cases by bright-line or totality-of-the-circumstances rule. Two approaches are considered in which models are trained with either positive language extracted by a domain-expert or with full texts of cases. Transformers attain up to 92.31% accuracy on full texts, further demonstrating the capability of NLP techniques on domain-specific tasks even without handcrafted features.

Keywords. bright-line rule, totality-of-the-circumstances, fourth amendment, elements, factors, text classification, transformers

Introduction

To conduct legal reasoning, machines using artificial intelligence (AI) will have to identify the criteria the law uses to resolve an issue and extract evidence supporting those criteria. AI will also have to determine what the law does with those criteria to determine what sort of legal test is being used. In this paper, the authors show that, at least in the context of search and seizure law, it is possible for an automated system to examine a judicial opinion and identify the type of test used.

This research makes several contributions in determining the effectiveness of current natural language processing (NLP) systems to perform binary classification between bright-line and totality-of-the-circumstances rules, an important distinction in US criminal law. The authors perform transfer learning on several transformer models to extract meaning from the text. Models are fine-tuned on either key positive language extracted from cases by a domain expert or on the full text of the cases processed in a sliding-window approach. The accuracies of these models are compared with consideration to model size and complexity. The extraction of relevant language representation from full texts and successful classification of cases demonstrates the capability of current NLP systems to satisfy this need.

1. Background

This section outlines key concepts and related work. The legal background of bright-line and totality-of-the-circumstances rules is presented, key applications of AI to the law are revisited, and the technical aspects of transformer models are detailed.

1.1. Bright-Line and Totality-of-the-Circumstances Rules

Identifying the type of legal test a court is using is a fundamental question in resolving a legal issue. In a generic sense, legal tests are identified as either factors tests or elements tests. In an elements test, one seeking to obtain a legal remedy must satisfy all of the elements. In a factors test, a court will weigh the extent to which each of the factors is present, with the presence or absence of none of the factors being essential to resolve the issue either way [1].

Consider an elements test that requires a litigant to demonstrate *a*, *b*, and *c*. If there is no evidence on element *b*, then the litigant fails. It is much easier to resolve an issue governed by an elements test than one governed by a factors test. If a court is to consider factors *a*, *b*, and *c*, the absence of any support for factor *b* does not resolve the question. Similarly, evidence of *a*, *b*, and *c* would be sufficient to resolve an issue governed by an elements test, but not a factors test. Elements are either present or absent, factors must be weighed. Resolving a factors test is not beyond the capacity of a machine [2], though elements tests have proven easier for computers to analyze [3]. Regardless of the ease of the type of test, the machine must be able to deduce the type of legal test at issue.

Fourth Amendment cases were chosen because there are two types of clues in judicial opinions as to which sort of test the court is using. In other contexts, the choice between a test that considers factors or elements often is not driven by policy considerations and thus there is less likely to be professional commentary on the type of rule a court chooses to apply. In the Fourth Amendment context, a court's choice between a bright-line or totality-of-the-circumstances rule is very much a part of the discussion of academic commentators [4]. This constitutional provision governs searches and seizures. Bright-line tests provide clarity for police officers conducting investigations, but they also amount to judicially-created rule for the management of police. Totality tests defer to police departments for policy but provide little insight on what a court will find acceptable. In the case of New York v. Belton, the court concluded that if an officer had probable cause to arrest a motorist, the officer could search the entire car incident to arrest [5]. This is very simply an elements test with a single element. *Belton* did not ask whether the defendant was being arrested for a crime likely to yield evidence when the car is searched.

Belton demonstrates, however, that classifying a legal test is often a complicated question. There is a totality-of-the-circumstances test embedded within the Belton bright-line test. An officer must have probable cause to arrest a motorist. Probable cause is, of course, a totality-of-the-circumstances test. [6]. *Belton* is nevertheless regarded as a case that creates a bright-line rule. The Supreme Court was asked to consider whether a lawful arrest was sufficient to search the interior of a car, and the court determined that the right to search the interior of a car *always* accompanies the right to arrest a motorist.

There is an additional caveat complicating the classification of cases into one of these two groups. There are times when courts claim to be conducting a totality-of-the-circumstances test but are regarding a small set of facts that are likely to recur to clearly resolve the issue. Practically speaking, then, a commentator may label a test to be a bright-line rule while a court claims to be conducting a totality-of-the-circumstances analysis. Navarette v. California [7] is such an example. In *Navarette*, an anonymous informant reported improper driving and a police officer pulled the suspect over to ensure he was not drunk. Previously the Supreme Court had held in Florida v. J.L. that an anonymous tip that a person was possessing a gun was insufficient to detain the person identified [8]. Justice Thomas, speaking for the majority in *Navarette*, claimed to be applying a totality-of-the-circumstances test. Justice Scalia, who rarely disagreed with Justice Thomas, dissented, claiming that the majority had not been applying a totality-of-the-circumstances test at all, but rather creating a bright-line rule that anonymous tips of drunk driving were always sufficient to justify a stop [7]. For computational purposes, these cases would be identified as totality-of-the-circumstances cases even when, as a practical matter, a smaller number of commonly occurring factors prove to conclusively resolve the issue [9].

1.2. Basic Approaches to AI and Law

AI has become a mainstay of the legal profession. The ability for a computer model to process thousands of legal documents in minutes has reduced cost and fostered a new field of research. The field of AI and law has many facets, most of which lie in three areas. The first area is using AI to parse large corpora of text for relevant named entities [10], passages [11], or case law and statutes [12]. The second area is using AI to predict outcomes or behavior. This can take many forms such as predicting outcomes of court cases [13][14]. This area also includes the controversial topic of using AI to predict recidivism [15]. The final area is legal question answering and legal expert systems, where a large body of documents is used to train an AI to either directly answer questions or indicate logical paths of legal reasoning in search of fallacies or defenses [16].

Most approaches that filter or classify text rely on classical machine-learning (ML) methods that quantify some relationship of word or token frequency (i.e., bag of words representation) with a resultant label. This is done through the process of count vectorization, where a document is transformed into an embedding vector that uses unique words or n-grams as dimensions and their frequency of occurrence as the values for each dimension. These embedding vectors are then used as input to various ML models for prediction, such as a support vector machine (SVM), multi-layer perceptron (MLP), or decision tree (DT).

1.3. Deep Learning for Natural Language Processing

Recent NLP methods leverage an attention mechanism known as a transformer. Devlin describes the Google AI Language Lab's Bidirectional Encoder Representations from Transformers (BERT) model [17]. The effectiveness of this approach is its consideration of input sentences bidirectionally. This approach is borrowed from Vaswani in [18]. BERT requires no handcrafting of features and is able to extract meaningful representations directly from unlabeled text.

Liu presents a Robustly Optimized BERT Approach (RoBERTa). This research improves the training process of BERT and optimizes it via dynamic sentence masking.

Rather than training on recurrences of a sentence with a mask over a single word, the mask is moved to different words between training epochs. This allows the model to develop improved understanding of sentence structure and parts of language. The improvement was such that RoBERTa overtook BERT in capability for language understanding and question answering tasks [19].

Despite the capabilities of these models, each of them have on the order of a hundred million parameters and require many billions of operations to process texts. ALBERT, introduced in [20], attains similar accuracy at up to 18× lower parameter counts, as shown in Table 1, with a 1.7× reduction in training time. This was accomplished primarily by removing the dependency of the hidden layer and word embedding sizes and sharing parameters between layers. Larger variants of ALBERT, while still smaller than BERT, were able to attain a new state of the art on many of the same NLP benchmarks.

Table 1. Model Parameters and Layers [20][21]

Model	Variant	Parameters	Layers	Hidden Layer Size
BERT	Base Uncased	108M	12	768
BERT	Large Uncased	334M	24	768
RoBERTa	Base	125M	12	768
RoBERTa	Large	355M	24	768
ALBERT	Base v2	12M	12	768
ALBERT	Large v2	18M	24	1024

Transformer models have revolutionized deep learning for NLP. Their ability to capture relationships between distant segments of text helps them excel at complex tasks. Transformers have been used to expand the state of the art in benchmarks such as the Stanford Question Answering Dataset (SQuAD) [22], which asks an AI model to take an SAT-like test. Another challenging benchmark is reading comprehension, where an AI is asked to answer questions about a passage of text [23]. Transformer models consistently outperform the state of the art in these difficult tasks. In the law domain, transformers have been employed in recent work for judgement prediction [14], case law entailment [24], and legal news retrieval [25]. As the employed transformer models are limited to texts of up to 512 words, previous works consider hierarchical constructs of models for larger passages [14]. In many cases, this method is no longer required. A sliding-window approach to training existing transformer models on large text datasets can be enabled and customized with stride length parameters in the SimpleTransformers library [21].

2. Approach

This section details the experimental steps taken to make this research a reality. Key components include case preparation and model training.

2.1. *Preparation of Cases*

This experiment began with cases in the United States Supreme Court (SCOTUS) decided since 1946 [26]. WestLaw noted 880 Fourth Amendment cases decided by the US Supreme Court. A subset of these cases was identified in the literature as using or creating a “bright-line rule” or “totality-of-the-circumstances test.” Various law review articles described Fourth Amendment cases as fitting into one of the two models [27]. Unfortunately, The legal literature identified only a relatively small subset of the SCOTUS corpus as creating one of these two types of legal tests for interpreting the Fourth Amendment. Cases outside SCOTUS identified in the legal literature were therefore added to the dataset. The characterizations of the cases in the academic literature were accepted except when the literature took issue with the test courts claimed to be using [9]. In total, the dataset included 195 cases, 112 totality and 83 bright line.

The third author, a domain expert, then identified all case language deemed relevant to the court's analysis of the type of rule applied in or created by the case. An extensive inter-annotator agreement study was conducted in which each case was triply confirmed as bright line or totality by two independent legal citations and the opinion of the resident expert. Inter-annotation citations as well as positive and negative language from a small selection of cases can be referenced in Table 2. Finally, the full text of each case was extracted. Initial tests showed that roughly 200 cases was the minimum effective corpus size required for convergence in training. Hundreds of additional cases are available, but the time and expertise required to annotate data reduced labelling scope.

Table 2. Inter-Annotator Agreement with Key Positive and Negative Language

Case	Sources	Positive Language	Negative Language	Class
Ohio v. Robinette	[28] [29]	Voluntariness is a question of fact to be determined from all the circumstances.	...we have consistently eschewed bright-line rules...	Totality
Thornton v. United States	[30] [31]	Once an officer determines that there is probable cause to make an arrest, it is reasonable to allow officers to ensure their safety and to preserve evidence by searching the entire passenger compartment.	This determination would be inherently subjective and highly fact specific, and would require precisely the sort of ad hoc determinations on the part of officers in the field and reviewing courts that Belton sought to avoid.	Bright Line
Florida v. Royer	[32] [33]	All circumstances must be considered to determined whether someone is detained	We do not suggest that there is a litmus-paper test for distinguishing...	Totality
Alabama v. White	[34] [35]	We conclude that under the totality of the circumstances...	The Court there abandoned the "two-pronged test"	Totality
New York v. Belton	[4] [36]	A single, familiar standard is essential to guide police officers...	A custodial arrest of a suspect based on probable cause is a reasonable intrusion under the Fourth Amendment...	Bright Line

Preprocessing was conducted to prepare the text, enumerate classes, and split data into training folds. Texts were converted to lowercase as initial tests found improved accuracy without concern for proper nouns. This was also thought to minimize the effect of the relatively small dataset on the large transformer models. The effects of named-entity recognition were not considered in this research.

2.2. Inter-Annotator Agreement

Inter-annotator agreement was calculated to verify dataset validity. Cohen's kappa (κ) is the typical measure of agreement used to quantify how likely a dataset's distribution of agreement came about by chance rather than by true differences in the data. More on κ and how it is calculated can be found in the original paper by Cohen [37].

Table 3. Metrics Used to Calculate Cohen's Kappa for Inter-Annotator Agreement

Source A/Source B	Totality	Bright Line
Totality	106	0
Bright Line	12	77

$p_e = 0.509$, $p_O = 0.938$, $\kappa = 0.875$

As shown in Table 3, this dataset has a κ of ~0.87, implying near perfect inter-annotator agreement across all cited cases as to whether each represents totality-of-the-circumstances or bright-line rule.

2.3. Model Training

The SimpleTransformers library was employed for ease-of-access to pretrained BERT, RoBERTa, and ALBERT models [21]. The specific pretrained models bert-base-uncased, bert-large-uncased, roberta-base, roberta-large, albert-base-v2, and albert-large-v2 were fine-tuned in this study [38]. Transfer learning was conducted with three-fold cross validation to ensure effective fine-tuning on the full distribution of data. Three-fold was chosen as a 67% training set was sufficient for convergence but reduced the total training time. Training was conducted for twenty epochs for each model variant. Separate training rounds were considered for both domain-expert extracted positive language and full-text cases where the transformers were tasked with extracting language automatically. Training was accelerated on an NVIDIA TITAN RTX GPU using the Apex library. Final accuracy was fairly sensitive to initialization and dataset split, but this was expected due to the relatively small size of the dataset. For reference, the size of the combined full-text and positive language dataset used for transfer learning in this study is roughly ~4.3 MB of text, whereas a typical NLP dataset to train models of this size from scratch can range from tens of gigabytes to multiple terabytes [39] [40]. Accuracy referenced in the results section was computed as the mean of the F1-scores for each class.

The SimpleTransformers library provided a large number of parameters to tune the transfer-learning process. Do_lower_case was set to true as the dataset text was lowercased in preprocessing. FP16 precision was left enabled by default to increase training speed. To ensure full-text cases fit within the maximum 512 word model sequence length, the sliding_window parameter was set to true. The stride parameter was kept at its default 0.8. Default training and testing batch sizes of eight were used. Default values were used for the Adam optimizer epsilon, the learning rate, and the warmup ratio.

3. Results

This section details key results from this study, particularly the accuracy for each of the tested models. The authors' interpretation of these results follows.

3.1. Accuracy

Model accuracy results were determined for transformer base and large model variants on both domain-expert extracted positive language and full-text trials. These can be referenced in Table 4. Simple ML methods trained with the same positive-language and full-text datasets are included as a baseline for comparison.

Table 4. Model Accuracy (F1 Score)

Model	Variant	Positive Language	Full Text
Majority	Totality	62.00	62.00
SVM	sklearn	87.00	64.00
MLP	sklearn	88.00	76.00
Decision Tree	sklearn	69.00	56.00
BERT	Base Uncased	89.23	**92.31**
RoBERTa	Base	87.18	90.26
ALBERT	Base v2	87.69	91.79
BERT	Large Uncased	86.15	91.79
RoBERTa	Large	**90.26**	78.97
ALBERT	Large v2	81.03	57.44

These results demonstrate that transformer models can perform high-accuracy classification of cases by both positive language and full text. For previous research, the process of handcrafting the dataset was often an essential step. These results show that, while domain-expert extracted positive language may yield good accuracy, transformers enable full-texts provide an even better result. Simple ML methods simply cannot compete on full texts. The transformer models are able to extract adequate feature representations from the text on their own without human intervention. This showcases the value of using modern NLP techniques for this and similar problems in the legal domain.

The smaller transformer models performed very consistently at around 88% accuracy for positive language and 92% for full texts. In many cases, the reduced parameter counts, training times, and inference costs of the base models may make them a more attractive solution. The large models are considerably less consistent and often perform worse. For this study, one factor may be the relatively small dataset. The large model variants contain many more parameters to tune. Without a large dataset exercising these parameters during the transfer-learning process, the model becomes data starved and is not as effective. Overfitting may also take place if large models are allowed to train for too long, though further investigation is necessary to see if a double-descent phenomenon is presenting itself in this case [41].

In comparing transformer model types, differences again arise when considering the larger variants, especially for full texts. The reduction in performance for the large RoBERTa model is perhaps best attributed to the lack of cased data. While BERT has an uncased model, which was used in this study, RoBERTa does not. Some components of the pretrained model that applied specifically to cased data may never have been activated or utilized here, reducing the model's overall effectiveness. ALBERT may suffer from the opposite effect. As a smaller, more optimized model, it may simply not have the parameter space required to correctly filter the barrage of features from the full-text cases. Embedding is the key component for this task, and the methods used to reduce size and compute expense for ALBERT have reduced its capability here. Even larger ALBERT variants, such as ALBERT-XXLarge, may be able to overcome the simpler embedding limitations, but were unfortunately beyond the scope of this research.

3.2. Analysis

Proper transfer learning of transformers is about much more than just the quantity of samples. The quality of data fed to the model for training is a considerable factor. It is a double-edged sword. If good text is supplied and an accurate representation is extracted quickly, additional training epochs may result in degradation of the model and loss of accuracy. Conversely, if the text supplied is poor in representation or limited in scope, the model may struggle with extracting a representation at all, resulting in alarmingly poor metrics for the same training period and parameters.

While this study attained high peak accuracy and saw convergence of nearly all model types, the accuracy between training runs varied considerably in some cases. This is likely due to the variance in the amount and quality of language between different cases. Three-fold cross validation reduced this variance by ensuring that the full dataset could be used in each training round. The results of training are understandably more biased by the presence or absence of proper language resources in training than the number of cases in the sample for each class. Proper balance of the training dataset for abstract text classification tasks is critical, especially when training large models with a relatively small dataset. Even slight bias may lead to overfitting and a decrease in testing accuracy for the underrepresented class. This was experienced in many of the sub-par accuracy transformer training rounds. With such a high number of interrelated parameters to tune, biases in the subset of the small dataset selected for training quickly become apparent. This caused a small number of training rounds to fail to converge. The authors emphasize that these results demonstrate a proof of concept. For this approach to be employed in a production tool, a larger dataset would have to be prepared.

4. Conclusions

This research demonstrates that an automated system can be taught to identify whether a court is using or creating an elements or factors test. Reproducing this experiment in other substantive areas will of course require some modification to these methods. Outside the Fourth Amendment context, there is less discussion, in judicial opinions or the academic literature, about the choice between a multi-factor totality-of-the-circumstance test and a clearer test that turns on whether a small set of criteria are fully satisfied or not. Fourth Amendment cases will therefore include more language than cases in many other contexts which automated systems may use to assess the type of cases used. Academic journals less frequently discuss the type of rule chosen in other contexts, providing less readily available annotation. Nevertheless, the very high degrees of accuracy obtained in the Fourth Amendment context suggests that transformer models are capable of differentiating the type of legal test used in a legal opinion, an effective first step.

4.1. Key Accomplishments

Correct classification of bright-line and totality-of-the-circumstances cases is achievable with current transformer-based NLP methods. Fine-tuned BERT, RoBERTa, and ALBERT models were successfully employed for binary classification of full texts in this study. Deep-learning transformers attained accuracies of up to 90.26% on positive language and 92.31% on full texts. This research demonstrates the scalability of transformers to longer lengths of text via a sliding-window approach. The results show that, while positive language is sufficient, transformer models are now capable of extracting their own effective feature representations from supplied text to perform at even higher accuracy. The process of fine-tuning a pre-trained transformer for text classification is shown as one attainable and accessible method for assistive AI in a variety of legal domains.

4.2. Future Work

Similar methods may be applied to a larger dataset. Continuing to grow the corpus should increase model accuracy. Further exploration to compare and contrast effectiveness of different transformer models and variants is merited. A study of the effect of cased and uncased language could also provide insight. A more thorough assessment of different learning rates, batch sizes, and other parameters could be effective, but was not within the scope of this study.

Acknowledgements

The authors would like to acknowledge the contribution of Rui Zheng to case aggregation and labelling tasks. This research began in satisfaction of a course project for the University of Pittsburgh School of Law's LAW 5719 Applied Legal Data Analytics and AI course taught by Dr. Kevin Ashley and Dr. Jaromir Savelka. The authors are very thankful for their insight, proofing, and expertise in the completion of this paper.

References

[1] Smith M. Advanced legal writing : theories and strategies in persuasive writing. New York: Aspen Law & Business; 2002.

[2] Ashley KD. Artificial intelligence and legal analytics: New tools for law practice in the digital age. Cambridge University Press; 2017.

[3] Gardner AvdL. An Artificial Intelligence Approach to Legal Reasoning. Cambridge, MA, USA: MIT Press; 1987.

[4] Alschuler A. Bright Line Fever and the Fourth Amendment. University of Pittsburgh Law Review. 1984 1;45.

[5] New York v. Belton, 453 US 454 - Supreme Court 1981;.

[6] Dery III GM, Dery GM. Issue 3 2000 III, Improbable Cause: The Court's Purposeful Evasion of a Traditional Fourth Amendment Protection in Wyoming v. Houghton, 50 Case W. Res; 2000.

[7] Navarette v. California, 572 U.S. 393, 2014.;.

[8] Florida v. JL, 529 US 266 - Supreme Court 2000;.

[9] Nelson DM. Illinois v. Wardlow: A Single Factor Totality. Utah Law Review. 2001;2001.

[10] Cardellino C, Alemany LA, Teruel M, Villata S. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In: Proceedings of the International Conference on Artificial Intelligence and Law. New York, New York, USA: Association for Computing Machinery; 2017. p. 9–18.

[11] Šsavelka J, Ashley KD. Segmenting U.S. court decisions into functional and issue specific parts. In: Frontiers in Artificial Intelligence and Applications. vol. 313. IOS Press; 2018. p. 111–120.

[12] Koniaris M, Anagnostopoulos I, Vassiliou Y. Network analysis in the legal domain: A complex model for European Union legal sources. Journal of Complex Networks. 2018 4;6(2):243–268.

[13] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European court of human rights: A natural language processing perspective. PeerJ Computer Science. 2016 10;2016(10):e93.

[14] Chalkidis I, Androutsopoulos I, Aletras N. Neural Legal Judgment Prediction in English. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. 2019 6:4317–4323.

[15] Angwin J, Larson J, Mattu S, Kirchner L. Machine Bias; 2016.

[16] Ashley KD. Case-Based Reasoning and its Implications for Legal Expert Systems; 1992.

[17] Devlin J, Chang MW, Lee K, Google KT, Language AI. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding;.

[18] Vaswani A, Brain G, Shazeer N, Parmar N, Uszkoreit J, Jones L, et al. Attention Is All You Need;.

[19] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach; 2019.

[20] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In: Proceedings of the International Conference on Learning Representations; 2020. .

[21] ThilinaRajapakse/simpletransformers: Transformers for Classification, NER, QA, Language Modelling, Language Generation, T5, Multi-Modal, and Conversational AI;.

[22] Rajpurkar P, Jia R, Liang P. Know what you don't know: Unanswerable questions for SQuAD. In: ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). vol. 2. Association for Computational Linguistics (ACL); 2018. p. 784–789.

[23] Lai G, Xie Q, Liu H, Yang Y, Hovy E. RACE: Large-scale ReAding comprehension dataset from examinations. In: EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings. Association for Computational Linguistics (ACL); 2017. p. 785–794.

[24] Rabelo J, Kim MY, Goebel R. Combining similarity and transformer methods for case law entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 290–296.

[25] Sanchez L, He J, Manotumruksa J, Albakour D, Martinez M, Lipani A. Easing Legal News Monitoring with Learning to Rank and BERT. In: Proceedings of the European Conference on Information Retrieval; 2020. p. 336–343.

[26] The Supreme Court Database;.

[27] Sommers CD. Presumed Drunk until Proven Sober: The Dangers and Implications of Anonymous Tips Following Navarette V. California. South Dakota Law Review. 2015 6;60(2):327.

[28] Katz LR. Baldwin's Ohio Handbook Series: Ohio Arrest, Search and Seizure. 2020;20(2).

[29] Mendelson A. The Fourth Amendment and Traffic Stops: Bright Line Rules in Conjunction with the Totality of the Circumstances Test. J Crim L and Criminology. 1988;88:930.

[30] Glandon KR. Bright Lines on the Road: The Fourth Amendment, The Automatic Companion Rule, the "Automatic Container" Rule, and a New Rule for Drug- or Firearm-Related Traffic Companion Searches Incident to Lawful Arrest. Am Crim L Rev. 2009;46:1267.

[31] Butterfield EJ. Bright Line Breaking Point: Embracing Justice Scalia's Call for the Supreme Court to Abandon an Unreasonable Approach to Fourth Amendment Search and Seizure Law. Tul L Rev. 2007;82:77.

[32] Saleem O. The Age of Unreason: The Impact of Reasonableness, Increased Police Force, and Colorblindness on Terry "Stop and Frisk". Okla L Rev. 1997;50:451.

[33] Urbonya KR. Rhetorically Reasonable Police Practices: Viewing the Supreme Court's Multiple Discourse Paths. Am Crim L Rev. 2003;40:1387.

[34] Bryk JK. Anonymous Tips to Law Enforcement and the Fourth Amendment: Arguments for Adopting an Imminent Danger Exception and Retaining the Totality of the Circumstances Test. Geo Mason U Civ Rts L J. 2003;13:277.

[35] Krippandorf E. Florida v. J.L.: To Frisk or Not to Frisk: The Supreme Court Sheds Light on a Use of Anonymous Tipsters as a Predicate for Reasonable Suspicion. New Eng J on Crim and Civ Confinement. 2002;28:161.

[36] Dripps DA. Responding to the Challenges of Contextual Change and Legal Dynamism in Interpreting the Fourth Amendment. Miss L J. 2012;81:1085.

[37] Cohen J. A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement. 1960 4;20(1):37–46.

[38] huggingface. Pretrained Models. 2020.

[39] Klimt B, Yang Y. The enron corpus: A new dataset for email classification research. In: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). vol. 3201. Springer Verlag; 2004. p. 217–226.

[40] Google Books Ngrams - AWS Public Data Set;.

[41] Nakkiran P, Kaplun G, Bansal Y, Yang T, Barak B, Openai IS. Deep Double Descent: Where Bigger Models and More Data Hurt. In: ICLR 2020; 2020. .

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200851

The Role of Vocabulary Mediation to Discover and Represent Relevant Information in Privacy Policies

Valentina LEONE [a,1], Luigi DI CARO [a]
[a] *Computer Science Department, University of Turin, Italy*

Abstract. To date, the effort made by existing vocabularies to provide a shared representation of the data protection domain is not fully exploited. Different natural language processing (NLP) techniques have been applied to the text of privacy policies without, however, taking advantage of existing vocabularies to provide those documents with a shared semantic superstructure. In this paper we show how a recently released domain-specific vocabulary, i.e. the Data Privacy Vocabulary (DPV), can be used to discover, in privacy policies, the information that is relevant with respect to the concepts modelled in the vocabulary itself. We also provide a machine-readable representation of this information to bridge the unstructured textual information to the formal taxonomy modelled in it. This is the first approach to the automatic processing of privacy policies that relies on the DPV, fuelling further investigation on the applicability of existing semantic resources to promote the reuse of information and the interoperability between systems in the data protection domain.

Keywords. legal vocabularies, ontology population, text similarity, data protection

1. Introduction

In the European Union (EU), the entry into force of the General Data Protection Regulation (GDPR) [1] has brought the domain of data protection to the forefront, encouraging the research in knowledge representation and natural language processing (NLP), among the other fields. On the one hand, several ontologies and vocabularies adopted Semantic Web standards to provide a formal representation of the data protection framework set by the Regulation. On the other hand, different NLP approaches have been applied to the text of privacy policies to address classification tasks that assign one or more labels to the paragraphs of a privacy policy, according to its content.

These two lines of research do not seem to pursue a common goal. The labels used in the classification tasks are not organised in a semantic structure and the outcomes of these tasks are hardly applicable outside the context of the project for which they were implemented. Consequently, the full potential of Semantic Web oriented vocabularies is not exploited to provide the text of privacy policies with a shared semantic superstructure and their effort to promote interoperability between systems on the Web is lost.

[1] Corresponding Author. E-mail: valentina.leone@unito.it

Among the most recent semantic resources that were proposed to model the data protection domain, the Data Privacy Vocabulary (DPV) [2] organises its concepts in a lightweight taxonomic structure. This vocabulary has drawn the attention of many projects that declared their interest in its adoption [3,4,5]. However, to the best of our knowledge, no effort has been yet made to automatically extract, from privacy policies, the information that is relevant with respect to the concepts modelled by this vocabulary.

In this paper we present a first approach that is driven by the concepts modelled in the DPV to automatically discover the relevant information in privacy policies. The proposed method integrates the knowledge represented in the DPV with the information modelled in BabelNet[2] [6], i.e. a general-purpose vocabulary that provides a semantic network of concepts linked through lexical and semantic relationships. The outcome of the method is provided in a machine-readable format that bridges the gap between the unstructured text of a privacy policy and the formal taxonomy of concepts provided by the DPV. The paper in structured as follows: Section 2 presents some related work and Section 3 describes the resources that we adopted in our experiments. Section 4 explains the steps that were implemented to map the information in the privacy policies on the concepts of the DPV, while Section 5 explains how the proposed methodology was evaluated. Section 6 describes the machine-readable representation that has been provided for the results and Section 7 ends the paper with some final remarks.

2. Related Work

Many works in the data protection field applied NLP techniques to the text of privacy policies for labelling their paragraphs according to the information they express. Polisis [7] relies on domain-specific word embeddings and a hierarchy of neural networks to classify the paragraphs of the policies in the OPP-115 corpus [8]. The approach described in Polisis is then refined and improved in [9]. Supervised machine learning models are also used in PrivacyGuide [10] to highlight the risk level associated to some privacy aspects described in privacy policies. An unsupervised learning technique is adopted in [11] to extract the topics emerging from a corpus of more that 4K privacy policies, then comparing those topics with the information represented by the labels provided in the OPP-115 corpus. KniGHT [12] exploits techniques of semantic text matching for mapping the sentences of a privacy policy on the most related articles of the GDPR.

Other projects focused on representing the information originating from different textual sources in the data protection field into structured machine-readable representations. The Lynx[3] project aims, in one of its use cases, to create a knowledge graph for the data protection field interlinking domain-related legal texts and providing algorithms able to automatically enlarge the knowledge base when new relevant documents are issued [13]. The SPECIAL[4] project [3] focused on the development of machine-readable policy languages and the DPV was proposed in the context of this project. The MIREL[5] project included the PrOnto ontology among its outcomes, proposing a technique based on Open Information Extraction to map the information extracted from privacy policies on the classes modelled by the ontology [14].

[2] https://babelnet.org/

[3] http://www.lynx-project.eu/

[4] https://www.specialprivacy.eu/

[5] https://mirelproject.eu/index.html

Table 1. The modules in the DPV and the labels in the OPP-115 corpus for paragraph-level annotations.

DPV Modules	Personal Data Category; Processing; Purpose; Legal Basis; Data Controller; Recipient; Data Subject; Technical Organisational Measures
OPP-115 Labels	First Party Collection/Use; Third Party Sharing/Collection; User Choice/Control; User Access, Edit & Deletion; Data Retention; Data Security; Policy Change; Do Not Track; International & Specific Audiences; Other

3. Scope and Limitations of the Adopted Resources

In our experiments, the DPV was jointly used with a corpus of privacy policies, named OPP-115 corpus[6] [8]. The different nature and scope of these resources required us to put some constraints on their use.

The Data Privacy Vocabulary[7] [2] was first released in July 2019. Through a formal representation that relies on RDF and OWL, it aims to provide a basic vocabulary of terms related to the data protection domain framed by the GDPR. The DPV is made of several modules, that provide a taxonomy of terms related to different aspects involved in the personal data handling. Those modules are listed in the first row of Table 1.

The OPP-115 corpus includes 115 privacy policies that were manually labelled with a two layered annotation made at paragraphs and text spans level. The paragraphs are associated with labels representing ten different data practices, listed in the second row of Table 1. Within an annotate paragraph, text spans are labelled with attribute-value pairs that are specific for a given data practice and that can assume a limited set of values. The OPP-115 corpus collects privacy policies that were issued by US-companies some years before the entry into force of the GDPR. Therefore, some concepts modelled in the DPV can not be expected to be mentioned in the corpus.

To take into account of the different scope of the resources, the extraction of relevant information from the text of the privacy policies is limited to the concepts in the *Personal Data Category* and *Purpose* modules in the DPV. Similarly, we only considered the paragraphs of the privacy policies that were assigned to the *First Party Collection/Use* label in the corpus, as we expect this information to be more likely to be found within them. From here on, even if no further specified, we assume that the implemented method and its evaluation were applied taking into account the aforementioned constraints in the joint adoption of the two resources.

4. Method

The method described in this Section is composed of three sequential steps, where the output of one step becomes the input of the following one. The first step creates broad mappings between some parts of the text in the privacy policies and the modules of the DPV. The second step tries to refine these mappings selecting, from the modules in the DPV, some classes that could be suitable for the refinement. The last step chooses, from the set of suitable classes, the one that will yield the needed refinement.

[6] https://www.usableprivacy.org/data

[7] https://dpvcg.github.io/dpv/

4.1. Broad Mappings of Text Chunks on the DPV Modules

To discover the parts of the text in privacy policies that are relevant with respect to the DPV, the first step of the method leverages the distinctiveness of the terminology that characterises each module of the vocabulary. This evidence was found collecting and ordering, by decreasing frequency, the terms used to name the classes in each module and to provide the description of their meaning in natural language (through the RDF property `dct:description`). As Table 2 shows, the collected terms are in most of the cases exclusive for each module and only few words overlap. Thus, the nouns in each list can be considered as *descriptors* for the type of information that each module of the DPV represents. Moreover, for each descriptor, we also considered its synonyms, that were automatically retrieved from BabelNet.

The discovery of relevant parts of the text in the privacy policies relied on these descriptors. For each sentence, the noun chunks (i.e. the nominal phrases) were extracted using the available libraries of the SpaCy dependency parser[8] and the chunks roots (i.e. the words connecting the noun chunks to the rest of the parsed sentence) were used to perform the mappings. When the root of a chunk matched a descriptor, the chunk was mapped on the corresponding module. In case of a match with a descriptor that appears in both the modules, the chunk was assigned to the module where the descriptor has the highest frequency. In case of a tie, the chunk was preliminarily assigned to both modules. The chunks whose roots did not find a match with a descriptor were considered not relevant in establishing a match with the DPV. Two examples of the mappings performed in this step are shown below. The module assigned to each chunk is indicated in a square box and the roots of the chunks, used to determine the mappings, are underlined.

$\boxed{\text{Purpose}}$ $\quad$ *customer service* $\underset{\text{root}}{\underline{\textit{purpose}}}$

$\boxed{\text{Personal Data Category}}$ $\quad$ *mobile device unique id* $\underset{\text{root}}{\underline{\textit{number}}}$

4.2. Detection of Candidate Classes for the Refinement of the Broad Mappings

Given the coarse assignments of noun chunks to one or two modules in the DPV, the second step focused on the refinement of these assignments identifying a set of more specific candidate classes in the taxonomies of the modules. Given a text chunk, a first control checks if the name of a class in the DPV, or one of its synonyms retrieved with BabelNet, matches the chunk or appears as a sub-string in it. If this is the case, the set of

[8] https://spacy.io/

Table 2. The six most frequent words used to name and describe the classes in the *Purpose* and *Personal Data Category* modules of the DPV. The number next to each noun represents the frequency of the noun in the module. More than six terms are present in both lists due to the tie in the frequencies.

DPV Module	**Top-6 of the frequent words**
Purpose	(service, 17), (user, 9), (product, 8), (research, 8), (optimisation, 7), (datum, 6), (activity, 6), (commercial, 6), (recommendation, 6), (interface, 4), (individual, 4), (purpose, 4)
Personal Data Category	(individual, 148), (information, 141), (history, 18), (health, 17), (personal, 17), (social, 13), (credit, 13), (datum, 13), (professional, 11)

candidate classes is made of a single element, i.e. the matching class. For instance, the fragment considered in the previous Section:

$$\boxed{\text{Purpose}} \quad \underset{\text{dpv:CustomerCare}}{\underline{\textit{customer service}}} \quad \underset{\text{root}}{\underline{\textit{purpose}}}$$

contains the sub-string *customer service* that is a synonym of the string *customer care*. In turn, the latter matches the homonym DPV class, that is considered as a candidate class to perform the refinement of the mapping with the *Purpose* module.

If no class is detected with this first check, then the lists of modules descriptors (see Section 4.1) are used to populate the list of candidate classes. Specifically, for each descriptor that matches a word in the text chunk, the class from which the descriptor was extracted is added to the list of candidate classes. If a candidate class is a leaf in the taxonomy of a module, then it is substituted by its direct superclass in order to avoid matches with too specific classes. The root of the text chunk is excluded from the search of the candidate classes, because it already contributed to the broad mappings with the DPV modules. For instance, in the following fragment:

$$\boxed{\text{Personal Data Category}} \quad \textit{mobile} \quad \underset{\text{dpv:DeviceBased}}{\underline{\textit{device}}} \quad \underset{\text{dpv:Identifying}}{\underline{\textit{unique}}} \quad \textit{id}\ \underset{\text{root}}{\underline{\textit{number}}}$$

the word *device* matches a descriptor that corresponds to the *DeviceBased* class, while the word *unique* matches a descriptor corresponding to the *UID* (i.e. user identifier) class. However, as the class is a leaf in the taxonomy of the *Personal Data Category* module, its direct superclass, i.e. *Identifying*, is added to the set of candidate classes of the chunk.

4.3. Selection of the Class for Refining the Broad Mappings

The third and last step of the method selects, among the candidate classes, the most suitable for refining the broad mapping between a text chunk and a module. Following many simple but consolidated state of the art approaches [15,16], the class is selected by computing the cosine similarity between the text chunk and its candidate classes. The vector representation of both the text chunk and its candidate classes was obtained from the pre-trained GloVe word embeddings[9] [17] that were combined according to some weights for representing the different contributions given by each word in the overall vector representation.

The vector representation for a text chunk is obtained collecting the set W_F of the embeddings for the content words in the chunk and the set W_S of the embeddings for the content words that occur in the same sentence of the text chunk. Assuming that all the words in the chunk contribute equally to its vector representation, a weight equal to 1 is assigned to each word embedding in W_F. By contrast, the weights associated to the word embeddings in W_S assume that the contribution of a word occurring in the same sentence of the chunk is equal to the frequency of that word in the sentence divided by the total number of distinct words in the sentence. The vector representation of the text chunk is computed, then, by multiplying each embedding in the set W_F and W_S by the corresponding weight and computing the mean vector resulting from the two sets.

By contrast, the vector representation for a candidate class is conceptually based on the computation of some Term Frequency-Inverse Document Frequency (TF-IDF) scores

[9] `https://nlp.stanford.edu/projects/glove/`. We use the 300-dimensional vectors.

Table 3. Statistics about the number of text chunks that were retrieved in the privacy policies and the number of classes of the DPV that were associated with at least a text chunk.

	Purpose	**Personal Data Category**	**Total**
Chunks (with repetitions)	852	4025	4877
Chunks (no repetitions)	224	747	971
Retrieved classes	17	85	102

associated to the words used in its description. Following the assumption that underpins the TF-IDF measure, terms that are used in one or few class descriptions should be emphasised, because they likely are more representative of a specific DPV class, while terms that are used frequently in the definitions of the classes should have less relevance. Therefore, being C the set of candidate classes for a text chunk, the TF-IDF scores for the content words used in the description of a class c in C were computed considering the frequency of these words in the description of c and the inverse document frequency of these words with respect to the definition of the other classes in C. The embeddings for the content words in the description of c were then multiplied by the corresponding TF-IDF scores and the average vector of the embeddings was computed to obtain the vector representation of c.

Finally, the cosine similarity is computed between the vectorial representations of the chunk and of each candidate class. The class that results in the highest cosine similarity value is considered as the best candidate for the refinement. The example below shows the similarity values computed for the text chunks discussed in the previous sections (the class that determined the final mapping is highlighted in bold).

Purpose	*customer service*	*purpose*
	dpv:CustomerCare 0.77	root

Personal Data Category	*mobile*	*device*	*unique*	*id*	*number*
		dpv:DeviceBased 0.76	dpv:Identifying 0.60		root

5. Evaluation

5.1. Statistics about the Performed Mappings

Table 3 shows a summary of the number of text chunks that were extracted with the methodology described in Section 4. Overall, we extracted 4877 chunks that were associated to 102 classes of the DPV (out of a total of 192 classes in the two modules of interest). Each chunk occurs one or more times in the corpus of privacy policies. Omitting the repetitions, the number of unique text chunks that were retrieved is equal to 971. Among them, 128 chunks were detected because the name of a class (or one of its synonyms) matched the chunk or appeared as a sub-string in it. The remaining 843 chunks were retrieved populating the lists of candidate classes, relying on the descriptors extracted for each module (see Section 4.2).

5.2. Precision Assessment of the Performed Mappings

The evaluation of the results relied on the annotations of the privacy polices provided by the OPP-115 corpus (see Section 3 for further details).

Table 4. DPV classes with the highest and lowest number of text chunks mapped on them.

	Personal Data Category	**Purpose**
Most Frequent	(Device Based, 758), (Email Address, 282), (Contact, 183)	(Commercial Interest, 337), (Purpose, 266), (Security, 49)
Less Frequent	(Philosophical Belief, 1), (Disciplinary Action, 1), (Thought, 1)	(Access Control, 1), (Service Optimization, 1), (Optimisation For Consumer, 7)

To estimate the precision of the mappings extracted by our method, we created a correspondence between the values of the *Personal Information Type* attribute of the OPP-115 corpus and some of the DPV classes in the *Personal Data Category* module. Those correspondences were manually identified analysing the descriptions provided both for the attribute values in the corpus and the classes in the DPV, unravelling similarities in the type of information that they represent. Table 5 shows the mappings that we considered. In this table, the numbers between squared brackets represent the level of a class in the taxonomy of the module (we say that the most general class in the taxonomy lies at level 0, all its direct subclasses lie at level 1, and so on). Most of the correspondences were made between an attribute value and a class at the second level in the module. We found that some attribute values are very general and no meaningful correspondences were found. A similar analysis was also performed on the values of the *Purpose* attribute in the OPP-115 corpus and the classes of the homonym module in the DPV. Table 6 shows the mappings that we considered. In this case, most of the attribute values were associated with classes at level 1 in the *Purpose* module.

Based on the correspondences that we drawn, we identified three different scenarios for the evaluation. Given a text chunk f that is extracted from a sentence s in a privacy policy: *(i)* f is part of a text span in s and the attribute-value pair associated to the span matches the class of f, following the correspondences that were identified for the evaluation; *(ii)* f is part of a text span that is labelled in s, but the attribute-value pair associated to the span do not match the class associated with f; *(iii)* f does not correspond to any of the text spans that were annotated in s. We computed the number of text chunks that

Table 5. Correspondences between the values of the *Personal Information Type* attribute in the OPP-115 corpus and the classes in the *Personal Data Category* DPV module. The last row lists the attribute values that did not find a match in the module.

Attribute Values	**Classes in the *Personal Data Category* Module**
Financial	Financial [1]
Health	Medical Health [2]
Contact	Contact [2], Name[3]
Location	Location [2]
Demographic	Demographic [2], Physical Characteristic [2], Professional [2], Family [2]
Personal identifier	Identifying [2], Financial Account[2]
User online activities	Behavioral [2], Social Media Communication [3]
User profile	Identifying[2], Preference[2]
Social media data	Social Network [2]
IP address & device ids	Device Based [2]
Computer information	Device Based [2]
Cookies & traking elements, Survey data, Generic personal information, Other, Unspecified	

Table 6. Correspondences between the values of the *Purpose* attribute in the OPP-115 corpus and the classes in the *Purpose* DPV module. The last row lists the attribute values that did not find a match in the module.

Attribute Values	**Classes in the *Purpose* Module**
Basic service/feature	Service Provision [1]
Additional service/feature	Service Provision [1], Service Personalization [1]
Advertising	Service Personalization [1]
Marketing	Commercial Interest [1], Service Personalization [1]
Analytics/research	Research And Development [1], Service Optimization [1]
Personalisation/Customisation	Service Personalization [1]
Service Operation & Security	Security [1]
Legal Requirement, Merger/Acquisition, Other, Unspecified	

fit each of the three scenarios and we collected the results in Table 7. Some insights from the evaluation are presented in the next Section.

5.3. Insights from the Results of the Evaluation

The first insight that comes from the retrieved mappings concerns the coverage of the two modules of interest in the DPV with respect to the classes that were associated to some text chunks in the privacy policies (see the last row of Table 3). The number of classes that were automatically mapped on the text chunks slightly exceeds (53.1%) half of the concepts represented in the DPV modules of interest. However, it should be noticed that many concepts in the DPV are very specific and likely difficult to find in the privacy policies text. Classes like *Music* or *Accent* in the *Personal Data Category* module were not mapped on any text chunk. By contrast, chunks related to the *IP Address, Location* and *Contact* classes were frequently extracted. This intuition is reinforced by looking at Table 4, that provides an excerpt of the classes for which the highest and lowest number of text chunks (considering repetitions) was found.

Concerning the evaluation technique explained in Section 5.2, we noticed that most of the labels mismatches occurred because the text spans in the corpus were associated to general labels (like *Other*). We therefore believe that, in this case, our vocabulary-driven approach could provide an advantage over the manual annotation proposed in the corpus, suggesting more precise labels for the text spans. By contrast, the scenario in which a text chunk, that was automatically extracted by our method, was not annotated in the corpus needs further investigations for evaluating to what extent the lack of an annotation indicates an incorrect automatic mapping or is rather a corpus fault. However, we believe that the results obtained with this first evaluation approach, based solely on the labels provided by the corpus, have provided promising insights that encourage the refinement of the evaluation approach, that could involve manual expert evaluation.

Table 7. Results of the evaluation that is based on the manual drawing of the correspondences between attribute values in the OPP-115 corpus and the classes in the DPV according to the three different scenarios discussed in Section 5.2. Percentages are computed with respect to the total number of noun chunks extracted for the corresponding module.

	Purpose	**Personal Data Category**	**Total**
Match	114 (13.4%)	1351 (33.6%)	1465 (30.0%)
Mismatch	296 (34.7%)	858 (21.3%)	1154 (23.7%)
No annotation	442 (51.9%)	1816 (45,1%)	2258 (46.3%)

6. Semantic Web Oriented Representation of the Results

We propose a machine-readable representation of the mappings that were automatically extracted by our method. The understanding that we would like to provide about the proposed mappings is that of *semantic domains* that are identified by the concepts of the DPV, and *domain elements* that correspond to the text chunks and that are related to the semantic domains. A standardised modelling solution to this intuition is provided by the *Collection* Ontology Design Pattern (ODP)[10], that can be used to represent the membership to a domain, not to be intended in the sharp sense defined in the set theory (as specified by the documentation provided for the ODP).

We used the RDF syntax to formalise the mappings extracted from the privacy policies by using the representational model provided by this ODP. For each DPV class that was associated to a text chunk in a privacy policy, a new class representing a related semantic domain was introduced. The text chunks were then associated to their semantic domains with the property `isMemberOf`, introduced by the ODP. The properties `skos:label` and `skos:example` were used to associate to the chunks their natural language strings and the sentences of the privacy policy from which they were extracted, as shown in the example below.

```
:DemographicDomain rdf:type dpv:Demographic, owl:Thing.

:DemographicAnalysisConcept rdf:type skos:Concept, owl:Thing;
  odp:isMemberOf :DemographicDomain;
  rdfs:label "demographic analysis"@en;
  skos:example "Perform statistical, demographic, and marketing
  analyses of users of the Sites and their purchasing patterns"@en.
```

This example shows the advantage of the proposed representation: an unstructured delivery of the results could erroneously suggest that, intuitively, if the concept *Demographic* contributed to the identification of the text chunk *demographic analysis*, then there is a close match between their meanings. By contrast, the representation of a semantic domain related to the *Demographic* concept and the association of the text chunk to this domain provides a new perspective on the proposed mapping. Indeed, *demographic analysis* and *demographic personal data* are different in their meaning, but it is likely that a demographic analysis will involve the processing of demographic personal data, thus legitimating a mapping of the text chunk with the corresponding domain.

7. Conclusion and Future Work

In this paper we presented the first approach that exploits a recently-released vocabulary for the data protection domain to discover the relevant information in the text of privacy policies. Moreover, we presented a machine-readable representation of the results, based on RDF and a standardised ontological solution. The obtained results show that NLP approaches in the data protection domain can benefit from existing semantic resources, to share information and promote interoperability between systems. We plan to continue the work on the refinement of the proposed approach applying it to a corpus of GDPR-compliant privacy policies. This would make it possible to overcome some of the aforementioned limitations of the OPP-115 corpus and to extend the applicability of the DPV to other modules.

[10] `http://ontologydesignpatterns.org/wiki/Submissions:Collection`

References

[1] 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). Official Journal of the European Union L119. p. 1-88 (May 2016).

[2] Harshvardhan JP, Polleres A, Bos B et al. Creating a vocabulary for data privacy: the first-year report of data privacy vocabularies and controls community group (DPVCG). 2019. In: Panetto H, Debruyne C, Hepp M, et al., editors. On the Move to Meaningful Internet Systems: OTM 2019 Conferences; Confederated International Conferences; 2019 Oct; Rhodes, Greece. Springer, Cham. pp 714-30.

[3] Bonatti PA, Kirrane S, Petrova, IM et al. Machine Understandable Policies and GDPR Compliance Checking. Künstl Intell. 2020 Jul;34:303-15.

[4] Ryan P, Crane M, Brennan R. Design Challenges for GDPR RegTech. In: Filipe J, Smialek M, Brodsky A, Hammoudi S, editors. Proceedings of the 22nd International Conference on Enterprise Information Systems. ICEIS 2020; 2020 May; Prague, Czech Republic. SCITEPRESS 2020; 2020. p. 787-95.

[5] Debruyne C, Pandit HJ, Lewis D. et al. "Just-in-time" generation of datasets by considering structured representations of given consent for GDPR compliance. Knowledge and Information Systems. 2020 Apr;62:3615–40.

[6] Navigli, R, Ponzetto, SP. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence. 2012 Dec;193:217-50.

[7] Harkous H, Fawaz K, Lebret R, et al. Polisis:Automated analysis and presentation of privacy policies using deep learning. In: 27th USENIX Security Symposium. USENIX Security '18; 2018 Aug; Baltimore, USA. USENIX Association; 2018. p. 531–48.

[8] Wilson S, Schaub F, Dara AA, et al. The creation and analysis of a website privacy policy corpus. In: Erk K, Smith NA, editors. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2016 Aug; Berlin, Germany. Association for Computational Linguistics; 2016. p. 1330-40.

[9] Nejad NM, Jabat P, Nedelchev R et al. Establishing a strong baseline for privacy policy classification. In: Hölbl M, Rannenberg K, Welzer T, editors. ICT Systems Security and Privacy Protection. SEC 2020; 2020 Sep; Maribor, Slovenia. Springer, Cham. p. 370-83.

[10] Tesfay WB, Hofmann P, Nakamura T et al. PrivacyGuide:Towards an implementation of the EU GDPR on internet privacy policy evaluation. In: Verma RM, Kantarcioglu M, editors. Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics. Eighth ACM Conference on Data and Application Security and Privacy; 2018 Mar; Tempe, USA. Association for Computing Machinery; 2018. p. 15–21

[11] Sarne D, Schler J, Singer A et al. Unsupervised topic extraction from privacy policies. In: Liu L, White R, editors. WWW '19: Companion Proceedings of The 2019 World Wide Web Conference. The Web Conference; 2019 May; San Francisco, USA. Association for Computing Machinery; 2019. p. 563–68.

[12] Nejad NM, Scerri S, Lehmann J. Knight: Mapping privacy policies to GDPR. In: Faron Zucker C, Ghidini C, Napoli A, et al. editors. Knowledge Engineering and Knowledge Management. EKAW; 2018 Nov; Nancy, France. Springer, Cham; 2018. p. 258–72.

[13] Montiel-Ponsoda E, Gracia J, Rodriguez-Doncel V. Building the legal knowledge graph for smart compliance services in multilingual Europe. In: Rodríguez-Doncel V, Casanovas P, González-Conejero J, editors. Proceedings of the 1st Workshop on Technologies for Regulatory Compliance; 30th International Conference on Legal Knowledge and Information Systems; 2017 Dec; Luxembourg. CEUR Wordshop Proceedings; 2017. p. 15–17.

[14] Palmirani M, Bincoletto G, Leone V, et al. Hybrid Refining Approach of PrOnto Ontology. In: Kő A, Francesconi E, Kotsis G, et al., editors. Electronic Government and the Information Systems Perspective. EGOVIS; 2020 Sep; Bratislava, Slovakia. Springer, Cham; 2020. p. 3-17.

[15] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information processing & management. 1988 Aug;24(5):513-23.

[16] Kenter T, de Rijke M. Short Text Similarity with Word Embeddings. In: CIKM '15: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. CIKM; 2015 Oct; Melbourne, Australia. Association for Computing Machinery; 2015. p. 1411–20.

[17] Pennington J, Richard S, Manning C. Glove: Global vectors for word representation. In: Moschitti A, Pang B, editors. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP; 2014 Oct; Doha, Qatar. Association for Computational Linguistics; 2014. p. 1532-43.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200852

A General Theory of Contract Conflicts with Environmental Constraints

Gordon J. PACE [a,1]
[a] *University of Malta, Msida, Malta*

Abstract. One advantage of using formal deontic logic to represent and reason about normative texts is that one can analyse such texts in a precise and incontrovertible manner. Conflict analysis is one such analysis technique — assessing whether a number of contracts, or more generally normative texts, are internally consistent, in that they may not lead to a situation in which active norms conflict or even contradict each other. In this paper we extend existing techniques from the literature to address conflicts in the context of environmental constraints on actions regulated by the contract, and which the parties involved can carry out. The approach is logic-agnostic and we show how it can be applied to a service provision contract written in $\mathcal{CL}$.

Keywords. deontic logic, contract conflicts

1. Introduction

Lessig's *code is law* dictum is based on the notion that there are different means of restricting or regulating behaviour. Lessig identifies different forms of such means, including legal, and architectural (or by physical design). For instance, by the very nature of the real world, physical laws cannot be violated, and thus restrict our behaviour no matter the legal constraints one may have. To use an example used by Lessig, until the invention of human means of flight, a law giving a person rights to his or her land and everything above it (and below — the principle of *'cuius est solum eius est usque ad coelum et ad infernos'*, or *'the owner of the land all the way up to heaven, and the way down to hell'*) was no different to a law which gives rights only over the first 100 metres above the owned land. Conversely, obliging a person to be in two geographically distant points $O(inRome) \wedge O(inGlasgow)$ is unsatisfiable due to the physical nature of the two actions or states of affairs. Such constraints implicitly limiting one's behaviour have (arguably) become more common with the rise of computer code — hence Lessig's *'code is law'*. Computer code which prohibits downloading a file unless one is logged in, and logging in requires to have registered before, which in turn requires to have provided personal information, means that an agreement which prohibits you from ever giving personal details is implicitly incompatible with one which obliges you to download a particular file. In a free-for-all world, the two agreements are compatible, but not in the world where computer code is regulating our interaction with the server.

The notion of conflicts in deontic logics [8] has been investigated before in order to identify situations which may arise from a formula in which at least one of the parties

[1] Corresponding author; E-mail: gordon.pace@um.edu.mt

has conflicting norms to satisfy e.g. being both obliged and prohibited from performing a particular action. Much of the work in the literature assume that there are no restrictions on the actions or states being regulated, although some approaches allow for the consideration of mutually exclusive actions. Such a consideration allows us to identify the conflict in $O(inRome) \wedge O(inGlasgow)$ if we know that the two actions are mutually exclusive. However, limiting oneself to mutually exclusive actions is not strong enough to deal with more complex environmental constraints or Lessig's code. For instance, in the file downloading example, the constraint is more sophisticated than a mere mutual exclusion of actions — with a four-fold temporal dependancy: *download* cannot happen before *login*, which in turn cannot happen before *registration* which cannot happen before *sendPersonalDetails*.

In this paper we investigate the interaction between environmental (or factual) constraints and contract conflicts. We define a general framework for temporal deontic logics and general action constraints, in which we characterise the notion of a conflict in a formula in the deontic logic under particular constraints. In order to show the use of this general framework, we show how it can be applied to the contract language $\mathcal{CL}$ [1] and use it to analyse an internet service provider contract for conflicts under temporal constraints arising from the underlying computer system.

2. Related Work

Detection of conflicts between normative documents, as a first step towards eliminating them if possible (e.g. when a contract is still being drafted) or to resolve or reconcile them when not (e.g. when such a conflict arises from existing contracts during a business acquisition), has long been considered an important challenge [18]. With normative documents expressed in the form of a deontic logic one has the opportunity of formally characterising the class of such conflicts and derive algorithms to extract them automatically, and one finds various such work for particular contract languages.

One of the first formal and computational characterisations of deontic conflicts and algorithms to detect them was for the contract language $\mathcal{CL}$ [1]. Fenech et al. [9] characterised the notion of conflicts in $\mathcal{CL}$ formulae through the use of a trace semantics of the deontic logic, with the analysis integrated in the conflict detection CLAN tool [10]. The conflict analysis was used for other deontic logics such as C-O diagrams [15]. Other approaches to conflict discovery taking a definitional approach to characterising conflicts include [13] using defeasible logic, [20] which takes a unification-based approach, [19] which addresses conflicts in dynamic settings,[22] which addresses conflict by taking by devising preferential rules, and [12] which defines a notion of *coherence*.

An alternative axiomatic approach to conflict discovery was used to identify conflicts in contract automata [6]. Unlike the former work, the notion of conflicts is captured in a number of axioms which characterise how conflicts arise and under which conditions they are maintained, but similar to the work in $\mathcal{CL}$, the approach also took an operational view of the deontic formulae inherent in the automata used.

Although the approaches mentioned above focus on formal representations of normative documents, there is work on using them for natural language texts, from using controlled natural languages e.g. see [3] to free text contracts e.g. see [2, 5, 4].

The approach we present in this paper takes a deontic logic-agnostic approach to characterising conflicts, but furthermore, we extend the analysis to take into account environmental constraints which may give rise to conflicts.

3. A General Model of Temporal Deontic Logics and Constraints

We assume that the normative texts will be with reference two parties $\mathbb{P} \stackrel{df}{=} \{1, 2\}$. We will use variables p, p' to range over $\mathbb{P}$, and will write $\overline{p}$ to refer to the party other than p.

Since we will be looking at action-based deontic logics, we will assume an alphabet Σ of possible actions, with variables a, a' ranging over this alphabet. In order to identify which party has attempted (or performed) an action, we will tag actions by the participating party: $\Sigma_{\mathbb{P}} \stackrel{df}{=} \{a_p \mid a \in \Sigma,\ p \in \mathbb{P}\}$. Furthermore, in order to enable multiple actions occurring simultaneously, we will look at actions sets ranging over the power set of the alphabet $2^{\Sigma_{\mathbb{P}}}$ which we will refer to as $\mathbb{\Sigma}_{\mathbb{P}}$, with variables A and A' ranging over this type. Finally, we will also need to refer to finite sequences of action sets $\mathbb{\Sigma}_{\mathbb{P}}^*$, using variables $\overline{A}$ and $\overline{A}'$ to range over them.

3.1. Deontic logics

Since our intention is to develop a general framework for action-based deontic logics in general, rather than for a particular one, we distill requirements for conflict analysis to a number of basic predicates and relations over the underlying logic we wish to analyse.

A deontic logic can be characterised by a set of well-formed formulae *DeonticFormula* in the logic, using variables ψ, ψ' to range over these well-formed formulae. We will be dealing with deontic logics which include a notion of discrete time, and will assume a derivative operational relation [7] expressing how a formula evolves when a set of actions is performed, writing $\psi \xrightarrow{A} \psi'$ to denote that upon receiving set of actions A, the residual formula of ψ (the new formula encoding the state of the contract) is ψ'. For instance, in the contract logic $\mathcal{CL}$ [1], one may write the contract $[a]\mathcal{O}(b)$ to indicate that *'if a is initially performed, then on obligation to perform b is enacted, and if not, no residual contract remains.'* The derivative relation would include $[a]\mathcal{O}(b) \xrightarrow{\{a\}} \mathcal{O}(b)$, $[a]\mathcal{O}(b) \xrightarrow{\{a,c\}} \mathcal{O}(b)$ and $[a]\mathcal{O}(b) \xrightarrow{\{c\}} \top$. We will write $\psi \xRightarrow{\overline{A}} \psi$ to indicate the transitive closure of single step derivatives over action set trace $\overline{A}$.

At the core of all deontic logics, we require the underlying normative literal clauses which identify immediate norms in force and which the logic can handle. For instance, standard deontic logic [11] and the contract language $\mathcal{CL}$ would include the notion of obligation to perform action a: $\mathcal{O}(a)$, prohibition to perform action a: $\mathcal{F}(a)$ and permission to do so: $\mathcal{P}(a)$. In contract automata [6], these literals are parametrised by the party to whom they apply: $\mathcal{O}_p(a)$, $\mathcal{F}_p(a)$ and $\mathcal{P}_p(a)$. In all these cases, the norm can also be applied to absence of an action e.g. in contract automata one can have $\mathcal{O}_p(!a)$. We assume a set of norm literals NormLiteral, and use variable ∂ to range over this set. Furthermore, we will assume the notion of the norm opposing ∂, written $!\partial$, with the operator being a partial function from NormLiteral to NormLiteral such that, when well defined, $!!\partial = \partial$. For instance, many deontic logics would have $!\mathcal{P}(a) = \mathcal{F}(a)$ and $!\mathcal{O}(a) = \mathcal{P}(!a)$. Given a set of norm literals in force $\mathcal{D} \subseteq$ NormLiteral, we will assume that there is a predicate $vio_A(\mathcal{D})$ which holds if action set A is in violation of literal set $\mathcal{D}$[2].

[2] It is worth noting that this cannot always be reduced to a relation on single norm literals. For instance, in interactive systems, the counter-party of a permission must provide an action set which contains *all* permitted actions, not just an action set for each permitted actions. See [6] for more details.

For a deontic logic we will assume that we have a way to extract the set of normative literals $\mathcal{D} \subseteq \text{NormLiteral}$ that are in force upon enacting that deontic formula, through the function: $\text{norms}_0 \in \mathit{DeonticFormula} \rightarrow 2^{\text{NormLiteral}}$. For example, in $\mathcal{CL}$, the formula $\mathcal{O}(a) \wedge [b]\mathcal{F}(c)$ only enforces an obligation to perform a now, with prohibition to perform c only to be enacted if b is initially performed. Thus, we would expect that $\text{norms}_0(\mathcal{O}(a) \wedge [b]\mathcal{F}(c)) = \{\mathcal{O}(a)\}$.

Definition 1. *A temporal deontic logic over alphabet Σ is characterised as a tuple $\langle \Sigma, \mathit{NormLiteral}, \mathit{DeonticFormula}, \mathit{norms}_0, \mathit{vio}, \mapsto \rangle$, where (i) NormLiteral are the basic underlying norm literals the logic can express; (ii) DeonticFormula is the set of well-formed formulae in the logic; (iii) $\mathit{norms}_0 \in \mathit{DeonticFormula} \rightarrow 2^{\mathit{NormLiteral}}$ is a function giving the norms in immediate effect; (iv) $\mathit{vio} \in \Sigma_{\mathbb{P}} \times 2^{\mathit{NormLiteral}} \rightarrow \mathbb{B}$ is the violation predicate which formalises when an action set violates a set of deontic literals; and (v) $\mapsto \in \mathit{DeonticFormula} \times \Sigma_{\mathbb{P}} \rightarrow \mathit{DeonticFormula}$ is the derivative function expressing how the logic evolves over occurrence of actions.*

In the rest of the paper, we overload the violation relation to well-formed formulae: $\mathit{vio}_A(\psi) \overset{df}{=} \mathit{vio}_A(\mathit{norms}_0(\psi))$.

Many of the temporal deontic logics in the literature have been given such an operational semantics. For instance, the contract logic $\mathcal{CL}$ [1] was given semantics in this form in [9]. $\mathcal{CL}$ is given a trace semantics from which one can easily calculate the derivative function. The trace semantics also carry information as to which norms are in force, which provides for a definition of the norms_0 function. Finally, the violation predicate corresponds to action sets which would reduce the contract formula to $\perp$ (the implicitly violated contract in $\mathcal{CL}$).

Similarly, contract automata [6] formalise contracts between interacting parties as deterministic automata with (i) transitions labelled by sets of actions; and (ii) states tagged by a set of norm literals which are in force when in that state. There is a direct correspondence between the set of *DeonticFormula* with the states in the contract automaton, the derivative function with the transition relation of the automata and set of norms in immediate effect corresponding to the norms in the state of the automaton one is in. Contract automata already provide a violation predicate which assesses whether an action set is in violation of the norm literals in the current state, which can be used directly for the *vio* predicate.

In order to enable a complete axiomatisation of conflicts, we have to take into consideration the fact that some norm literals are related together. For instance, in interacting system agreements, an obligation on one party to perform an action subsumes permission to perform that action i.e. the other party is required to provide the necessary handshake to allow the action to happen [17]. In order to characterise such relations between norm literals, we will assume a strictness relation between sets of normative literals $\sqsubseteq$, such that $\mathcal{D} \sqsubseteq \mathcal{D}'$ holds if and only if $\mathcal{D}'$ is at least as strict as $\mathcal{D}$ i.e. would lead to at least as many violations. We assume a number of properties of the strictness relation:

Assumption 1. *(i) The strictness relation $\sqsubseteq$ is a partial order with $\emptyset$ being a minimum; (ii) If $\mathcal{D}' \sqsubseteq \mathcal{D}''$, then for any $\mathcal{D}$, $\mathcal{D} \cup \mathcal{D}' \sqsubseteq \mathcal{D} \cup \mathcal{D}''$.*

It is worth noting that from these assumed properties, it immediately follows that adding extra clauses can only make a contract stricter: $\mathcal{D} \sqsubseteq \mathcal{D} \cup \mathcal{D}'$.

3.2. Action predicates

In order to define constraints on the context, which limit what sets of actions are possible, we will use a boolean expressions over actions — such that action set A is possible if and only if the boolean expression holds when variables in A are instantiated to true, and all others to false. For example, to express the constraint that actions a and b are mutually exclusive, we would write this as $\neg(a \wedge b)$. Similarly, if we want to express the constraint that action a cannot appear unless b also appears, we would write $\neg b \implies \neg a$. As a final example, we can express that actions a, b and c cannot all appear together as $\neg(a \wedge b \wedge c)$.

Definition 2. *An* action constraint $\alpha \in \mathit{ActionConstraint}$ *is a boolean expression over actions defined using the following syntax:* $\alpha ::= \bot \mid \Sigma_{\mathbb{P}} \mid \neg\alpha \mid \alpha \vee \alpha$. *An action constraint corresponds to a collection of action sets defined as follows*[3]*:*

$$\begin{aligned}
[\![\bot]\!] &\stackrel{df}{=} \emptyset \\
[\![a]\!] &\stackrel{df}{=} \{A \mid A \subseteq \Sigma_{\mathbb{P}} \wedge a \in A\} \\
[\![\neg\alpha]\!] &\stackrel{df}{=} [\![\alpha]\!]^c \\
[\![\alpha \vee \alpha']\!] &\stackrel{df}{=} [\![\alpha]\!] \cup [\![\alpha']\!]
\end{aligned}$$

We will define other standard boolean operators in the usual manner: $\alpha \wedge \alpha' \stackrel{df}{=} \neg(\neg\alpha \vee \neg\alpha')$, $\alpha \implies \alpha' \stackrel{df}{=} \neg\alpha \vee \alpha'$ *and* $\alpha \iff \alpha' \stackrel{df}{=} (\alpha \implies \alpha') \wedge (\alpha' \implies \alpha)$.

We will overload the violation predicate, writing $vio_\alpha(\mathcal{D})$ *to indicate that all interpretations satisfying* α *violate* $\mathcal{D}$: $\forall A \in [\![\alpha]\!] \cdot vio_A(\mathcal{D})$.

An action constraint α *is said to be stronger than another action constraint* α', *written* $\alpha \vdash \alpha'$ *if and only if* $[\![\alpha]\!] \subseteq [\![\alpha']\!]$.

Since action constraints can vary over time (e.g. once locked, the door cannot be unlocked without the keycard being swiped), we extend action constraints temporarily using an approach similar to the way we expressed temporal deontic logics i.e. a transition system made up of well-formed formulae in *TemporalActionConstraint*, such that for a formula $\tau \in \mathit{TemporalActionConstraint}$, $\mathrm{constraint}_0(\tau)$ gives the action constraint initially in force for formula τ, and $\mapsto$ is the temporal derivative function for the transition system.

Definition 3. *A* temporal action constraint language *over alphabet* Σ *is characterised as a tuple* $\langle\Sigma, \mathit{TemporalActionConstraint}, \mathit{constraint}_0, \mapsto\rangle$, *where (i) TemporalActionConstraint is the set of well-formed formulae in the temporal action constraint language; (ii)* $\mathit{constraint}_0 \in \mathit{TemporalActionConstraint} \rightarrow \mathit{ActionConstraint}$ *is a function giving the action constraint in effect at the beginning; (iii)* $\mapsto \in \mathit{TemporalActionConstraint} \times \Sigma_{\mathbb{P}} \rightarrow \mathit{TemporalActionConstraint}$ *is the derivative function expressing how formula in the language evolves over occurrence of actions.*

For instance, consider the use of the Safety Linear Time Logic (Safety LTL) [21] as a temporal action constraint language, which would allow us to express the constraint on environment behaviour that *once locked, the door cannot be unlocked without the key-*

[3] We write S^c to denote the complement of set S.

card being swiped as the formula *Door*, defined to be $G(lock \implies (\neg unlock\ W\ swipe))^4$. Derivatives of Safety LTL formulae can be defined in a standard manner as used for runtime verification e.g. see [14] whilst the action constraint is taken to be the weakest formula which, if it holds, would reduce the LTL formula to *false*. For instance, the derivative of formula $G(lock \implies (\neg unlock\ W\ swipe))$ with respect to $\{lock\}$ would be $(\neg unlock\ U\ swipe) \wedge Door$, whilst $\text{constraint}_0(Door) = \neg(lock \wedge unlock \wedge \neg swipe)$ indicating that the door is not allowed to be locked and unlocked immediately.

As in the case of deontic logic formulae we showed earlier, we write $\tau \overset{\overline{A}}{\Rightarrow} \tau$ to indicate the transitive closure of single step derivatives over action set trace $\overline{A}$.

4. Deontic Conflicts

Much of the literature conflicts are formalised as a binary relation between contracts. For instance, in [6], we used a binary relation between norms $\psi \maltese \psi'$, defined to contain (i) conflicts between opposite norms, (ii) obligations to perform mutually exclusive actions, and (iii) defined to be closed under symmetry and increased strictness. This approach works well since the only context constraint is that of pairwise mutually exclusive actions. However, when we enrich the class of constraints which may be used, conflicts between pairs of norm clauses no longer suffices.

Consider, for example, an environmental constraint which ensures that the three actions a, b and c can never occur together. Now consider the normative clauses which place an obligation on party p to perform each action separately: $O_p(a)$, $O_p(b)$ and $O_p(c)$. No two of these three clauses conflict with each other under the environmental constraint, but the three together are in conflict since they clearly cannot be satisfied together.

In order to deal with conflicts under such an enriched class of environmental constraints have to talk about conflicts over a *set* of normative clauses rather than limiting it to two clauses i.e. deducing that under the context constraint mentioned above, the set of normative clauses $\{O_p(a), O_p(b), O_p(c)\}$ is in conflict.

4.1. Conflicts in norm literal sets

We can now define the notion of conflict within a set of normative clauses as a predicate over sets of norm literals. Such a set of norms is in conflict if (i) there are opposing norm literals or (ii) if the context constraints result in an unsatisfiable contract. In addition, conflicts are closed under (i) increased strictness of the norms; and (ii) strengthening of constraints. These four principles provide the

Definition 4. *A set of norm literals $\mathcal{D}$ is said to have an internal conflict under context constraint χ, written $\maltese_\chi(\mathcal{D})$, if it follows from the following axioms:*
Axiom 1: *Opposing literal norms conflict: $\maltese_{true}(\{\partial, !\partial\})$.*
Axiom 2: *Conflicts may arise from norms impossible to satisfy due to action constraints: if $vio_\chi(\mathcal{D})$ then $\maltese_\chi(\mathcal{D})$.*
Axiom 3: *Conflicts are closed under increased strictness: if $\maltese_\chi(\mathcal{D})$ and $\mathcal{D} \sqsubseteq \mathcal{D}'$, then $\maltese_\chi(\mathcal{D}')$.*
Axiom 4: *Conflicts are closed under constraint strengthening: if $\chi' \vdash \chi$ and $\maltese_\chi(\mathcal{D})$, then $\maltese_{\chi'}(\mathcal{D})$.* □

[4] In LTL, property $G\pi$ holds for a trace if and only if π holds for any suffix of the trace, $\pi\ W\ \pi'$ holds if and only if π will hold on every suffix of the trace until π' holds (although π' may never hold, in which case π must continue holding indefinitely) and $X\ \pi$ which holds if π holds if we ignore the first event in the trace.

Consider a deontic norm set with obligations on all three actions a, b and c: $\mathcal{D} = \{\mathcal{O}_p(a),\ \mathcal{O}_p(b),\ \mathcal{O}_p(c)\}$ and the constraint: $\chi = a \wedge b \implies \neg c$. We can show that $vio_\chi(\mathcal{D})$ and thus, by Axiom 2 that $\maltese_\chi(\mathcal{D})$.

As another example, consider a norm set which includes an obligation and prohibition to perform the same action: $\mathcal{O}_p(a)$ and $\mathcal{F}_p(a)$. Firstly note that in a deontic logic in which permission is weaker than obligation i.e. $\mathcal{P}_p(a) \sqsubseteq \mathcal{O}_p(a)$, the contrapositive holds: $!\mathcal{O}_p(a) \sqsubseteq !\mathcal{P}_p(a)$. Now, by Axiom 1, $\maltese_\chi(\{\mathcal{O}_p(a),\ !\mathcal{O}_p(a)\})$, from which (and the inequality just given) it follows that $\maltese_\chi(\{\mathcal{O}_p(a),\ !\mathcal{P}_p(a)\})$ using Axiom 3. However, $\mathcal{F}_p(a)$ is equivalent to $!\mathcal{P}_p(a)$, from which we can conclude that: $\maltese_\chi(\{\mathcal{O}_p(a),\ \mathcal{F}_p(a)\})$ which can be extended to any set containing these norm literals using Axiom 4 and Assumption 1 of the strictness relation: $\maltese_\chi(\{\mathcal{O}_p(a),\ \mathcal{F}_p(a)\} \cup \mathcal{D})$.

4.2. Temporal conflicts

The notion of internal conflicts can be extended beyond sets of norm literals by ensuring conflicts never arise no matter what input is received.

Definition 5. *Given a temporal deontic logic formula $\psi \in DeonticFormula$ and temporal action constraint $\tau \in TemporalActionConstraint$, we say that ψ has a conflict under constraint τ, written $\maltese_\tau(\psi)$ if, for some action set trace, the immediate deontic norms of the derivative deontic formula are in conflict under the immediate derivative constraint:*

$$\begin{aligned}\maltese_\tau(\psi) \overset{df}{=} \exists \overline{A} \in \Sigma_{\mathbb{P}}^* \cdot \forall \psi' \in DeonticFormula \cdot \forall \tau' \in TemporalActionConstraint\cdot \\ \psi \overset{\overline{A}}{\Rightarrow} \psi' \wedge \tau \overset{\overline{A}}{\Rightarrow} \tau' \implies vio_{constraint_0(\tau')}(norms_0(\psi'))\end{aligned}$$

□

Using this definition, for instance, we can discover that under an environmental constraint that says that after action a, b and c cannot occur together: $\tau = G(a \implies \neg X\,(b \wedge c))$, the $\mathcal{CL}$ formula $\psi = [a](O(b) \wedge O(c))$ is in conflict. Using the singleton trace $\overline{A} = \langle\{a\}\rangle$, we can show that using derivatives of $\mathcal{CL}$ and LTL: $\psi \overset{\overline{A}}{\Rightarrow} O(a) \wedge O(b)$ and $\tau \overset{\overline{A}}{\Rightarrow} \neg(b \wedge c) \wedge \tau$. Using the definitions of constraint$_0$ and norms$_0$, we can show that a conflict arises by proving that $vio_{\neg(b \wedge c)}(\{O(a),\ O(b)\})$ which follows from the previous definitions.

5. Use Case: An Internet Service Provider Contract in $\mathcal{CL}$

In order to investigate the use of the conflict analysis techniques we describes, we have use an instantiation of our contract conflict theory for $\mathcal{CL}$, and extended an Internet Service Provider contract from [16]. The contract between the service provider and the client, shown in Figure 1. In particular, note the client information deletion parts of the contract:

10(b) *The **Provider** is obliged to delete all the **Client**'s information within a period of five (5) days of the **Client** requesting to close their account.*

10(c) *As long as the **Client**'s account is open, the **Provider** is obliged to keep the client's information.*

13(b) *The **Provider** is obliged to close the **Client**'s account within a period of three (3) days of the **Client** submitting a request.*

This deed of **Agreement** is made between:

1. **[name]**, from now on referred to as **Provider** and
2. **[name]**, from now on referred to as the **Client**.

INTRODUCTION

3. The **Provider** is obliged to provide the **Internet Services** as stipulated in this **Agreement**.

DEFINITIONS

1. **Internet traffic** may be measured by both **Client** and **Provider** by means of equipment and may take the two values **high** and **normal**.

OPERATIVE PART

7. CLIENT'S RESPONSIBILITIES AND DUTIES
   (a) The **Client** shall not:
      i. supply false information to the Client Relations Department of the **Provider**.
   (b) Whenever the Internet Traffic is **high** then the **Client** must pay [*price*] immediately, or the **Client** must notify the **Provider** by sending an e-mail specifying that he will pay later.
   (c) If the **Client** delays the payment as stipulated in 7b, after notification he must immediately lower the Internet traffic to the **normal** level, and pay later twice (2 ∗ [*price*]).
   (d) If the **Client** does not lower the Internet traffic immediately, then the **Client** will have to pay 3 ∗ [*price*].
   (e) The **Client** shall, as soon as the Internet Service becomes operative, submit within seven (7) days the Personal Data Form from his account on the **Provider**'s web page to the Client Relations Department of the **Provider**.
8. CLIENT'S RIGHTS
   (a) The **Client** may choose to pay either: (i) each month; (ii) each three (3) months; (iii) each six (6) months;
9. PROVIDER'S SERVICE
   (b) As part of the Service offered by the **Provider** the **Client** has the right to an e-mail and an user account.
   (c) **Provider** is obliged to offer with no limitation and within a period of seven (7) days a password and any other equipment specific to the Client, necessary for the correct usage of the user account, upon receiving of all the necessary data about the client from the Client Relations Department of the **Provider**.
   (d) Each month the **Client** pays the *bill* the **Provider** is obliged to send a Report of Internet Usage to the Client.
10. PROVIDER'S DUTIES
   (a) The **Provider** guarantees that the Client Relations Department, as part of his administrative organization, will be responsive to requests from the **Client** or any other Department of the **Provider**, or the **Provider** itself within a period less than two (2) hours during *working hours* or the day after.
   (b) The **Provider** is obliged to delete all the **Client**'s information within a period of five (5) days of the **Client** requesting to close their account.
   (c) As long as the **Client**'s account is open, the **Provider** is obliged to keep the client's information.
11. PROVIDER'S RIGHTS
   (a) The **Provider** takes the right to alter, delete, or use the *personal data* of the **Client** only for statistics, monitoring and internal usage in the confidence of the **Provider**.
   (b) **Provider** may, at its sole discretion, without notice or giving any reason or incurring any liability for doing so:
      ii. Suspend Internet Services immediately if **Client** is in breach of Clause 7a;
13. TERMINATION
   (a) Without limiting the generality of any other *Clause* in this *Agreement* the **Client** may terminate this *Agreement* immediately without any notice and being vindicated of any of the Clause of the present Agreement if:
      i. the **Provider** does not provide the Internet Service for seven (7) days consecutively.
   (b) The **Provider** is obliged to close the **Client**'s account within a period of three (3) days of the **Client** submitting a request.
   (c) The **Provider** is forbidden to terminate the present Agreement without previous written notification by normal post and by e-mail.
   (d) The **Provider** may terminate the present Agreement if: any payment due from **Client** to **Provider** pursuant to this **Agreement** remains unpaid for a period of fourteen (14) days;
16. GOVERNING LAW
   (a) The **Provider** and the present **Agreement** are governed by and construed according to the Law Regulating Internet Services and to the Law of the State.
      i. The Law of the State stipulates that any **ISP Provider** is obliged, upon request to seize any activity until further notice from the State representatives.

Figure 1. Extracts from a contract between an internet service provider and their client

Using days for the unit of discrete time, we can encode the contract in $\mathcal{CL}$ e.g. clause 13(b) would be formulated[5] as: $[?^* \cdot \mathit{requestTermination}]O((1+?+?\cdot?)\cdot \mathit{closeAccount})$. In addition, we have some logistical constraints arising from the technical setup of the service provider. Consider the following constraints about backups expressed in LTL:

$$\begin{array}{lll}
 & \neg \mathit{deleteClientInfoFromBackup}\ W\ \neg \mathit{deleteClientInfo} & (1)\\
\wedge & \mathit{deleteClientInfoFromBackup} \implies X(\neg \mathit{deleteClientInfoFromBackup}\ W\ \neg \mathit{deleteClientInfo}) & (2)\\
\wedge & G(\mathit{deleteClientInfoFromBackup} \implies \mathit{weeklyBackupInProgress}) & (3)\\
\wedge & G(\mathit{weeklyBackupInProgress} \implies \bigwedge_{1\leq i\leq 6} \neg X^i \mathit{weeklyBackupInProgress}) & (4)\\
\wedge & G(\neg(\mathit{requestCloseAccount} \wedge \mathit{deleteClientInfo})) & (5)
\end{array}$$

[5] The formula says that any time the action guard (written in a regular expression like syntax, with ? meaning 'any action', star means repetition, + means choice and · indicates sequential composition) is satisfied i.e. a request for termination has just been received, the obligation to close the account within three time units is enacted (the action sequence of the obligation is also written in regular expression-like syntax.

Constraints (1) and (2) indicate that client information is not deleted from the backup unless it is deleted from the main database first. Constraint (3) further indicates that deletion from the backup can only occur when the weekly backup is taking place. Constraint (4) indicates that weekly backups never occur less than a week apart and finally, (5) indicates that due to demands for termination being processed in batch at the end of the day, a client's information is never deleted immediately upon a request to close the account.

Analysing the $\mathcal{CL}$ contract under these constraints will allow us to discover a conflict arising when the client requests to close their account triggering (i) an obligation to delete the client's information within five days including that kept in the backup (arising from clause 10(b)); (ii) an obligation to close the account within three days (from clause 13(b)); and (iii) an obligation to keep the data until the account is closed (from clause 10(c)). If the client request happed on the same day as backups are done, the constraints result in the backup not being deleted until at least 7 days have elapsed of the request, with the service provider thus not being able to comply to the contract's terms.

The implication of the discovery of this conflict is that either the contract is to be fixed to remove the conflict, or the internal systems must be changed in order to relax the constraints resulting in the conflict. The choice depends on the importance of the offending contract clauses and the possibility (or otherwise) of changing the constraints: Is it reasonable to start taking more frequent backups or to propagate deletions to the backups promptly, or is it simpler to extend the deletion period from five to 10 days?

6. Conclusions

In this paper we have presented a unified theory of temporal deontic contracts. Unlike previous work on the topic, we do not restrict ourselves to a particular deontic logic and, more importantly, allows taking into consideration temporal constraints on the actions taking place. We have illustrated the use of the theory by using its instantiation to the contract language $\mathcal{CL}$ and LTL for constraint expression in order to find conflicts in an Internet Service Provider contract taken from the literature.

We are currently looking at automated ways of automating the analysis and the modelling of different deontic and temporal logics in our model, including the use of symbolic model checking techniques in order to be able to explore contract sanity efficiently. Furthermore, our work can form the basis of conflict resolution, in order to refine or characterise text more effectively.

We also note that the theory can be extended to deal with other forms of contract analysis such as the detection of useless clauses — contract clauses which can be left out or simplified due to the fact that the environmental constraints will never have an effect e.g. a clause which states that *'the provider is obliged to either delete the data from the main database immediately, or to notify the user and delete it within 1 day'* can be simplified to *'the provider is obliged to notify the user and delete their data from the main database within 1 day'* due to the constraint that deletion cannot happen on the same day as a request for closing an account.

References

[1] A dynamic deontic logic for complex contracts. *The Journal of Logic and Algebraic Programming*, 81(4):458 – 490, 2012.

[2] João Paulo Aires, Daniele Pinheiro, Vera Strube de Lima, and Felipe Meneguzzi. Norm conflict identification in contracts. *Artif. Intell. Law*, 25(4):397–428, 2017.

[3] Krasimir Angelov, John J. Camilleri, and Gerardo Schneider. A framework for conflict analysis of normative texts written in controlled natural language. *J. Log. Algebraic Methods Program.*, 82(5-7):216–240, 2013.

[4] Shaun Azzopardi, Albert Gatt, and Gordon J. Pace. Integrating natural language and formal analysis for legal documents. In *Proceedings of the 10th Conference on Language Technologies and Digital Humanities 2016*, 2016.

[5] Shaun Azzopardi, Albert Gatt, and Gordon J. Pace. Reasoning about partial contracts. In *Legal Knowledge and Information Systems - JURIX 2016: The Twenty-Ninth Annual Conference*, pages 23–32, 2016.

[6] Shaun Azzopardi, Gordon J. Pace, Fernando Schapachnik, and Gerardo Schneider. Contract automata - an operational view of contracts between interactive parties. *Artif. Intell. Law*, 24(3):203–243, 2016.

[7] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

[8] Xavier Parent Ron van der Meyden Leendert van der Torre Dov Gabbay, John Horty. *Handbook of Deontic Logic and Normative Systems*.

[9] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. Automatic conflict detection on contracts. In *Proceedings of Theoretical Aspects of Computing ICTAC*, 2009.

[10] Stephen Fenech, Gordon J. Pace, and Gerardo Schneider. CLAN: A tool for contract analysis and conflict discovery. In *Proceedings of Automated Technology for Verification and Analysis, 7th International Symposium, ATVA*, 2009.

[11] Georg Henrik Von Wright. Deontic Logic. *Mind*, 60(237):1–15, January 1951.

[12] Daniel Gorín, Sergio Mera, and Fernando Schapachnik. Model checking legal documents. In *The Twenty-Third Conference on Legal Knowledge and Information Systems (JURIX)*, volume 223 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2010.

[13] Guido Governatori and Robert Mullins. Deontic closure and conflict in legal reasoning. In *Legal Knowledge and Information Systems (JURIX)*, volume 322 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2019.

[14] Klaus Havelund and Grigore Rosu. Monitoring programs using rewriting. In *16th IEEE International Conference on Automated Software Engineering (ASE 2001), 26-29 November 2001, Coronado Island, San Diego, CA, USA*, pages 135–143. IEEE Computer Society, 2001.

[15] Enrique Martínez and Gerardo Schneider. Automated analysis of conflicts in software product lines. In *Software Product Lines - 14th International Conference, SPLC 2010, Jeju Island, South Korea, September 13-17, 2010. Workshop Proceedings (Volume 2 : Workshops, Industrial Track, Doctoral Symposium, Demonstrations and Tools)*, pages 75–82, 2010.

[16] Gordon J. Pace, Cristian Prisacariu, and Gerardo Schneider. Model checking contracts - A case study. In *Automated Technology for Verification and Analysis, 5th International Symposium, ATVA*, volume 4762 of *Lecture Notes in Computer Science*.

[17] Gordon J. Pace and Fernando Schapachnik. Permissions in contracts, a logical insight. In *Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference, University of Vienna, Austria, 14th-16th December 2011*, pages 140–144, 2011.

[18] Seyed-Ali Sadat-Akhavi. *Methods of Resolving Conflicts between Treaties*. Number Vol. 3 in Graduate Institute of International Studies (Series). Leiden: Brill Academic Publishers, 2003.

[19] Silvano Colombo Tosatto, Guido Governatori, and Pierre Kelsen. Detecting deontic conflicts in dynamic settings. In *Proceedings of Deontic Logic and Normative Systems DEON*, volume 8554 of *Lecture Notes in Computer Science*.

[20] W. Weber Vasconcelos, M.J. Kollingbaum, and T.J. Norman. Normative conflict resolution in multi-agent systems. *Auton. Agents Multi Agent Syst.*, 19(2):124–152, 2009.

[21] Shufang Zhu, Lucas M. Tabajara, Jianwen Li, Geguang Pu, and Moshe Y. Vardi. A symbolic approach to safety ltl synthesis. In *Hardware and Software: Verification and Testing - 13th International Haifa Verification Conference, HVC 2017*.

[22] G. Governatori, F. Olivieri, M. Cristani, and S. Scannapieco. Revision of defeasible preferences. *International Journal of Approximate Reasoning*, 104:205–230, 2019.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200853

Free Choice Permission in Defeasible Deontic Logic

Guido GOVERNATORI [a,1] Antonino ROTOLO [b]
[a] *Data61, CSIRO, Australia*
[b] *Alma Human AI, University of Bologna, Italy*

Abstract. Free Choice Permission is one of the challenges for the formalisation of norms. In this paper, we follow a novel approach that accepts Free Choice Permission in a restricted form. The intuition behind the guarded form is strongly aligned with the idea of defeasibility. Accordingly, we investigate how to model the guarded form in Defeasible Deontic Logic extended with disjunctive permissions.

Keywords. Free Choice Permission, Disjunctive Permissions, Defeasible Deontic Logic

1. Introduction

Free Choice Permission is one of the problems of Deontic Logic where there seems to be a mismatch between the intuition and its formalisation in a deontic language. The problem arises when there is a disjunctive permission, meaning that that one is permitted to chose between two alternatives, let us say A and B. The intuition suggests that both alternatives are individually permitted, so A is permitted *and* B is permitted. Formally, we can represent it by the schema

$$\mathsf{P}(A \vee B) \rightarrow (\mathsf{P}A \wedge \mathsf{P}B) \tag{1.1}$$

The schema is not generally valid in Deontic Logic and, when added as an axiom, it leads to the so-called permission explosion problem [9]: everything is permitted whenever there is a disjunctive permission.

Several logics have been devised to reconcile the intuition with the formal representation for the Free Choice Permission. However, some recent work [7] casts some doubts about the validity of the schema; the following scenario provides a counter-example.

Example 1. When you have dinner with guests, the etiquette allows you to eat or to have a conversation with your fellow guests, but speaking while eating is forbidden.

The example can be formalised as

$$\mathsf{P}(\mathit{eat} \vee \mathit{speak}) \tag{1.2}$$

$$\mathit{eat} \rightarrow \mathsf{O}\neg \mathit{speak} \tag{1.3}$$

$$\mathit{speak} \rightarrow \mathsf{O}\neg \mathit{eat} \tag{1.4}$$

When one performs one of the two alternatives in the disjunctive permission, the other alternative becomes forbidden, and this leads to a contradiction if one assumes (1.1) as an

[1] Corresponding author: Guido Governatori, e-mail: guido.governatori@data61.csiro.au

axiom. This seems to support the view that free choice permission is a pseudo-problem that can be solved by explicitly representing the conjunction when the natural language description supports such a reading [9]. However, the following example [7] indicates that normative reasoning requires the ability to derive an individual permission from a disjunctive permission.

Example 2. A shop has the following policy for clothes bought online. If the size of an item is not a perfect fit, then the customer is entitled to either exchange the item for free, or keep the item and receive a $10 refund. However, customers electing to keep the item are not entitled to the refund, and customers opting for the refund are not entitled to exchange the item for free. Furthermore, customers who elect to exchange the item (when entitled to do so) have to return it with the original package.

The representation of this scenario is similar to that of Example 1, namely:

$$\neg \mathit{perfectFit} \rightarrow \mathsf{P}(\mathit{return} \lor \mathit{refund}) \tag{1.5}$$

$$\mathit{return} \rightarrow \mathsf{O}\neg \mathit{refund} \tag{1.6}$$

$$\mathit{refund} \rightarrow \mathsf{O}\neg \mathit{return} \tag{1.7}$$

$$\mathsf{P}\mathit{return} \land \mathit{return} \rightarrow \mathsf{O}\mathit{original} \tag{1.8}$$

When a garment is not a perfect fit, and the customer elects the option to return the item, then, intuitively, the permission of return it holds. However, without Free Choice Permission (or a guarded version of the principle), it is not possible to formally derive the permission, and then we cannot conclude the obligation that it must be returned with the original package.

Based on the discussion so far, it seems that there is the need for a mechanism to derive individual permissions from a disjunctive permission, but the mechanism should be guarded to prevent the derivation of contradictions when some of the alternatives are forbidden. Accordingly, [7] proposed the following guarded version of Free Choice Permission:

From $\mathsf{P}(A \lor B)$ to $\mathsf{P}A$ and $\mathsf{P}B$ provided that it is not the case that $\mathsf{O}\neg A$ or $\mathsf{O}\neg B$.

Furthermore, [7] devised a set of axioms to extend standard classical propositional Deontic Logic to avoid the problem typically caused by Free Choice Permission.

The principle advanced by [7] for a guarded version of Free Choice Permission has an intrinsic defeasible nature. In Defeasible Logic a defeasible conclusion is provable, if there is an argument (rule) in favour, and no arguments for the opposite apply. For Free Choice Permission, the choice among alternatives (disjunctive permission) is the argument in favour of an individual permission (for one of the alternatives) and the inability to derive the prohibition (a prohibition is the opposite of a permission) means that the arguments for the opposite do not hold.

The examples we have provided in this section indicates that there is the need for the formal representation of disjunctive permissions, and that the guarded version of Free Choice Permission is a natural inference pattern of normative reasoning. In the next section, we are going to see how to extend Defeasible Deontic Logic to accommodate such a reasoning pattern.

2. Defeasible Deontic Logic

Defeasible Deontic Logic [5] (DDL) is a sceptical computationally oriented rule-based formalism designed for the representation of norms. The logic extends Defeasible Logic [1] with deontic operators to model obligations and (different types of) permissions, and provides an integration with the logic of violation proposed in [8]. The resulting formalism offers features for the natural and efficient representation of exceptions, constitutive and prescriptive rules, and compensatory norms. The logic is based on a constructive proof theory that allows for full traceability of the conclusions, and flexibility to handle and combine different facets of non-monotonic reasoning. To keep efficiency feasible the language is restricted to literals (atomic propositions and their negation) and deontic literals (literals in the scope of a deontic modality). In what follows, we are going to expand the language to cover disjunctive permissions, and we are going to revise the proof theory to accommodate the extended language. The revised proof conditions are a natural generalisation of the standard proof conditions.

Accordingly, we consider a logic whose language is defined as follows.

Definition 1. Let PROP be a set of propositional atoms, and O, P, P_s and P_w the modal (deontic) operators for obligation, permission, strong permission and weak permission respectively. The set Lit $= \text{PROP} \cup \{\neg p \mid p \in \text{PROP}\}$ denotes the set of *literals*. $\sim q$ denotes the *complement* of a literal q: if q is a positive literal p, then $\sim q$ is $\neg p$, and if q is a negative literal $\neg p$, then $\sim q$ is p. The set of *deontic literals* is DLit $= \{\Box l, \neg\Box l \mid l \in \text{Lit}, \Box \in \{\mathsf{O}, \mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}\}$. If $c_1, \ldots, c_n \in$ Lit, then for $\Box \in \{\mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$, $\Box(c_1 \vee \cdots \vee c_n)$ is a disjunctive (strong, weak) permission.

We adopt the standard DL definitions of *strict rules*, *defeasible rules*, and *defeaters* [1]. However,

For the sake of simplicity (and space limitations), and to better focus on the non-monotonic aspects that DDL offers, we only consider defeasible rules; however, the idea presented in the this paper can be easily accommodated in the proof conditions for the full language (see,[5]) to accommodate strict rules and defeaters.

Definition 2. Let Lab be a set of arbitrary labels. Every rule is of the type '$r\colon A(r) \hookrightarrow C(r)$', where $r \in$ Lab is the name of the rule; $A(r) = \{a_1, \ldots, a_n\}$, the *antecedent* (or *body*) of the rule, is the set of the premises of the rule (alternatively, it can be understood as the conjunction of all the elements in it). Each a_i is either a literal, a deontic literal, a disjunctive obligation or a disjunctive permission; The set of rules is partitioned in three sets of rules, where the arrow $\hookrightarrow \in \{\Rightarrow, \Rightarrow_{\mathsf{O}}, \Rightarrow_{\mathsf{P}}\}$ indicates the type of rules: constitutive rule ($\Rightarrow$), prescriptive rules ($\Rightarrow_{\mathsf{O}}$) and permissive rules ($\Rightarrow_{\mathsf{P}}$). $C(r)$, the *consequent* (or *head*) of the rule, is a single literal in case of constitutive rules and prescriptive rules, and a set of literals (intended to be read as a disjunction) in case of permissive rules.

Constitutive rules derive institutional facts, i.e., propositions understood to hold as defined in the underlying normative system. Prescriptive rules are to determine what obligations are in force in the normative system. In contrast, permissive rules generate the permissions in force in the normative system.

The key aspect of the logic is that we consider strong permissions as explicit derogations of obligations to the contrary. Thus, for an explicit permission, we need to have a rule that provides the conditions under which something is permitted. We also assume that 'obligation' implies 'strong permission', i.e.,

$$\mathsf{O}A \rightarrow \mathsf{P}_s A$$

To derive an obligation, we need to have a rule that makes it obligatory. In this view, the strongest way to derogate an obligation is to have a stronger rule that establishes that the same subject matter is forbidden (and we assume the usual interdefinabilty of obligations and prohibitions). For weak permission, we take the view that it is the lack of an obligation to the contrary (or lack of the prohibition). Thus, in case there are no rules for an obligation, then this will enable us to assess that the weak permission holds. The situation where we are not able to derive an obligation because a strong permission derogates it is another case where, in addition to the strong permission, we have the weak permission as well. Every time a strong permission holds, the corresponding weak permission holds a well. Hence, the logic satisfies

$$\mathsf{P}_s A \rightarrow \mathsf{P}_w A$$

Finally, we consider a generic permission, to be used when it is not clear if the permission is either strong or weak, that corresponds to the disjunction of the two, that is:

$$\mathsf{P}A \equiv \mathsf{P}_s A \vee \mathsf{P}_w A$$

Given a set of rules R, we use the following abbreviations for specific subsets of rules: $R[q]$ is the subset of R where q is an element of the consequent of the rules in R. R^c is the subset of the constitutive rules of R. R^{O} is the subset of the prescriptive rules of R. R^{P} is the subset of the permissive rules of R.

Definition 3. A *defeasible theory* is a structure $D = (F, R, >)$, where F, the set of facts, is a set of literals and modal literals, R is a set of rules and $>$, the superiority relation, is a binary relation over R.

A theory corresponds to a normative system, i.e., a set of norms, where for every norm there is a rule modelling it. The superiority relation is used for conflicting rules, i.e., rules whose conclusions are complementary literals, and it just determines the relative strength between the two rules.

Definition 4. A *proof* P in a defeasible theory D is a linear sequence $P(1) \ldots P(k)$ of *tagged literals* in the form of $+\partial q$, $-\partial q$, $+\partial_\Box q$, $-\partial_\Box q$ for $\Box \in \{\mathsf{O}, \mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$, $+\partial^*_\Diamond q_1 \vee \cdots \vee q_m$ and $-\partial^*_\Diamond q_1 \vee \cdots \vee q_m$ for $\Diamond \in \{\mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$ and $* \in \{f, max, min, \}$, where $P(1) \ldots P(k)$ satisfy the proof conditions given in Definitions 6–14.

The tagged literal $+\partial q$ means that q is *defeasibly provable* as an institutional statement, or in other terms, that q holds in the normative system encoded by the theory. The tagged literal $-\partial q$ means that q the normative system *defeasibly refutes* q. The tagged literal $+\partial_{\mathsf{O}} q$ means that q is *defeasibly provable* in D as an obligation, while $-\partial_{\mathsf{O}} q$ means that q is *defeasibly refuted* as an obligation; similarly for permission, with the difference that for permissions we will consider disjunctions as well. Thus, we can have tagged literals such as $+\partial_{\mathsf{P}_s} q_1 \vee \cdots \vee q_n$. The initial part of length i of a proof P is denoted by $P(1..i)$.

A rule is *applicable* for a literal q if q occurs in the head of the rule and all the elements in the antecedent of the rule have already been proved with the appropriate mode. A rule is *discarded* if at least one of the literals in the antecedent has not been proved.

Definition 5. Given a derivation P, rule $r \in R[q]$ is *applicable* at step $P(n+1)$ iff for all $a_i \in A(r)$, for $\Box \in \{\mathsf{O}, \mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$ and $\Diamond \in \{\mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$:

1. if $a_i = \Box l$ then $+\partial_\Box l \in P(1..n)$;
2. if $a_i = \neg\Box l$ then $-\partial_\Box l \in P(1..n)$;
3. if $a_i = \Diamond(c_1 \vee \cdots \vee c_m)$ then $+\partial_\Diamond c_1 \vee \cdots \vee c_m \in P(1..n)$;
4. if $a_i = l \in \mathrm{Lit}$ then $+\partial l \in P(1..n)$.

A rule $r \in R[q, j]$ is *discarded* iff $\exists a_i \in A(r)$ such that

1. if $a_i = \Box l$ then $-\partial_\Box l \in P(1..n)$;
2. if $a_i = \neg\Box l$ then $+\partial_\Box l \in P(1..n)$;
3. if $a_i = \Diamond(c_1 \vee \cdots \vee c_m)$ then $-\partial_\Diamond c_1 \vee \cdots \vee c_m \in P(1..n)$;
4. if $a_i = l \in \mathrm{Lit}$ then $-\partial l \in P(1..n)$.

The definitions of the negative tags can be obtained from the definitions of the corresponding positive tags by applying the principle of strong negation (that transforms the Boolean operators and quantifiers in their dual, and swapping "applicable" and "discarded" [2, 6]. For space reasons, we only provide the proof conditions for the positive tags; the exception is the prof condition for $-\partial$ that is given to illustrate the principle of strong negation.

Definition 6. The proof condition to establish when an *institutional statement is defeasibly provable* is defined as follows:
$+\partial$: If $P(n+1) = +\partial q$ then
(1) $q \in F$ or
  (2.1) $\sim q \notin F$ and
  (2.2) $\exists r \in R[q]$ such that r is applicable, and
  (2.3) $\forall s \in R[\sim q]$, either
    (2.3.1) s is discarded, or either
    (2.3.2) $\exists t \in R[q]$ such that t is applicable and $t > s$.

As usual, we use the strong negation to define the proof condition for $-\partial$. Defining the negative proof conditions based on the principle of strong negation ensures that we have a constructive procedure to establish the failure to attempt to satisfy the corresponding positive condition.

Definition 7. The proof condition to establish when an *institutional statement is defeasibly refutable* is defined as follows:

$-\partial$: If $P(n+1) = -\partial q$ then
(1) $q \notin F$ and
  (2.1) $\sim q \in F$ or
  (2.2) $\forall r \in R[q]$: eit her r is discarded, or
  (2.3) $\exists s \in R[\sim q]$, such that
    (2.3.1) s is applicable, and
    (2.3.2) $\forall t \in R[q]$ either t is discarded or not $t > s$.

The proof conditions for $\pm\partial$ are the standard conditions in defeasible logic, see [1] for the full explanations.

Definition 8. The proof condition to establish when an *obligation is defeasibly provable* is defined as follows:

$+\partial_{\mathsf{O}}$: If $P(n+1) = +\partial_{\mathsf{O}} q$ then
(1) $\exists r \in R^{\mathsf{O}}[p]$ such that r is applicable and
(2) $\forall s \in R[\sim p]$ either
  (2.1) s is discarded or
  (2.2) $s \in R^{\mathsf{P}}[\sim p]$ and $\exists q \in C(s), q \neq p, -\partial_{\mathsf{O}}{\sim}q \in P(1..n)$ or
  (2.3) $\exists t \in R[p]$ such that
    (2.3.1) t is applicable and $t > s$ and
      (2.3.2) if $t \in R^{\mathsf{P}}[p]$, then $\forall q \in C(t), q \neq p, +\partial_{\mathsf{O}}{\sim}q \in P(1..n)$.

To show that q is defeasibly provable as an obligation we require a prescriptive rule (norm) that is applicable (all the elements of the body of the rule have already been proved with the appropriate mode), Clause (1). Then we have to check that all rules that can generate a conclusion in conflict with the obligation are rebutted (Clause (2)). For an obligation, the rules we have to consider are the prescriptive and permissive rules for $\sim p$. One way to rebut the rule is to establish that the rule is discarded, meaning that at least one of the elements in the body of the rule has been refuted. The other way to rebut an attacking rule is to show that the attacking rule is weaker than an appropriate rule (this step, Clause 2.3, is known as reinstatement). In case we use a permissive rule for reinstatement, we have to ensure that the rule has the potential to derive the permission for the literal we want to prove. Suppose that we have a permissive rule for $a \vee b$, thus the rule would be able to conclude $\mathsf{P}(a \vee b)$, but we do not know which of the two options potentially hold. This means that we cannot use the permissive rule for $a \vee b$ in the reinstatement phase unless we know that all the options but the one in which we are interested are forbidden (Clause 2.3.2): if we want to prove $+\partial_{\mathsf{O}} a$, then $+\partial_{\mathsf{O}} \neg b$ is required, meaning that the prescriptive rule is effectively a rule for $\mathsf{P}a$. As we will see shortly, permissive rules with opposite conclusions are not in conflict with each other. Thus, it might be possible to argue that we cannot use a permissive rule in the reinstatement when the attacking rule is a permissive rule as well. For this variant, we can change the if condition in Clause 2.3.2 with the more restrictive "if $t \in R^{\mathsf{P}}[p]$ and $s \in R^{\mathsf{O}}[\sim p]$" to obtain the desired result.

We are now ready to provide the proof conditions under which disjunctive permissions (and then individual permissions) can be derived. This requires several steps.

First, we have to determine when a disjunctive permission corresponding to full consequent of a permissive norm is derivable. For this case, given in Definition 9, we have to see that there is an applicable permissive rule, and it is possible to perform at least one of the options without violating any other norm. In other terms, we have to see if we can refute as an obligation at least one of the literals corresponding to one of the permitted alternative. If this is the case, then the disjunctive permission can be assessed as a genuine permission.

Definition 9. The proof condition to establish when a *disjunctive permissive norm is defeasibly provable* is defined as follows:
If $P(n+1) = +\partial^{f}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m$, then
(1) $\exists r \in R^{\mathsf{P}}[p_1, \ldots, p_m]$ such that r is applicable $C(r) = \{p_1, \ldots, p_m\}$ and
(2) $\exists p_i \in C(r)$ such that $-\partial_{\mathsf{O}} p_i \in P(1..n)$.

The second case (Definition 10) is to determine what is the largest subset of disjuncts that are not forbidden. While the condition is given in term of a disjunction, the condition can be used to derive a single individual permission (when the permission is the only disjunct that is not forbidden). The idea behind this proof condition is similar to the process we described for the derivation of an obligation. The key aspect has two components: First, given an applicable disjunctive permissive rule, all the elements that are not in the disjunction we want to prove are provable as forbidden. Second, only obligation rules can be used to attack the disjunct to be included in the disjunction.

Definition 10. The proof condition to establish when a *maximal disjunctive permission is defeasibly provable* is defined as follows:
If $P(n+1) = +\partial^{max}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m$ then
(1) $+\partial_{\mathsf{O}} p_1, \cdots + \partial_{\mathsf{O}} p_m \in P(1..n)$ or
(2) $\exists r \in R^{\mathsf{P}}[p_1, \ldots, p_m]$ such that r is applicable and
 (2.1) $\forall q \in C(r) - \{p_1, \ldots, p_m\}, +\partial_{\mathsf{O}}{\sim}q \in P(1..n)$, and
 (2.2) $\forall p_i, 1 \leq i \leq m, \forall s \in R^{\mathsf{O}}[{\sim}p_i]$ either
  (2.2.1) s is discarded or
  (2.2.2) $\exists t \in R[p_i]$ such that t is applicable, $t > s$ and
   it $t \in R^{\mathsf{P}}[p_i]$, then $\forall q \in C(t) - \{p_1, \ldots, p_m\}, +\partial_{\mathsf{O}}{\sim}q \in P(1..n)$.

Consider a theory containing the rules $r_1\colon \Rightarrow_{\mathsf{P}} a \vee b \vee c$ and $r_2 :\Rightarrow_{\mathsf{O}} \neg a$ where $r_2 > r_1$. In this case, both rules are applicable. However, r_2 prevails over r_1 as far as the permission of a is concerned; but there are no rules against b nor c. Thus, we can conclude the strong permission of $b \vee c$. To derive an individual (strong) permission, we need that all other options are ruled out as forbidden. Hence, to obtain $\mathsf{P}_s b$ we need to have a rule such as $r_3\colon \Rightarrow_{\mathsf{O}} \neg c$ that is stronger than r_1.

For the third step (Definition 11), consider again the three rules given above. As we have just discussed we have $\mathsf{P}_s(a \vee b \vee c)$ given the existence of an explicit rule mandating such permission; we have $\mathsf{P}_s b$ from the explicit permissive rule and the prescriptive rules forbidding a and c, making b the only really permissive alternative. Is it reasonable to conclude $\mathsf{P}_s(a \vee b)$ or $\mathsf{P}_s(b \vee c)$? These are two permissive disjunctions whose content can be performed (albeit in the same way) without resulting in a violation, thus they appear to be genuine permissions.

Definition 11. The proof conditions to establish when a *minimal disjunctive permission is defeasibly provable* are defined as follows:
If $P(n+1) = +\partial^{min}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m$, then
(1) $\exists r \in R^{\mathsf{P}}[p_1, \ldots, p_m]$ such that r is applicable and
(2) $\exists p_j \in C(r), 1 \leq j \leq n$ such that $+\partial^{max}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_j \in P(1..n)$.

The final step is to put the previous three definitions together (Definition 12). A disjunction is provable as a permission if it satisfies one the three previous definitions. In case, one would disallow the last case, it is enough to remove Clause (3) from the definition below.

Definition 12. The proof condition to establish when a *disjunctive permission is defeasibly provable* is defined as follows:
If $P(n+1) = +\partial_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m$, then either
(1) $+\partial^{f}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m \in P(1..n)$ or
(2) $+\partial^{max}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m \in P(1..n)$ or
(3) $+\partial^{min}_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m \in P(1..n)$

For weak permissions we start with the condition to derive individual weak permissions. As we discussed before a weak permission corresponds to the lack of the obligation to the contrary. Hence, we can establish (1) the connection with the failure to derive the opposite obligation and (2)to be able to derive the corresponding strong permission. Then, we can use the condition for an individual permission to lift the condition to the case of a disjunctive weak permission, using the general modal logic inference pattern $\Box A \rightarrow \Box(A \vee B)$. Notice that differently to the case of $+\partial^{min}_{\mathsf{P}_s}$, the disjunction is not bound to an existing rule and can be used for an arbitrary disjunction. The limitation for strong permissions depends on the nature of permission that requires the explicit existence of a (derogating) permissive rule.

Definition 13. The proof conditions to establish when a *weak permission is defeasibly provable* and a *weak conjunctive permission is defeasibly provable* are defined as follows:
If $P(n+1) = +\partial_{\mathsf{P}_w} p$ then
(1) $-\partial_{\mathsf{O}} {\sim} p \in P(1..n)$ or
(2) $+\partial_{\mathsf{P}_s} p \in P(1..n)$.

If $P(n+1) = +\partial_{\mathsf{P}_w} p_1 \vee \cdots \vee p_m$ then
(1) $\exists p_i, 1 \leq i \leq m$ such that $+\partial_{\mathsf{P}_w} p_i \in P(1..n)$.

Finally, the case for a generic permission is just the simple combination of the corresponding conditions for strong and weak permissions and their disjunctions.

Definition 14. The proof conditions to establish when an *generic permission is defeasible provable* are defined as follows:
If $P(n+1) = +\partial_{\mathsf{P}} p$ then
(1) $+\partial_{\mathsf{P}_s} p \in P(1..n)$ or
(2) $+\partial_{\mathsf{P}_w} p \in P(1..n)$.

If $P(n+1) = +\partial_{\mathsf{P}} p_1 \vee \cdots \vee p_m$ then
(1) $+\partial_{\mathsf{P}_s} p_1 \vee \cdots \vee p_m \in P(1..n)$ or
(2) $+\partial_{\mathsf{P}_w} p_1 \vee \cdots \vee p_m \in P(1..n)$.

3. Properties of the Logic

In this section we are going to present a few results to demonstrate that the behaviour of the logic coincides with the properties identified by [7] for a guarded version of Free Choice Permission.

The first set of results concern some standard properties of Deontic Defeasible Logic, namely consistency and coherence.

Theorem 15. *For any Defeasible Theory D,*

- *for any proof tag # and any literal or disjunction l it is not possible to have both $D \vdash +\#l$ and $D \vdash -\#l$ (consistency).*
- *for any literal l it is not possible to have both $D \vdash +\partial_{\mathsf{O}} l$ and $D \vdash +\partial_{\mathsf{O}} \neg l$ (coherence)*

These results are immediate corollary of a result published in [6] that essentially specifies that any defeasible logic whose proof tags are defined using the principle of strong negation is consistent and coherent. Notice that the coherence result does not hold for permissions. Indeed it is possible to have permissive rules for opposite conclusions, and these are not in conflict with each other: $\mathsf{P}A$ and $\mathsf{P}\neg A$ are not contradictory in Standard Deontic Logic.

The next set of statements offers a description of the relationship among the different deontic modalities and basic properties of the guarded Free Choice Permission approach proposed by [7] and adopted in this work.

Theorem 16. *For any Defeasible Theory D, and for any literals l, l_1, l_2 and l_3:*

1. *it is not possible to have $D \vdash +\partial_{\mathsf{O}} l$ and $D \vdash +\partial_{\Box} {\sim} l$ (for $\Box \in \{\mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$).*
2. *if $D \vdash +\partial_{\mathsf{O}} l$, then $D \vdash +\partial_{\Box} l$ (for $\Box \in \{\mathsf{P}, \mathsf{P}_s, \mathsf{P}_w\}$).*
3. *if $D \vdash +\partial_{\mathsf{P}_s} l$, then $D \vdash +\partial_{\mathsf{P}_w} l$.*
4. *if $D \vdash +\partial^f_{\mathsf{P}_s} l_1 \vee l_2$ and $D \vdash +\partial_{\mathsf{O}} {\sim} l_1$, then $D \vdash +\partial_{\mathsf{P}_s} l_2$.*
5. *if $D \vdash +\partial^f_{\mathsf{P}_s} l_1 \vee l_2 \vee l_3$ and $D \vdash +\partial_{\mathsf{O}} {\sim} l_1$, then $D \vdash +\partial_{\mathsf{P}_s} l_2 \vee l_3$.*
6. *if $D \vdash +\partial^f_{\mathsf{P}_s} l_1 \vee l_2$, $D \vdash +\partial_{\mathsf{P}_w} l_1$ and $D \vdash +\partial_{\mathsf{P}_w} l_2$, then $D \vdash +\partial_{\mathsf{P}_s} l_1$ and $D \vdash +\partial_{\mathsf{P}_w} l_2$.*

Here we provide the (easy to verify) correspondence with the properties above and the axioms of [7].

1. $\mathsf{O}A \wedge \mathsf{P}\neg A \rightarrow \bot$
2. $\mathsf{O}A \rightarrow (\mathsf{P}_s A \wedge \mathsf{P}_w A)$
3. $\mathsf{P}_s A \rightarrow \mathsf{P}_w A$
4. $\mathsf{P}_s(A \vee B) \wedge \mathsf{O}\neg A \rightarrow \mathsf{P}_s B$
5. $\mathsf{P}_s(A \vee B \vee C) \wedge \mathsf{O}\neg A \rightarrow \mathsf{P}_s(B \vee C)$
6. $\mathsf{P}_s(A \vee B) \wedge \mathsf{P}_w A \wedge \mathsf{P}_w B \rightarrow \mathsf{P}_s A \wedge \mathsf{P}_s B$.

4. Conclusions

In this paper, we started with the idea that the guarded version of Free Choice Permission proposed in [7] has essentially a defeasible nature, and we used the idea to create a variant of Defeasible Deontic Logic that accounts for disjunctive permissions and accommodates the guarded Free Choice Permission in a constructive and computationally oriented way. We have shown that the resulting logic satisfies several properties advanced for a logic for the guarded Free Choice Permission. The work on computational complexity and efficient implementation of the logic is left for future work, but, given the structure of the proof conditions (and the similarity with other variants of Defeasible Logic), we expect the complexity to be computationally feasible.

In [4], we have studied how to extend the Defeasible Deontic Logic with conjunctive obligations. The two variants seem to be orthogonal and complement each other. In the current variant, we do not handle negated disjunctions in the scope of permissions. These can be handle by applying De Morgan, and then use the proof theory for conjunctive obligations and permissions. However, the details of such integration are still to be studied.

Another issue we plan to investigate in the future concerns disjunctive obligations. Specifically, we can ask under what conditions it is possible to derive an individual obligation from a disjunctive obligation. Namely, we will examine the so-called Deontic Disjunctive Syllogism

From $\mathsf{O}(A \vee B)$ to $\mathsf{O}B$ provided $\mathsf{O}\neg A$.

In Standard Deontic Logic, it can be represented by the following inference rule

$$\frac{\mathsf{O}(A \vee B) \qquad\qquad \mathsf{O}\neg A}{\mathsf{O}B} \tag{4.1}$$

and it is logically equivalent to axiom K of Standard Deontic Logic

$$\mathsf{O}(A \rightarrow B) \rightarrow (\mathsf{O}A \rightarrow B) \tag{4.2}$$

and to the Deontic Detachment inference rule

$$\frac{\mathsf{O}(A \to B) \qquad \mathsf{O}A}{\mathsf{O}B} \tag{4.3}$$

Deontic Detachment is often related to the so-called Contrary-to-duty paradoxes of Deontic Logic, and it has been debated whether such a principle should be accepted for reasoning with normative systems. While there is debate on Deontic Detachment, the following two examples suggest that the (logically equivalent) Deontic Disjunctive Syllogism is a natural and intuitive patters and as such should be a valid inference pattern.

Example 3. Horty [10, p. 430–431] who proposed the example of two norms “fight in the army or perform alternative service” and “don’t fight in the army” (possibly from two different sources), where he claims that the obligation to perform the alternative service follows, from an intuitive standpoint, from the two (partially) conflicting norms.

Example 4. [3] the rules of sudoku (9×9) prescribe that

1. for every cell, for every row, column or block the cell must contain one digit from 1 to 9; and
2. for every row, column or block, if a cell contains a digit, no other cell in the same row, column or block can contain that digit.

The first rule establishes that for each cell there are (nine) permissible alternatives, but if a digit already appears in the row or column or block where a cell appears, then it is forbidden to put the digit in the cell.

The work we presented offers some insights into how to handle conjunctive obligations. Nevertheless, we still have to investigate how to properly integrate and coordinate the interactions between disjunctive obligations and disjunctive permissions.

References

[1] G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic*, 2:255–287, 2001.

[2] G. Antoniou, D. Billington, G. Governatori, M.J. Maher, and A. Rock. A Family of Defeasible Reasoning Logics and its Implementation. In W. Horn, editor. *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, pages 459–463. IOS Press, 2000.

[3] G. Governatori. A Short Note on the Chisholm Paradox. In G. Casini, L. Di Caro, G. Governatori, V. Leone, and R. Markovich, editors. *Proceedings of the 4th International Workshop on MIning and REasoning with Legal texts*. CEUR-WS.org, 2020.

[4] G. Governatori, S. Colombo Tosatto, and A. Rotolo. A Defeasible Deontic Logic for Pragmatic Oddity. In F. Liu, A. Marra, P. Portner, and F. Van De Putte, editors. *Deontic Logic and Normative Systems: 15th International Conference (DEON2020/2021)*. College Publications, 2021.

[5] G. Governatori, F. Olivieri, A. Rotolo, and S. Scannapieco. Computing Strong and Weak Permissions in Defeasible Logic. *Journal of Philosophical Logic*, 42:799–829, 2013.

[6] G. Governatori, V. Padmanabhan, A. Rotolo, and A. Sattar. A Defeasible Logic for Modelling Policy-based Intentions and Motivational Attitudes. *Logic Journal of the IGPL*, 17:227–265, 2009.

[7] G. Governatori and A. Rotolo. Is Free Choice Permission Admissible in Classical Deontic Logic? In F. Liu, A. Marra, P. Portner, and F. Van De Putte, editors. *Deontic Logic and Normative Systems: 15th International Conference (DEON2020/2021)*. College Publications, 2021.

[8] G. Governatori and A. Rotolo. Logic of Violations: A Gentzen System for Reasoning with Contrary-To-Duty Obligations. *Australasian Journal of Logic*, 4:193–215, 2006.

[9] S.O. Hansson. The Varieties of Permissions. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.

[10] J. Horty. Deontic Modals: Why Abandon the Classical Semantics. *Pacific Philosophical Quarterly*, 95:424–460, 2014.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200854

A Genetic Approach to the Ethical Knob

Giovanni IACCA [c], Francesca LAGIOIA [a,b], Andrea LOREGGIA [b], and Giovanni SARTOR [a,b,1]

[a] *CIRSFID - Alma AI, University of Bologna, Italy*
[b] *European University Institute, Florence, Italy*
[c] *University of Trento*

AbstractAs Autonomous vehicles (AVs) are entering shared roads, the challenge of designing and implementing a completely autonomous vehicle is still open. Aside from technological issues regarding how to manage the complexity of the environment, AVs raise difficult legal issues and ethical dilemmas, especially in unavoidable accident scenarios. In this context, a vast speculation depicting moral dilemmas has developed in recent years. A new perspective was proposed: an "Ethical Knob" (EK), enabling passengers to ethically customise their AVs, namely, to choose between different settings corresponding to different moral approaches or principles. In this contribution we explore how an AV can automatically learn to determine the value of its "Ethical Knob" in order to achieve a trade-off between the ethical preferences of passengers and social values, learning from experienced instances of collision. To this end, we propose a novel approach based on a genetic algorithm to optimize a population of neural networks. We report a detailed description of simulation experiments as well as possible applications.

Keywords. Autonomous vehicles, Ethical Knob, Genetic Algorithm, Ethical Dilemmas

1. Introduction

Determining how self-driving cars should tackle moral decisions is a major challenge for designers, deployers and regulators. Scholars, policy makers, general media, blog posts and even dedicated websites discuss how AVs should behave in hypothetical accident scenarios, where they have to make decisions involving harms to humans [1,15,12]. Consider for instance the following scenario: in a dangerous and unavoidable accident situation, an AV must decide between staying on course and hitting several pedestrians or swerving, thus killing one passer-by. Should the AV sacrifice one person to save the lives of many? Imagine next that the choice of swerving will cause the passengers' death. Should the AV let its passengers die rather than driving into several pedestrians? Many academic articles [1,11,12] have discussed similar scenarios, on the basis of the classical Trolley Problem, i.e. the ethical thought experiment discussed by Foot [3] and Thomson [17].

[1] F. Lagioia, A. Loreggia and G. Sartor have been supported by the H2020 ERC Project "CompuLaw" (G.A. 833647).

In this context, scholars refer to different ethical theories, such as utilitarian [1], deontological, e.g., Kantian [6], virtue ethics [9] or contract theory [5,10] approaches, and investigate how to program AVs based on such theories [2,4,9].

A further question is whether all AVs should have the same mandatory ethics setting (MES) [5,11] or every user/owner should have the choice to select his or her own personal ethics setting (PES) [1,7]. It has indeed been claimed that an AV should have different ethics settings consistent with several ethical theories, allowing each individual passenger/owner to decide what moral approach her AV should have [16]. Thus, an AV would be considered and function as a “moral proxy” for drivers/owners ethical outlook, rather than a distinct “moral agent” [14]. It has also been argued that AVs could be equipped with an “ethical knob”, enabling passengers to determine the degree to which the AV prioritizes their lives over the lives of third parties [2]. The provision of personal ethics settings reflects the value of autonomy and is sensitive to the moral views of the members of society. A recent web poll by robohub.org, concerning who should determine how an AV responds in ethical dilemma situations, supports this result. Most of the participants (44%) thought that the passengers should decide how an AV responds in ethical dilemma situations, while 33% thought that lawmakers should have the final say [13]. Despite the potential advantages, the idea of a PES has also attracted some criticisms, since people might then potentially choose their moral settings based on racist ideologies or other types of wholly unacceptable outlooks [11]. In this regard, we may question whether there might be a middle ground between a completely open choice of ethics settings and the view that everyone should have the same MES. Allowing for a PES does not mean that all conceivable trade-offs should be allowed, since certain morally troubling options could be ruled out.

In this paper we examine the possibility of providing AVs with the ability to learn how to set their ethical knob in such a way as to reconcile the individual preferences of their passengers and social values (as implemented through legal sanctions and social norms).

2. Ethical Knob, Individual Preferences and Social Values

In [2], it was assumed that the owner(s)/passenger(s) would set the ethical knob in their car by choosing a value from a continuous range between 0, denoting an extreme egoistic attitude (only passengers’ lives are valued), and 1, denoting an extreme altruistic attitude (only pedestrians’ lives are valued). Thus the knob was meant to express directly the ethical attitude of the AV passengers, i.e., the value they attribute to their life relative to the value of the lives of third parties (pedestrian potentially involved in road collisions). The AV would make the most advantageous choice, according to the set knob value, the number of lives at stake, as well as the probability that both passengers and third parties suffer harm, as a consequence of the driving decision.

In this work, we assume that the position of the knob no longer indicates the passengers’ moral attitude, but rather the AV’s assessment of the relative importance of the lives of passenger(s) and third parties. This assessment is the outcome of a learning process based on the the AV’s engagement in accidents, and on its evaluation of the outcomes of such accidents. This evaluation takes into account the passengers’ moral attitudes (their intrinsic preferences), as well as legal sanctions and social norms (extrinsic incentives).

In particular, the knob position is the outcome of an agent-based simulation, built on a genetic algorithm. Each step of the simulation takes into account 100 accidents simultaneously, each involving a particular AV. The simulation is run for 500 iterations. This process artificially "evolves" the knob values, according to the AVs utility function (which is parameterised to the moral attitude of the passenger(s) as well as to legal and social sanctions and rewards).

3. Methodology

Genetic algorithms [8] mimic the evolution process of a population that initially is made up of random individuals. Each individual is represented by a chromosome which is a possible solution to the problem being addressed. Individuals are evaluated based on a fitness function, indicating how well they perform. Better-performing individuals are given a higher chance of reproducing. In such a way, chromosomes that represent better solutions tend to spread in the population, while those representing inferior solutions tend to disappear. Small mutations, i.e., perturbation of some genes of chromosomes, may occur based on a random choice. This prevents the convergence toward a local minimum. In this work we implement the standard genetic algorithm schema described by the pseudo-code below 1.

Algorithm 1 Evolutionary algorithm of the Ethical Knob

```
procedure EK($n$)                    ▷ Input: $n$ number of individuals in the population
    Initialize a random population $P$ of $n$ individuals
    for Every generation do
        $EvaluateFitness(P)$
        $parents = SelectParents(P)$
        $offsprings = crossOver(parents)$
        $P = mutation(offsprings)$
    end for
    return $P$
end procedure
```

In our simulation, the genetic algorithm maximizes the payoff of individuals involved in the population P. Each AV is an individual $p_i \in P$, constituted by a neural network. In each iteration, the individual AV is located in a scenario where it faces a dilemma and decides what action to take (i.e., it can either go straight and put at risk pedestrian(s) or swerve and put at risk passenger(s)). Each scenario is defined by the following variables:

- $nPed_{p_i}$: number of pedestrians. It is a random number in $[0, maxPed]$; in the experiments we set $maxPed = 6$.
- $nPass_{p_i}$: number of passengers. It is a random number in $[0, maxPass]$; in the experiments we set $maxPass = 6$.
- a_{p_i}: intrinsic level of altruism for passengers in p_i. It is a random number in $[0, 1]$.
- s_{p_i}: intrinsic level of selfishness for passengers in p_i. It equal to $1 - a_{p_i}$, namely, it is the complement of the level of altruism.

- $prodPed_{p_i}$: probability of injuring pedestrians when the AV goes straight. It is a random number in $[0,1]$.
- $prodPass_{p_i}$: probability of injuring passengers when the AV swerves. It is a random number in $[0,1]$.

In each scenario and before taking an action, a neural network evaluates the aforementioned features and predicts the value of the knob to decide the action to take. For the purpose of the genetic algorithm, the chromosome of each individual corresponds to parameters of the neural network.

Initialization and fitness evaluation. Initially, a population P of $n = 100$ individual AVs is created at random. At the beginning of each iteration, each individual $p_i \in P$ is located in a scenario instantiated at random. The AV chooses its action on the basis of its knob level and the confronted scenario, the choice being represented by the value of the variable act_{p_i}: if $act_{p_i} = 0$ the AV goes straight, if $act_{p_i} = 1$ it turns. The value of act_{p_i} is computed as follows:

$$act_{p_i} = \begin{cases} 0 & \text{if } nPed_{p_i} \cdot probPed_{p_i} \cdot (1 - knob_{p_i}) \leq nPass_{p_i} \cdot probPass_{p_i} \cdot knob_{p_i} \\ 1 & \text{otherwise} \end{cases}$$

The formula indicates that the AV goes straight (rather than turning) based on the comparison of two quantities. The first if obtained by multiplying the number of pedestrians, the probability of harming them by going straight, and their relative importance according to the knob position. The second is similarly obtained by multiplying the number of passengers, the probability of harming them by turning, and their importance. If the first quantity is lower than the second, the AV goes straight, while if it is higher it turns.

After the action has been chosen, the response by the environment is given by the variable $dead_{p_i}$, which indicates whether the action of going straight has resulted in pedestrians' injuries or whether the action of turning has resulted in passengers' injuries. The variable $dead_{p_i}$ may be randomly instantiated to 0 (safe) or 1 (harmed), based on the probabilities $probPed_{p_i}$ or $probPass_{p_i}$.

Then the fitness of each $p_i \in P$ is evaluated using the following function:

$$f(p_i) = \Delta u(p_i) + reward(p_i)$$

The fitness has two components. The first is the delta-payoff $\Delta u(p_i)$, which is the difference between the utility of the choice made and the expected utility of the alternative choice. Both utilities include an evaluation of the outcome (injuries to passenger(s) or pedestrian(s)) according to the intrinsic moral attitude of the passengers/owners, as well as to the legal sanction for unjustified harms. The second component, i.e., $reward(p_i)$, expresses the social evaluation of the AV's behaviour. It is positive when the AV's behaviour is better than the average of the population, while it is negative when it is worse than the average. The $reward(p_i)$ may be understood as the impact of such evaluation on an individual's self/social esteem. For each $p_i \in P$, $\Delta u(p_i) = u(p_i) - u_{alt}(p_i)$, where $u(p_i)$ represents the payoff gained in the scenario after the action taken and $u_{alt}(p_i)$ is the payoff that would be obtained through the alternative action. $u(p_i)$ is computed as follow:

$$u(p_i) = \begin{cases} nPass_{p_i} \cdot s_{p_i} + (1 - dead_{p_i}) \cdot nPed_{p_i} \cdot a_{p_i} - dead_{p_i} \cdot nPed_{p_i} \cdot cPed & act_{p_i} = 0 \\ (1 - dead_{p_i}) \cdot nPass_{p_i} \cdot s_{p_i} + nPed_{p_i} \cdot a_{p_i} & act_{p_i} = 1 \end{cases}$$

where the first line yields the utility (for preserving people lives/health) obtained by going straight, and the second line yields the utility obtained by turning.

In the first line:

- $nPass_{p_i} \cdot s_{p_i}$ is the selfish utility obtained by preserving passengers (who are all preserved when the car goes straight)
- $(1 - dead_{p_i}) \cdot nPed_{p_i} \cdot a_{p_i}$ is the altruistic utility obtained by preserving pedestrians (in case they are not injured, i.e. $dead_{p_i} = 0$ even through the AV's choice puts them at risks)
- $dead_{p_i} \cdot nPed_{p_i} \cdot cPed$ it is the total legal sanction (compensation) due for causing the death of a pedestrian, where *cPed* is the sanction for injuring a single pedestrians. The sanction is applied when the AV has behaved negligently, in the sense of choosing to harm pedestrians in a situation in which the expected harm to pedestrian exceeds the expected benefit to passengers. The value of *cPed* is 1 if $probPed_{p_i} \cdot nPed_{p_i} > probPass_{p_i} \cdot nPass_{p_i}$, otherwise it is 0.

In the second line:

- $(1 - dead_{p_i}) \cdot nPass_{p_i} \cdot s_{p_i}$ is the selfish benefit obtained when passengers survive (even if they were put at risk)
- $nPed_{p_i} \cdot a_{p_i}$ is the altruistic utility obtained by preserving pedestrians

As noted above, the AV assesses the action it has performed by comparing the utility obtained through that action and the expected utility that would have been obtained by taking the alternative. The latter utility is computed according to the following formula:

$$u_{alt}(p_i) = \begin{cases} nPass_{p_i} \cdot s_{p_i} \cdot (1 - probPass_{p_i}) + nPed_{p_i} \cdot a_{p_i} & act_{p_i} = 0 \\ nPass_{p_i} \cdot s_{p_i} + nPed_{p_i} \cdot a_{p_i} \cdot (1 - probPed_{p_i}) + & \\ -nPed_{p_i} \cdot cPed \cdot probPed_{p_i} & act_{p_i} = 1 \end{cases}$$

In order to compute the social reward function, we need to know how individuals in the community behave on average. To do that, we compute the average knob of the community as $knob_P = \frac{1}{|P|} \sum_{p_j \in P} knob_{p_j}$. The average know is used to compute the action of an average AV in each scenario p_i. The action is computed replacing the value of $knob_{p_i}$ with the value of $knob_P$ in the formula of act_{p_i}.

We then check whether the action taken by the AV differs from the action that would be taken by the average individual. If the average individual would go straight and the AV turns, then the action is rewarded (having done an action that is meritorious, since it minimizes the risk of losses more than the average). On the other hand, if the average individual would turn and the AV goes straight, then it is blamed.

$$reward(p_i) = \begin{cases} 0.25 & \text{if } act_{(P,p_i)} = 0 \text{ and } act_{p_i} = 1 \\ -0.25 & \text{if } act_{(P,p_i)} = 1 \text{ and } act_{p_i} = 0 \end{cases}$$

In this simulation, we have assumed that the level of altruism is randomly chosen.

Parents selection. After the evaluation step, a subset of individuals are selected as a basis to compute the next generation. For the selection process, we used a tournament selection algorithm: individuals are paired at random and those with the highest fitness in the couples are selected as parents. It should be noted that the same individual may appear more than once in the set of parents. In our experiments, the set of parents is set at 80% of the original population.

Crossover. The idea at the basis of the crossover operator is to mimic the combination of genes that takes part in reproduction. The chromosomes of one individual are combined with those of another individual. In such a way, the solution space is explored starting from a random point, and at each iteration the algorithm moves the solution towards a better solution. In this work, chromosomes are represented by the weights of neural networks. The crossover operator creates a new chromosome by choosing at random one weight from one parent or the other. Parents are paired at random to generate a new individual, until a new population of n individuals is generated.

Mutation. The mutation operator is applied to each child's chromosomes. It is used to prevent premature convergence, i.e., to avoid getting stuck at local optima. The operator acts on the chromosome by altering certain genes with some probability. In this work, if a gene is chosen for mutation, its value is randomly varied in a range of 1% of the original value.

4. Experimental Evaluation

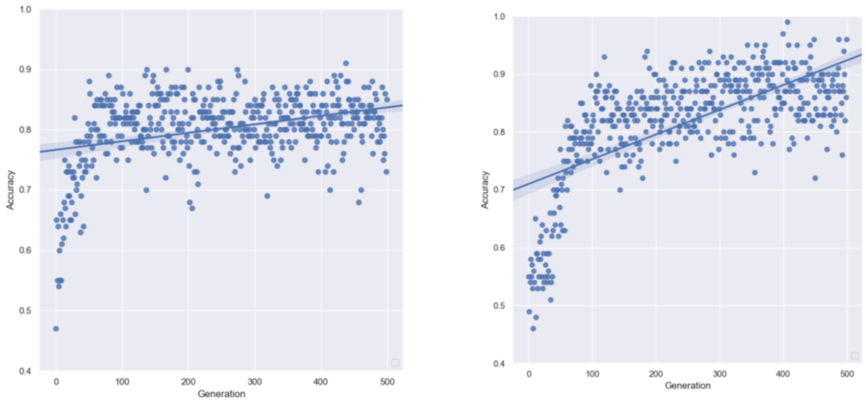
To evaluate the genetic algorithm described in the previous section and reported in the pseudo-code of Algorithm 1, we developed a Python 3.6 framework that implements it. Neural networks are defined using Keras ver. 2.2.4 over Theano ver. 1.0.3. Each individual $p_i \in P$ represents an AV, having the following features:

- Altruism level (i.e., a_{p_i}): a number in the range $[0,1]$, which describes how much an individual cares about the others relative to itself;
- Fitness (i.e., $f(p_i)$): a value which describes the goodness of the individual with respect to its behaviour in the population;
- Knob Level (i.e., $knob_{p_i}$): a number in the range $[0,1]$, representing the Ethical Knob described in [2]; the device determines the behaviour of the AV in ethical dilemma situations;
- Neural Network: it computes a regression task whose objective function is to optimise the level of the knob. In our empirical evaluation, the neural network has the following characteristics: 3 layers, one input layer with 5 nodes, one hidden layer with 3 nodes and one output layer with one node. The ReLu is used as the activation function for the hidden layer, while tanh is the activation function for the output layer.

In order to evaluate the performances of the framework and analyze whether the genetic approach is able to optimize the neural network, we performed four different experiments. Such experiments aim to test different approaches, as described below.

Experiment 1: $reward(p_i) = 0$ and $cPed = 0$. The aim is to test a simple situation in which the fitness function does not take into account any penalties from legal norms or any reward/stigma deriving from social norms.

Figure 1. Accuracy for different settings: each blue dot represents the number of individuals in the population who take the action that maximizes the fitness function.



(a) Experiment 2

(b) Experiment 3

Table 1. Accuracy and confusion matrix (standard deviation in brackets) for the different settings. In each scenario all the features are drawn randomly.

Setting	Accuracy	TP	TN	FP	FN
Experiment 1	0.8487 (0.01)	0.5390 (0.00)	0.3097 (0.01)	0.1403 (0.01)	0.0110 (0.00)
Experiment 2	0.8442 (0.00)	0.5600 (0.00)	0.2842 (0.00)	0.0358 (0.00)	0.1200 (0.00)
Experiment 3	0.9467 (0.01)	0.5500 (0.00)	0.3967 (0.01)	0.0533 (0.01)	0.0000 (0.00)
Experiment 4	0.8357 (0.01)	0.6800 (0.00)	0.1557 (0.01)	0.1643 (0.01)	0.0000 (0.00)

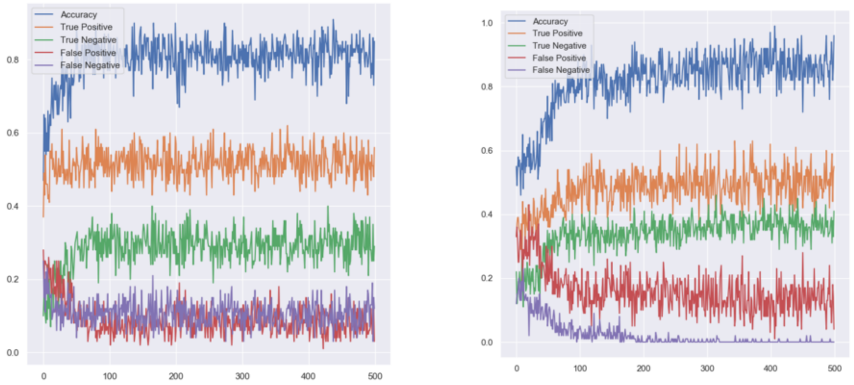
Experiment 2: $reward(p_i) = 0$ and $cPed = 1$. The aim is to check whether legal norms may influence the system's performance.

Experiment 3: the reward is in $\{-0.25, 0.25\}$ and $cPed = 0$. The aim is to explore whether social norms may influence the system's performance.

Experiment 4: the reward is in $\{-0.25, 0.25\}$ and $cPed = 1$. The aim is to check whether and to what extent the combination of legal and social norms may influence the system's performance.

The prediction task can be seen as a binary classification task in which the AV learns to take the action which maximizes the payoff. In particular, looking at the fitness function, we classify samples as: Real Positive, when the preferable action is to turn; Real Negative, when the preferable action is to go straight; Predicted Positive, when the neural network predicts a knob level which makes the AV turn; Predicted Negative, when the neural network predicts a knob level which makes the AV go straight.

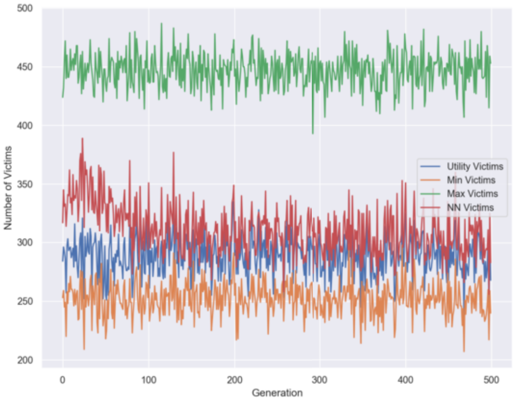
Figure 2. Accuracy and confusion matrix for different settings: blue line reports accuracy, orange line reports true positive, green line reports true negative, red line reports false positive and purple line reports false negative.



(a) Experiment 2

(b) Experiment 3

Figure 3. Number of victims over 500 generations in a deterministic environment. Different lines represent different approaches. In this setting, the utility function line coincides with the min function.



4.1. Data analysis

Based on the previous definitions, for each experiment we plot 3 different metrics, describing how individuals in the population evolve, generation after generation. Specifically for each generation we plot: Accuracy, which describes how many predictions coincide with the preferable actions; Confusion Matrix, which shows true positives, true negatives, false positives and false negatives; Number of victims, which describes the number of casualties that may be caused by an AV, using the knob values proposed by neural networks. In particular, the last metric is compared with number of victims caused by 3 different AVs: one which always minimizes the number of victims, one which always chooses the optimal action and one which always maximizes the number of victims. Figure 1 shows the accuracy for Experiments 2 and 3, in both of which, the accuracy increases. This suggests that neural networks in the population improve generation after generation. Notice that the increase of accuracy in Experiment 3 is steeper, suggesting that the opinion of the community (i.e., the stigma or the honour given based on average behaviour in the community) has a higher influence on the evolutionary process. When no reward is used–as in Experiment 1 and 2– or when it is applied in combination with a cost for harming pedestrians-like in Experiment 4–increment of the performance is less evident.

In order to understand whether the different components of the fitness function influence the final payoff of an individual, we performed a post-hoc analysis. Figure 2 shows the values of confusion matrix for Experiment 2 and 3 during the evolutionary process: the introduction of a cost decreases the number of false positives (see Figure 2a). On the other hand, the reward seems to reduce the number of both false positives and false negatives (see Figure 2b).

Moreover, at the end of the simulation when the networks are optimized, we generate 100 new scenarios. We use them as input for all the neural networks in the population and then count how many scenarios per individuals were tackled correctly. Table 1 shows the average values of accuracy and standard deviation when the probability of death for both the passenger and pedestrians is set to 1. Even though the accuracy is high, for some scenarios a high number of false positives can be noted. We conjecture this is due to the introduction of the reward. Indeed, whenever the AV is in doubt (usually because the number of pedestrians is close to that of passengers) the neural network predicts a false positive rather than a false negative, since the former can be rewarded whenever the community considers the AV choice an heroic action. The number of victims in the scenario is a metric which describes the system's ability to mediate between individuals' preferences and casualties minimization. Figure 3 shows the number of victims caused on 4 different approaches. The green line represents the maximal number of victims for each generation, while the orange represents the minimal number of victims for each generation. It is interesting to notice the following: firstly, when the AV operates in a deterministic scenario (i.e., the probabilities of harming pedestrians and individuals are set to 1), the utility function works as a proxy for the min function. Secondly, the number connected with the neural network prediction (i.e., the red line) decreases very quickly after a few generations. This is a signal that the optimization process is working towards the desired direction.

We have also run experiments where the ethical attitude of individuals (car owners/passengers) was given, rather than being randomly assigned. Not surprisingly in this case reducing the number of deaths required higher legal sanctions or social rewards.

5. Conclusion and future research

We have presented a model where AVs learn how to set their knob, i.e., what importance to give to the safety of passengers relative to the safety of pedestrians. This is obtained by having the AVs make choices and learn from the value of the outcome of such choices. The assessment of the value of the AV's choices is dependant on considering the passengers' moral attitude (their intrinsic preferences) as well as legal sanctions and social norms (extrinsic incentives). In particular, the merit of a choice has been determined by comparing the outcome of the choice and the expected outcome that would be obtained by making a different choice. An alternative model, which only takes into account the absolute outcome of a choice (the number of individuals not harmed minus those that were harmed, plus the applicable sanction and reward), has been considered. The learning takes place thanks to an evolutionary algorithm that differentially replicates the AVs making most successful choices. The results obtained show how convergence of socially valuable behaviour can be obtained by providing appropriate mechanisms for sanction and reward. In the future we aim to expand our model. For instance, we intend to endow our agents with memory — enabling them to learn probability distributions by considering their past outcomes and those of observable others—, and to model how individual ethical approaches are influenced by societal preferences. We plan to insert out agents in existing traffic simulators (such as SUMO) to test our model in a dynamic environment. This will enable us to address more complex and realistic traffic situations, involving multiple choices under resource constraints. Finally, we will investigate possible regulations of the setting of knobs, particularly in regard to liability issues (see [2]).

References

[1] J.-F. Bonnefon, A. Shariff, and I. Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576, 2016.

[2] G. Contissa, F. Lagioia, and G. Sartor. The Ethical Knob: ethically-customisable automated vehicles and the law. *Artificial Intelligence and Law*, 25(3):365–378, 2017.

[3] P. Foot. The problem of abortion and the doctrine of double effect. *Oxford Review*, (5), 1967.

[4] J. C. Gerdes and S. M. Thornton. Implementable ethics for autonomous vehicles. In *Autonomes fahren*, pages 87–102. Springer, 2015.

[5] J. Gogoll and J. F. Müller. Autonomous cars: In favor of a mandatory ethics setting. *Science and Engineering Ethics*, Jul 2016.

[6] J. K. Gurney. Crashing into the unknown: An examination of crash-optimization algorithms through the two lanes of ethics and law. *Alb. L. Rev.*, 79:183, 2015.

[7] J. Himmelreich. Never mind the trolley: The ethics of autonomous vehicles in mundane situations. *Ethical Theory and Moral Practice*, 21(3):669–684, 2018.

[8] J. Holland. Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control and artificial intelligence*, 1975.

[9] W. Kumfer and R. Burgess. Investigation into the role of rational ethics in crashes of automated vehicles. *Transportation research record*, 2489(1):130–136, 2015.

[10] D. Leben. A rawlsian algorithm for autonomous vehicles. *Ethics and Information Technology*, 19(2):107–115, 2017.

[11] P. Lin. Here's a terrible idea: Robot cars with adjustable ethics settings. *Wired*, 2014.

[12] P. Lin. *Why Ethics Matters for Autonomous Cars*, pages 69–85. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[13] J. Millar. You should have a say in your robot car's code of ethics. *WIRED. com*, 2014.

[14] J. Millar. Technology as moral proxy: Autonomy and paternalism by design. *IEEE Technology and Society Magazine*, 34(2):47–55, 2015.

[15] S. Nyholm and J. Smids. The ethics of accident-algorithms for self-driving cars: an applied trolley problem? *Ethical Theory and Moral Practice*, pages 1–15, 2016.

[16] A. Sandberg and H. Bradshaw-Martin. What do cars think of trolley problems: ethics for autonomous cars. *Beyond AI: Artificial Golem Intelligence*, page 12, 2013.

[17] J. J. Thomson. Killing, letting die, and the trolley problem. *The Monist*, 59(2):204–217, 1976.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200855

Topic Modelling Brazilian Supreme Court Lawsuits

Pedro Henrique LUZ DE ARAUJO [a,1] and Teófilo DE CAMPOS [a,2]
[a] *Department of Computer Science, University of Brasília, Brasília, DF, Brazil*

Abstract. The present work proposes the use of Latent Dirichlet Allocation to model Extraordinary Appeals received by Brazil's Supreme Court. The data consist of a *corpus* of 45,532 lawsuits manually annotated by the Court's experts with theme labels, a multi-class and multi-label classification task. We initially train models with 10 and 30 topics and analyze their semantics by examining each topic's most relevant words and their most representative texts, aiming to evaluate model interpretability and quality. We also train models with 30, 100, 300 and 1,000 topics, and quantitatively evaluate their potential using the topics to generate feature vectors for each appeal. These vectors are then used to train a lawsuit theme classifier. We compare traditional bag-of-words approaches (word counts and tf-idf values) with the topic-based text representation to assess topic relevancy. Our topics semantic analysis demonstrate that our models with 10 and 30 topics were capable of capturing some of the legal matters discussed by the Court. In addition, our experiments show that the model with 300 topics was the best text vectoriser and that the interpretable, low dimensional representations it generates achieve good classification results.

Keywords. topic models, legal domain, document analysis, Latent Dirichlet Allocation

1. Introduction

Brazil's court system suffers from an excessive amount of lawsuits [1]. About 80 million suits awaited judgement in 2017, which amounts to almost one for every three Brazilians. There was an increase of 19.4 million suits between 2009 and 2017. Furthermore, the average processing time reaches more than seven years in some cases. Such long waiting times negatively impact Brazil's legal certainty and brings about greater budgetary needs—Brazil spent R$ 90.7 billions in 2017 to maintain the judiciary, corresponding to about 28 billion[3] dollars [2].

Natural Language Processing (NLP) and Machine Leaning techniques can contribute to a quicker, cheaper and more efficient analysis of legal proceedings and as a result help promote greater effectiveness and democratization of justice. Some works already explore the use of artificial intelligence in the context of Brazil's courts [3,4,5].

[1] Corresponding Author: Pedro Henrique Luz de Araujo, UnB - Brasília, DF, Brazil; E-mail: pedro.luz@aluno.unb.br.

[2] Corresponding Author: Teófio Emidio de Campos, UnB - Brasília, DF, Brazil; E-mail: t.decampos@oxfordalumni.org.

[3] Considering average exchange rate of 2017: 3.19 reais to 1 dollar.

That being said, we are not aware of publications regarding the topic modelling of Brazilian lawsuits.

Topic models are a family of statistical models used to discover in an automatic and unsupervised manner themes (topics) present in a collection of documents [6]. The topics are obtained from the statistical analysis of the words that comprise the documents. Since annotations and labelling of documents are not needed, topic models enable the organisation, exploration and indexing of massive amounts of data in a scale that could be prohibitively expensive if human made. The trained models may also be used for downstream tasks such as sentiment analysis [7] and document classification [8]. In addition, the approach is not restricted to text data and may be used to model genomic data, images and social networks [6].

In this paper, we employ Latent Dirichlet Analysis (LDA) to model Extraordinary Appeals (*Recursos Extraordinários*—RE) received by Brazil's Supreme Court (*Supremo Tribunal Federal*—STF). Each suit has been manually annotated by the Court's employees to include information on its general repercussion (*repercussão geral*) themes. This is a multi-label classification task, which we will further discuss in Section 3. Our contributions are:

1. The qualitative analysis of the semantics of each topic from models with 10 and 30 topics trained on the STF data.
2. The quantitative analysis of topic relevance by using topic distribution vectors as input for general repercussion theme classification. We experiment with models of 10, 30, 100, 300 and 1,000 topics.

The rest of the paper is organized as follows. Section 2 briefly review Topic Model literature and NLP applied to the legal domain approaches. Sections 3 and 4 describe the dataset and the model employed, respectively. Section 5 reports our experiments and Section 6 presents and discusses the results. Section 7 concludes the paper.

2. Related Work

2.1. Topic Models

Topic models have been an area of research since 1990, when Deerwester et al. [9] proposed Latent Semantic Indexing (LSI). The method uses Singular Value Decomposition (SVD) to factorize a matrix of term-document co-occurrence values to construct a "semantic" space where terms and documents closely associated are near one another. The method is further explored by Hofmann [10], who introduced probabilistic LSI (PLSI). Like LSI, PLSI decomposes a co-occurrence matrix, but while the former uses a linear algebra approach, the latter method is statistical, modelling the document-word co-occurrence probability as a mixture of conditionally independent multinomial distributions. On the other hand, PLSI has some weaknesses, such as the linear growth of the parameters with the size of the corpus, which causes overfitting issues, and the lack of procedure to assign probability to a document not seen in the training set.

To overcome PLSI weaknesses, Blei et al. [11] proposed Latent Dirichlet Allocation (LDA). The authors show that LDA can be used for a range of tasks, such as document modelling, text classification and collaborative filtering, outperforming approaches based on unigrams and PLSI.

Since then, the study of extensions of LDA by relaxing some of its assumptions has been an active area of research [6]. For example, by relaxing the assumption that the order of the documents can be neglected, Blei and Lafferty [12] propose Dynamic Topic Models, capable of modelling the time evolution of topics in a corpus.

2.2. *Natural Language Processing and Topic Models in Legal Text*

Efforts have been made to apply Natural Language Processing and Machine Learning techniques to legal text. NLP has been used to automatically extract and classify relevant entities in court documents [13,14,4]. Other works [15,16,17,18] focus on using automatic summarization to reduce the amount of information legal professionals have to process. Document classification has been explored for decision prediction [19,20], area of legal practice attribution [21] and fine-grained legal-issue classification [22].

LDA has been employed to model legal corpora. Carter et al. [23] model documents from the Australian High Court; Remmits [24] models decisions from the Supreme Court of the Netherlands; O'Neill et al. [25] used LDA to explore British legislative texts.

Some works explore the processing of Brazilian legal documents. Correia da Silva et al. [3] use a CNN to classify STF's documents. De Vargas Feijó and Moreira [5] introduce a dataset for decision summarization. Luz de Araujo et al. [4] built a manually annotated corpus for named entity recognition and classification with legislation and legal decision classes. On the other hand, we are not aware of publications examining topic modelling of Brazilian legal corpora.

3. Data

We use the VICTOR dataset [26], a corpus containing 45,532 Extraordinary Appeals. Each instance is a legal proceeding as it is received by the STF, that is, before it is processed and judged. Each lawsuit is represented as an ordered sequence of pages containing text.

The dataset contains manual annotation that assigns to each lawsuit one or more general repercussion[4] themes. More specifically, the options are the 28 most important themes according to the STF, each one identified by a unique integer[5]; e.g., theme 6 deals with the State's duty to supply costly medications to citizens who suffer from serious diseases and are not able to buy them. The integer 0 identifies the instances that contain at least one theme that does not belong to any of those 28 classes. It follows that theme assignment is a multi-label classification task.

The data is divided into train/validation/test splits containing 70%/15%/15% of all suits, respectively. The theme distribution is the same in all splits as figure 1 shows.

The following preprocessing steps were applied to the raw text: lower-casing, removal of stop words and alphanumeric tokens, email and URL tokenization, and identification of simple law citations; e.g., we change *Lei* (law) 11.419 to LEI_11419.

[4] An appeal must have general repercussion to be judged by the STF. This means that lawsuit must relate to relevant economic, political, social or legal issues that exceed the interests of the parties.

[5] A list of all themes is available at `http://www.stf.jus.br/portal/jurisprudenciaRepercussao/abrirTemasComRG.asp.`

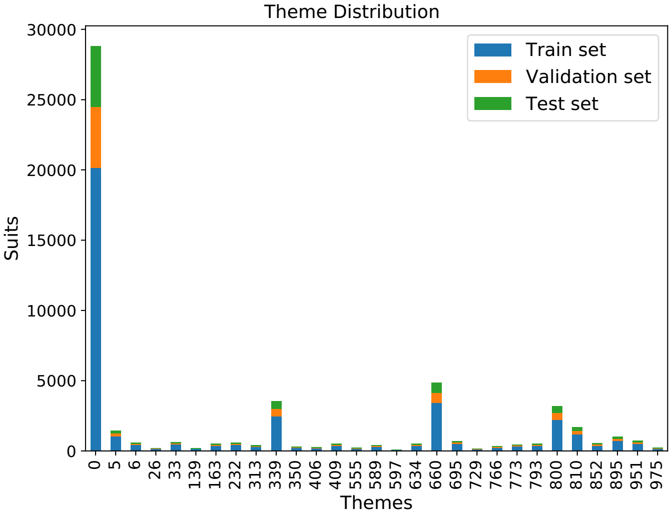


Figure 1. Theme counts.

4. Model

Inspired by previous attempts to model different kinds of legal text [23,24,25], we choose Latent Dirichlet Allocation [11] as the method for topic generation. LDA is a probabilistic generative model of a corpus, where each document is represented as a random mixture over latent topics. Each topic is in turn a distribution over words. That is, LDA assumes the following generative process for a corpus D of m documents of length n_i, $i \in [1,\ldots,m]$, assuming a fixed set of k topics:

1. $\boldsymbol{\theta}_i$, $i \in \{1,\ldots,m\}$, the topic distribution of document i, is chosen from a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha})$
2. $\boldsymbol{\phi}_j$, $j \in \{1,\ldots,k\}$, the word distribution of topic j, is chosen from a Dirichlet distribution $\text{Dir}(\boldsymbol{\beta})$.
3. For each word position (i,j), $i \in \{1,\ldots,m\}$, $j \in \{1,\ldots,n_i\}$:
   (a) A topic $\mathbf{z}_{i,j} \sim \text{Multinomial}(\boldsymbol{\theta}_i)$ is chosen.
   (b) A word $\mathbf{w}_{i,j} \sim \text{Multinomial}(\boldsymbol{\phi}_{z_{i,j}})$ is chosen.

Given this generative assumption, the LDA procedure assigns: a topic distribution for each document, a topic for each word in each document and a word distribution for each topic.

5. Experiments

5.1. Model Training for Exploratory Analysis

We perform an exploratory analysis of the data aiming to understand its most relevant topics by training LDA models. We train two models on the training split of the data, one with 10 topics and the other with 30. Since the whole data does not fit into memory, we use the algorithm proposed by [27] for the online training of LDA models, based on stochastic optimisation with gradient steps.

To select the most informative words, we restrict our vocabulary to the words that appear in at least 50 lawsuits of the training set and in no more than 50% of them. In addition, we filter words with only one letter, with the intuition that they probably do not help with topic interpretability. The obtained vocabulary contains 81,418 entries.

We use mini-batches of 4,096 suits, with a maximum number of 400 iterations per mini-batch, and train for 4 epochs. The hyper-parameters were chosen empirically and were sufficient for the convergence of most lawsuits in the training set.

5.2. Topic Distribution as Text Representation

In order to have a quantitative analysis of the detected topics, we use LDA as a lawsuit feature extractor; that is, the topic distribution of each lawsuit is used as its vector representation and fed to a classifier to predict general repercussion themes. We run experiments with models of 10, 30, 100, 300 and 1,000 topics, using eXtreme Gradient Boosting [28] (XGBoost) as the classifier.

We compare the topic representation with two traditional bag-of-words representations: i) Tf-idf values and ii) word counts. To establish a fair comparison, all models use the same vocabulary. Since we have a multi-label task, we employ a One-vs-All approach where we train a binary classifier for each theme and the final classification is the aggregation of all predictions. Formally, let C be the set of all themes, t a threshold value, $f_c(\cdot)$ the decision function of the classifier for class c, and l a lawsuit:

$$\forall c \in C, \text{assign } c \text{ to } l \text{ if } f_c(l) \geq t\,. \tag{1}$$

We set 0.5 as the threshold value.

Finally, we use the validation set to tune the following XGBoost hyperparameters through random search: number of trees, maximum depth and shrinkage factor.

All results are reported on the test set unless otherwise stated. As a baseline method we choose a classifier that assigns all themes to any input, which achieves a F1 score weighted by class frequency of 41.17% and an average F1 score of 5.48%.

6. Results

6.1. Topic Analysis

In order to evaluate the topic quality of the models with 10 and 30 topics we examine the most relevant words and lawsuits from each topic and assign it a label [29]. Table 1 presents the results of the labelling process. For each topic we show its four most relevant words, where relevance is defined [30] as

$$r(\mathbf{w},\mathbf{z}|\lambda) = \lambda \log P(\mathbf{w}|\mathbf{z}) + (1-\lambda) \log \frac{P(\mathbf{w}|\mathbf{z})}{P(\mathbf{w})}\,, \tag{2}$$

and the parameter λ $(0 \leq \lambda \leq 1)$ determines weight given to the probability of term $\mathbf{w}$ given topic $\mathbf{z}$ relative to the ratio between that probability and the marginal probability of **w** on the whole corpus. For each topic, through manual inspection, we select the value with the most descriptive top words, which have been translated to English, except in the case of acronyms and names, which are shown in italic.

Table 1. Topic labels and their respective four most relevant words (10 topics).

Topic	λ	Assigned label	Words
1	0.6	Public servant remuneration	servants, servant, limitation, remuneration
2	0	Criminal Law	narcotic, hydrometer, clandestine, interrogation
3	0.6	Pension Law	benefit, event, retirement, pension
4	0.6	Civil Law	bank, contract, consumer, *projudi*
5	0.6	Right to health	health, city, municipal, medication
6	0.4	OCR errors	*ento*, no, *ro*, *co*
7	0.6	Tax Law	*icms*, *ipi*, tax, income
8	0	Entities	*econorte*, *rcte*, *pieter*
9	0.4	Labor Law	*fgts*, *pss*, hours, payroll
10	0.6	Document access	original, site, access, report

Regarding the model with 10 topics, the results show that most topics are identified with legal matters routinely discussed by the STF. That being said, topics 6 and 8 were challenging to label. The lawsuits with the highest proportion of these topics were useful in that enterprise.

In the first case, the most representative lawsuits were found to contain a great amount of OCR noise. The most relevant suit, with 99.99957% topic 6 content, contains the following passage: “r cm emoi oit incm m t i o i m cofl inoioem oulfl tofl cmcmh co ffl ffl ffl a z a z ffl o t a o u ffl otoidtoaz d to a i o tn ffl em cmcocoulococm eo cocm [...]”, which is pure gibberish.

While examining topic 8, we discovered that its most representative lawsuits contained a lot of named entities; e.g., from the 15 most frequent words in the suit with most topic 8 content, 8 referred to people or organisations.

The model with 30 topics, as shown in Table 2, was also able to identify interpretable topics, many of them directly related to legal matters discussed by the Court. To label each topic, we once again analyze its most relevant words from each topic while varying the value of λ. To label the most challenging topics we also examine their most representative lawsuits. Due to the greater number of topics, some of them deal with much more specific matters than in the case of the model with 10 topics. For example, while the model with fewer topics has only one generic topic for Tax Law, the one with 30 topics has four different topics related to different facets of that legal area (topics 3, 25, 27 and 28).

That said, some of the topics have relevant words that do not belong to related matters. Topic 19, for example, assign high probabilities to words related to both Consumer Law and the Brazilian state of Bahia, with mentions to cities such as Bahia’s capital city Salvador. On the other hand, there are topics with very specific relevant words, such as topic 20, that groups names of people. These results can be explained by the nature of the data, which combines various types of documents; e. g. petitions, judgments, orders, proxy statements, certificates, and other supporting documents. We expect that by training only on the Court’s rulings the topics would be even more related to specific legal matters discusses by the Justices.

Table 2. Topic labels and their respective four most relevant words (30 topics).

Topic	λ	Assigned label	Words
1	0.6	Civil liability	damage, damages, compensation, non-material
2	0.22	Expiration of social security benefit	benefit, expiration, limit, social security (*previdenciário*)
3	0.6	Tax Law	treasury, tax, revenue, taxation
4	0.1	Miscellaneous - Legal vocabulary, enttities and laws	serial number, *pet*, stamp, *itaperuna*
5	0.4	Public servant bonus	bonus, performance, inactive, evaluation
6	0.4	Rural social security	rural, contribution, LEI_8212, pension
7	0.6	Public servant remuneration readjustment	readjustment, servants, remuneration, *urv*
8	0.4	OCR errors	*ento*, no, *ro*, *ffl*
9	0.6	Members of the military	military, servant, servicemen, servants
10	0	Criminal Law	clandestine, *sepetiba*, semi-open, narcotic
11	0.4	Contract law	contract, contracts, fee, accounts
12	0.05	Technical Councils	*confea*, *crea*, agronomy, LEI_6496
13	0.2	Public tender	tender, candidate, notice, openings
14	0.4	Anticipation of remuneration readjustment	*upag*, *pccs*, labor, LEI_8460
15	0.6	Right to health	health, medication (plural), treatment, medication (singular)
16	0.9	Savings account, interest and monetary correction	correction, monetary, savings account, delay
17	0.6	Document access	original, site, acesse, report
18	0.6	labor complaints	*estran*, *tst*, entity, claimant
19	0.4	Miscellaneous - Consumer Law and Bahia (Brazilian state)	consumer, *salvador*, *bahia*, *pdf*
20	0	Entities - names	*lauxen*, *tainá*, *heloise*, *soeli*
21	0.7	Qualification	*num*, normal, internment, *foz*
22	0.5	insurance	insurance, *previd*, institute, *dpu*
23	0.4	Payroll	hours, *fgts*, payroll, overtime
24	0	Miscellaneous - Organisations, charters and non-Portuguese words	*andaterra*, *peixer*, funds, market
25	0.5	Fiscal documents	*ltda*, *ipi*, *nfe*, *icms*
26	0.4	Rio Grande do Sul (Brazilian state)	*sul* , *grande*, *alegre*, *paese*
27	0.4	Income tax	updated, months, *rra*, *irpf*
28	0.2	Tax Law - circulation of goods	compatible, *issqn*, exit, *eireli*
29	0.2	Miscellaneous - Procedure and Paraná (Brazilian state)	*paraná*, *arq*, *curitiba*, *mov*
30	0.4	Payments	*jam*, *vlr*, received, credit

6.2. *Quantitative Analysis*

Figure 2 compares the performance on the validation set of classifiers trained on text features obtained from models with 10, 30, 100, 300 and 1,000 topics. All models greatly outperformed a baseline that simply assigns all themes to each instance. Increasing the dimensionality of the representation up to 300 topics improves performance. The model with 1,000 topics, on the other hand, is comparable to the one with 300.

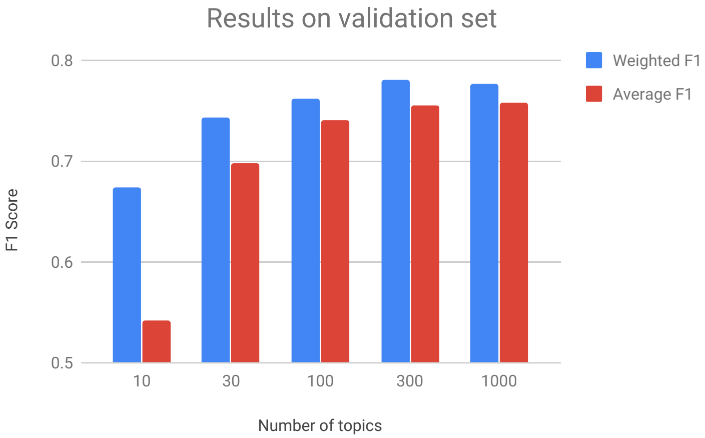


Figure 2. Validation set performance of classifiers trained with different numbers of topics.

Table 3 compares the 300-dimensional lawsuit representation with the word counts and tf-idf values bag-of-words representations on the test set. The topic distribution representation did not outperform the traditional methods, but achieved good performance—much better than the baseline that assigns all themes. These results suggest that the detected topics are related to the themes relevant to the Court and have the potential to aid the judiciary with the management of cases.

Furthermore, it has an advantage over the traditional approaches with respect to the dimensionality of the representation—it describes a lawsuit using 300 dimensions instead of 81,418, a relative reduction of 99.63%. As a result, the training and inference is much faster.

Table 3. F1 scores (in %) on the test set of each text representation method. Assigning all themes to all samples yield a weighted (by class frequency) F1 score of 41.17 and an average F1 score of 5.48.

	Word counts	Tf-idf	300 topics
Weighted	**89.29**	89.22	78.07
Average	87.54	**88.37**	75.81

7. Conclusion

We proposed the use of Latent Dirichlet allocation to build topic models of Extraordinary Appeals from Brazil's Supreme Court (STF). We labelled and analysed the models with 10 and 30 topics, showing the correspondence between them and legal matters that reach the Court. We used the obtained topic distribution vectors as input for a supervised multi-label classification task in order to establish a quantitative analysis of topic relevance. The topic distribution representation, with an optimal value of 300 topics, achieved good results using much lower dimensionality than the traditional methods. The technique can be leveraged to help organize, explore and extract information of the massive amounts of data that reach the Court.

8. Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. TdC received support from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), grant PQ 314154/2018-3. We acknowledge the support of "Projeto de Pesquisa & Desenvolvimento de aprendizado de máquina (machine learning) sobre dados judiciais das repercussões gerais do Supremo Tribunal Federal - STF". We are also grateful for the support from Fundação de Apoio à Pesquisa do Distrito Federal (FAPDF, project KnEDLe, convênio 07/2019) and Fundação de Empreendimentos Científicos e Tecnológicos (Finatec).

References

[1] de Cássia Carvalho Lopes R. migalhas com, editor. Eventual Influences of Common Law on the Brazilian Legal System; 2017. Available at https://www.migalhas.com/HotTopics/63,MI255372,51045-Eventual+Influences+of+Common+Law+on+the+Brazilian+Legal+System Available from: https://www.migalhas.com/HotTopics/63,MI255372,51045-Eventual+Influences+of+Common+Law+on+the+Brazilian+Legal+System.

[2] Secretaria de Comunicação Social do Conselho Nacional de Justiça. CNJ, editor. Sumário Executivo do Relatório Justiça em Números; 2018. Available at http://www.cnj.jus.br/files/conteudo/arquivo/2018/09/da64a36ddee693ddf735b9ec03319e84.pdf.

[3] da Silva NC, Braz FA, de Campos TE, Gusmao DB, Chaves FB, Mendes DB, et al. Document type classification for Brazil's supreme court using a Convolutional Neural Network. In: 10th International Conference on Forensic Computer Science and Cyber Law (ICoFCS). Sao Paulo, Brazil; 2018. Winner of the best paper award.

[4] Luz de Araujo PH, de Campos TE, de Oliveira RRR, Stauffer M, Couto S, Bermejo P. LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text. In: International Conference on the Computational Processing of Portuguese (PROPOR). Lecture Notes on Computer Science (LNCS). Canela, RS, Brazil: Springer; 2018. p. 313–323. Available from: https://cic.unb.br/~teodecampos/LeNER-Br/.

[5] de Vargas Feijó D, Moreira VP. RulingBR: A Summarization Dataset for Legal Texts. In: Villavicencio A, Moreira V, Abad A, Caseli H, Gamallo P, Ramisch C, et al., editors. Computational Processing of the Portuguese Language. Cham: Springer International Publishing; 2018. p. 255–264.

[6] Blei DM. Probabilistic Topic Models. Commun ACM. 2012 Apr;55(4):77–84. Available from: http://doi.acm.org/10.1145/2133806.2133826.

[7] Mauá DD. Modelos de tópicos na classificação automática de resenhas de usuários. [Master thesis]. Escola Politécnica da Universidade de São Paulo; 2009.

[8] Rubin TN, Chambers A, Smyth P, Steyvers M. Statistical topic models for multi-label document classification. Machine Learning. 2012 Jul;88(1):157–208. Available from: https://doi.org/10.1007/s10994-011-5272-5.

[9] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of The American Society for Information Science. 1990;41(6):391–407.

[10] Hofmann T. Probabilistic Latent Semantic Indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR. New York, NY, USA: ACM; 1999. p. 50–57. Available from: http://doi.acm.org/10.1145/312624.312649.

[11] Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. Journal of Machine Learning Research. 2003 Mar;3:993–1022. Available from: http://dl.acm.org/citation.cfm?id=944919.944937.

[12] Blei DM, Lafferty JD. Dynamic Topic Models. In: Proceedings of the 23rd International Conference on Machine Learning. ICML. New York, NY, USA: ACM; 2006. p. 113–120. Available from: http://doi.acm.org/10.1145/1143844.1143859.

[13] Dozier C, Kondadadi R, Light M, Vachher A, Veeramachaneni S, Wudali R. Named entity recognition and resolution in legal text. In: Semantic Processing of Legal Texts. Springer; 2010. p. 27–43.

[14] Cardellino C, Teruel M, Alonso Alemany L, Villata S. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker. In: Proceedints of the 16th International Conference on Artificial Intelligence and Law (ICAIL). London, United Kingdom; 2017. Preprint available from https://hal.archives-ouvertes.fr/hal-01541446.

[15] Kanapala A, Pal S, Pamula R. Text summarization from legal documents: a survey. Artificial Intelligence Review. 2017 Jun;Available from: https://doi.org/10.1007/s10462-017-9566-2.

[16] Galgani F, Compton P, Hoffmann A. Combining Different Summarization Techniques for Legal Text. In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data. HYBRID. Stroudsburg, PA, USA: Association for Computational Linguistics; 2012. p. 115–123. Available from: http://dl.acm.org/citation.cfm?id=2388632.2388647.

[17] Kumar R, Raghuveer K. Legal document summarization using latent dirichlet allocation. International Journal of Computer Science and Telecommunications. 2012;3:114–117.

[18] Kim MY, Xu Y, Goebel R. Summarization of Legal Texts with High Cohesion and Automatic Compression Rate. In: New frontiers in artificial intelligence. Springer; 2013. .

[19] Aletras N, Tsarapatsanis D, Preotiuc-Pietro D, Lampos V. Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective. PeerJ in Computer Science. 2016 10;.

[20] Katz DM, Bommarito I Michael J, Blackman J. Predicting the Behavior of the Supreme Court of the United States: A General Approach. arXiv e-prints. 2014 Jul;p. arXiv:1407.6333.

[21] Şulea OM, Zampieri M, Vela M, van Genabith J. Predicting the Law Area and Decisions of French Supreme Court Cases. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP. INCOMA Ltd.; 2017. p. 716–722. Available from: https://doi.org/10.26615/978-954-452-049-6_092.

[22] Undavia S, Meyers A, Ortega JE. A Comparative Study of Classifying Legal Documents with Neural Networks. In: Federated Conference on Computer Science and Information Systems (FedCSIS); 2018. p. 515–522.

[23] Carter DJ, Brown J, Rahmani A. Reading the high court at a distance: Topic modelling the legal subject matter and judicial activity of the high court of Australia, 1903-2015. UNSWLJ. 2016;39:1300.

[24] Remmits Y. Finding the Topics of Case Law: Latent Dirichlet Allocation on Supreme Court Decisions [Bachelor's Thesis]; 2017. Bachelor's thesis, Radboud University, July 2017.

[25] O'Neill J, Robin C, O'Brien L, Buitelaar P. An analysis of topic modelling for legislative texts. In: Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts; 2016. .

[26] Luz de Araujo PH, de Campos TE, Braz FA, Correia da Silva N. Victor: a dataset for Brazilian legal documents classification. In: Proceedings of The 12th Language Resources and Evaluation Conference (LREC). Marseille, France: European Language Resources Association; 2020. p. 1449–1458. Source code, dataset and further information available from https://cic.unb.br/~teodecampos/ViP/lrec/. Available from: https://www.aclweb.org/anthology/2020.lrec-1.181.

[27] Hoffman MD, Blei DM, Bach F. Online Learning for Latent Dirichlet Allocation. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1. NIPS. USA: Curran Associates Inc.; 2010. p. 856–864. Available from: http://dl.acm.org/citation.cfm?id=2997189.2997285.

[28] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD. New York, NY, USA: ACM; 2016. p. 785–794. Available from: http://doi.acm.org/10.1145/2939672.2939785.

[29] Grimmer J, Stewart BM. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. Political Analysis. 2013;21(3):267–297.

[30] Sievert C, Shirley K. LDAvis: A method for visualizing and interpreting topics. In: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. Baltimore, Maryland, USA: Association for Computational Linguistics; 2014. p. 63–70. Available from: https://www.aclweb.org/anthology/W14-3110.

doi:10.3233/FAIA200856

Multilingual Legal Information Retrieval System for Mapping Recitals and Normative Provisions

Rohan NANDA [a,b,1], Llio HUMPHREYS [c], Lorenzo GROSSIO [d] and Adebayo KOLAWOLE JOHN [e]

[a] *Maastricht Law and Tech Lab, Faculty of Law, Maastricht University, Netherlands*
[b] *Institute of Data Science, Maastricht University, Netherlands*
[c] *Computer Science Department, University of Turin, Italy*
[d] *Law Department, University of Turin, Italy*
[e] *Businesspoint Intelligence Solutions Ltd, Ireland*

Abstract. This paper presents a multilingual legal information retrieval system for mapping recitals to articles in European Union (EU) directives and normative provisions in national legislation. Such a system could be useful for purposive interpretation of norms. A previous work on mapping recitals and normative provisions was limited to EU legislation in English and only one lexical text similarity technique. In this paper, we develop state-of-the-art text similarity models to investigate the interplay between directive recitals, directive (sub-)articles and provisions of national implementing measures (NIMs) on a multilingual corpus (from Ireland, Italy and Luxembourg). Our results indicate that directive recitals do not have a direct influence on NIM provisions, but they sometimes contain additional information that is not present in the transposed directive sub-article, and can therefore facilitate purposive interpretation.

Keywords. legal information retrieval, recitals, European legislation, interpretation

1. Introduction

It is well known in the AI & Law community that norms require legal interpretation: ‘It is clear that these documents are not themselves the law from the fact, that we must first interpret statutes and cases to get at the law which they represent, and from the fact that reasonable persons can disagree as to just what the law is, although there is rarely disagreement as to what, words make up the statute or case in question.’ [5], page 2. Canons of interpretation have been used in Civil and Common Law countries, while the European Court of Justice [9] recommends resolving ambiguous, imprecise or incomplete norms with purposive interpretation (i.e. taking account of the purpose of the norm). Therefore, it is important to note the holistic character of the law in that the meaning of normative provisions often emerges from a wider legislative corpus.

[1] Corresponding Author: Rohan Nanda, Maastricht University, Bouillonstraat 3, 6211 LH Maastricht; E-mail:r.nanda@maastrichtuniversity.nl

This paper is concerned with ‘hidden’ links between norms in EU directives and norms in the legislation that transpose them into national law, known as national implementing measures (NIMs). Hidden links are implicit links which are not explicitly referred to within the text of the normative provision via long- or short-form citation. Conceptually similar is one such type of link, but there are others, such as Constitutive, Motivation, Impact etc [3]. Moreover, some of the text in the recitals of the preambles of directives are remarkably similar to some (sub-)articles in the same directive, and are also not made explicit. Recitals can provide additional information and citations to justify the norms in the directive. Den Heijer et al. [1] found that the use of recitals often do not correspond to their stated objectives in official drafting rules [2], and are more significant than commonly appreciated. Due to space constraints, the readers may refer to [6] and [3]. Incidentally, the equivalent to recitals in the countries we looked at consist of generic procedural references with no reference to specific subject-matter of the NIMs. That may not be the case in other Member States and maybe the subject of future work.

There are many different kinds of possible relationships between legal provisions (see [15] and [3]). Previous work on mapping recitals and normative provisions was limited to EU legislation in English and utilized only one text similarity technique [6]. In this paper, we propose, develop and validate a multilingual legal information retrieval system for mapping conceptually similar directive recitals, directive (sub-)articles and NIM provisions. We develop state-of-the-art syntactic and semantic text similarity models to identify conceptually similar norms. The multilingual information retrieval system was validated by evaluating the text similarity techniques on the gold standard mappings (between norms) over a multilingual parallel corpus of 5 directives and their corresponding NIMs from Luxembourg, Ireland and Italy. Our research questions are as follows:

- RQ1) How are directive recitals related to the provisions of the National Implementing Measures (NIMs)?
- RQ2) Which automated text similarity techniques are best able to capture conceptually similar directive recitals, directive (sub-)articles and NIM provisions?
- RQ3) The NIMs of which Member State are most/least semantically correlated with recitals?

2. Methodology

Figure 1 presents the overall workflow of our methodology.

2.1. Corpus generation

A multilingual parallel corpus of 43 directives and their corresponding NIMs from Luxembourg, Ireland and Italy (15,400 norms) was presented in [12]. This corpus only contains mappings between directive (sub-)articles and NIM provisions. It does not contain mappings between directive recitals and directive (sub-)articles, and between directive recitals and NIM provisions. Since the preparation of such fine-grained mappings for the entire multilingual corpus of 43 directives is a highly time-consuming and expensive process, we selected 5 directives and their corresponding NIMs from this corpus. The list of selected directives and their corresponding NIMs is presented in Table 1, and can be found on eur-lex.europa.eu by searching for the CELEX numbers in that table. The title

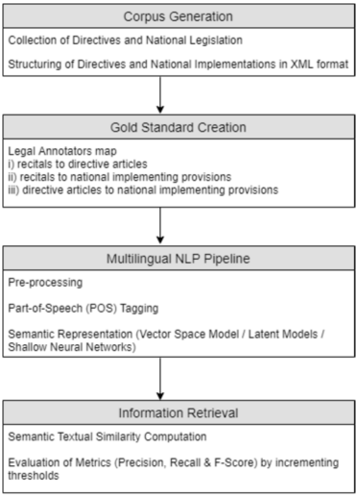

Corpus Generation
Collection of Directives and National Legislation
Structuring of Directives and National Implementations in XML format
Gold Standard Creation
Legal Annotators map
i) recitals to directive articles
ii) recitals to national implementing provisions
iii) directive articles to national implementing provisions
Multilingual NLP Pipeline
Pre-processing
Part-of-Speech (POS) Tagging
Semantic Representation (Vector Space Model / Latent Models / Shallow Neural Networks)
Information Retrieval
Semantic Textual Similarity Computation
Evaluation of Metrics (Precision, Recall & F-Score) by incrementing thresholds

Figure 1. Overall workflow of the multilingual legal information retrieval system

and references of the NIMs for a particular directive can also be obtained on the EUR-Lex portal. This information can be used to obtain the full-text version on the websites of the national legislation portal. The directive recitals, directive (sub-)articles and NIM provisions were all stored in separate XML files for each directive and language.

Table 1. The CELEX numbers of directives and NIMs in the corpus

Directive	NIM (Ireland)	NIM (Luxembourg)	NIM (Italy)
32003L0010	72003L0010IRL_133619	72003L0010LUX_142437	72003L0010ITA_132468
32002L0044	72002L0044IRL_133618	72002L0044LUX_142436	72002L0044ITA_124474
32001L0024	72001L0024IRL_180124 72001L0024IRL_28393	72001L0024LUX_114418	72001L0024ITA_30729
31999L0092	71999L0092IRL_111679	71999L0092LUX_120249	71999L0092ITA_111680
32001L0113	72001L0113IRL_116060	72001L0113LUX_116062	72001L0113ITA_116061

2.2. Gold standard creation

The gold standard corpus was prepared by a researcher in legal informatics, and then checked by a legal expert. The following mappings were prepared for each language: i) directive (sub-)articles to NIM provisions; ii) directive recitals to directive (sub-)articles; and iii) directive recitals to NIM provisions. Mappings were assigned whenever there was content in the two norms that used similar or different wording to express more or less the same content. It was not deemed necessary that the whole text of the norms should be conceptually similar, only a part of it. This is because the similar parts help identify related norms, while the non-similar part should add further information and thus help in interpretation.

2.3. Multilingual NLP pipeline for mapping norms

We developed a multilingual NLP pipeline for mapping directive recitals, directive (sub-)articles and NIM provisions (in this section, described as norms). Using spaCy (https://spacy.io/) tokenizers, we segmented the legal norms into sentences and words, and converted the tokens into lowercase. We removed common noisy words (using spaCy's default list of stopwords) as well as punctuation. We did not select words to retain based on their part-of-speech tag because in the case of short text similarity models (as we are comparing legal norms instead of documents), we need to utilize all the available linguistic features to achieve an acceptable magnitude of text similarity [12]. The text representation phase then encodes the linguistic (syntactic and semantic) features.

For syntactic text representation tests, we utilized: Term Frequency-Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Unifying Similarity Measure (USM)[11] - which combines cosine similarity, N-gram similarity, and approximate string matching, weighted with an arithmetic mean. A dimension size of 50 was chosen for the LSA as it yielded the best performance. The number of topics for the LDA was set to 500 (as it achieved best performance). We used a gram-size of 4 for N-gram similarity in the USM. For our semantic text representation tests, we utilized: FastText (https://fasttext.cc/) , the Paragraph Vector model [4], and DistilBERT, a lighter model of BERT (Bidirectional Encoder Representations from Transformers) [14]. We used two versions of FastText embeddings: 1)FT-Legal: trained on the complete multilingual parallel corpus including 4,300 directives in English, French and Italian and 27,365 NIMs from Ireland, 14,365 from Luxembourg and 16,233 from Italy, and 2) FT-Generic (https://fasttext.cc/docs/en/crawl-vectors.html): pre-trained on Common Crawl and Wikipedia. We used the word-average method, which divides the sum of the word embeddings in a legal norm by the norm length. The embedding dimension size was set to 128. The default hyperparameters were: context window: 5, number of negative samples: 5 and learning rate: 0.1. The Paragraph Vector model [4] was trained on the same multilingual parallel corpus as FastText and used the same embedding dimension size of 128. For the pre-trained DistilBERT embeddings, we used the spaCy-sentence-transformers (https://spacy.io/universe/project/spacy-sentence-bert) library to obtain fixed-length (768 dimensions) legal norm vectors.

2.4. *Information retrieval*

After obtaining the legal norm vectors (see Section 2.3), we compute a cosine similarity measure between them. For instance, to find the most similar NIM provision for a particular directive recital *R1* (case 1 in the above list), a cosine similarity score is computed between the directive recital vector of *R1* and the provisions of the relevant NIM. The NIM provision vectors with a cosine similarity value greater than or equal to the threshold value are retrieved. Each similarity measure is evaluated by comparing the retrieved legal norms with gold standard mappings. Evaluation metrics recall, precision and F-Score are computed for the three types of mappings: directive (sub-)articles to NIM provisions, directive recitals to directive (sub-)articles, and directive recitals to NIM provisions. Evaluation metrics are recorded by incrementing threshold values from 0 to 1 (the increment interval is set to 0.01). The threshold which yields the best F-Score is chosen.

3. Results and analysis

In this section, we present the results of the evaluation of different text similarity measures. The macro-average precision, recall and F-Score are computed for each mapping type across all three languages. For instance, to map recitals with the NIM provisions of Luxembourg (written in French), the European directive in French was utilized. We discuss the results of each mapping type in the following subsections.

3.1. Mappings between directive recitals and NIM provisions

Figure 2 presents the macro-average precision, recall and F-Score metrics of various text similarity techniques over multilingual mappings between directive recitals and NIM provisions. We observe that the Luxembourg mappings achieved a higher F-Score than the Italian and the English (Ireland) mappings for each similarity measure. This is because of the presence of more common words and phrases between the French directive recitals and the Luxembourg NIM provisions. We also observed that in case of Irish English-language legislation, the precision is much lower than for the French-language (Luxembourg) and Italian legislation. On the other hand, the recall was the highest in the English language. This is because of a higher number of NIM provisions in Ireland (as shown in Table 2) compared to Luxembourg and Italy.

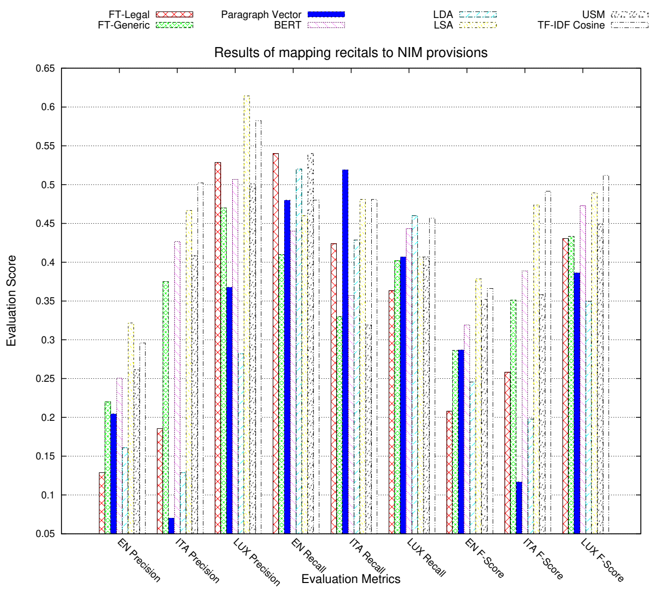


Figure 2. Comparison of the semantic textual similarity techniques for mapping directive recitals and NIM provisions

Due to a high number of provisions in the Irish legislation, the directive recitals shared a decent magnitude of similarity to many unrelated NIM provisions that were not included in the gold standard mapping. This resulted in a higher number of false positives, which led to a low precision score. TF-IDF Cosine and LSA text similarity techniques outperformed other techniques in terms of F-Score in all three languages. This shows that a large number of mappings between directive recitals and NIM provisions can be identified by weighting important terms through TF-IDF and LSA transform. The performance of semantic text similarity models like FastText (FT-Legal and FT-Generic), Paragraph Vector and BERT was comparatively poorer. However, it is important to note that the performance of BERT model was superior to both FastText and Paragraph Vector. We also observed that the performance of the pre-trained FT-Generic was slightly superior to the domain-specific FT-Legal embeddings.

The best overall macro-average F-Score values are 0.5119, 0.4914 and 0.3786 for mappings between directive recitals and the NIM provisions of Luxembourg, Italy and Ireland respectively. These results indicate that the majority of directive recitals do not share a high degree of semantic similarity with the NIM provisions.

Table 2. The number of provisions in the NIM corpus of each piece of legislation

Ireland NIMs	Luxembourg NIMs	Italian NIMs
269	194	146

3.2. Mappings between directive (sub-)articles and NIM provisions

Figure 3 presents the macro-average precision, recall and F-Score for the mappings between directive (sub-)articles and NIM provisions.These results indicate that the Luxembourg directive mappings consistently achieved a higher recall, precision and F-Score than Italian and English ones for all the similarity measures. This is consistent with the research presented in [12]. The best overall macro-average F-Score values are 0.8243, 0.7276 and 0.6712 for mappings between directive (sub-)articles and the NIM provisions of Luxembourg, Italy and Ireland respectively. The best performance was achieved by TF-IDF Cosine similarity measure in all three languages. Mappings between directive (sub-)articles achieved a much higher F-Score compared to mappings between directive recitals and NIM provisions in all three languages. This is quite intuitive because directive (sub-)articles are supposed to be transposed into the national legislation of Member States. There is no obligation to transpose directive recitals into NIM provisions

3.3. Mappings between directive recitals and directive (sub-)articles

Figure 4 presents macro-average precision, recall and F-Score metrics for mappings between directive recitals and directive (sub-)articles by the best performing measure, TF-IDF Cosine. Gold standard mappings for directive recitals and directive (sub-)articles are the same for all three languages because directives have the same structure and content in all EU languages. The similar F-Scores in this case signify that these mappings are not influenced by language differences. Further, it indicates that these mappings can be identified with the same F-Score in different languages. This result also validates our text similarity and gold standard mapping approach. The minor differences in the F-Score are due to the different NLP pipeline models used for tokenization and splitting sentences.

3.4. Discussion

The results from section 3.1 and 3.2 indicate that the French language mappings had the best F-Scores. Table 3 shows an example of a directive recital / directive sub-article / NIM

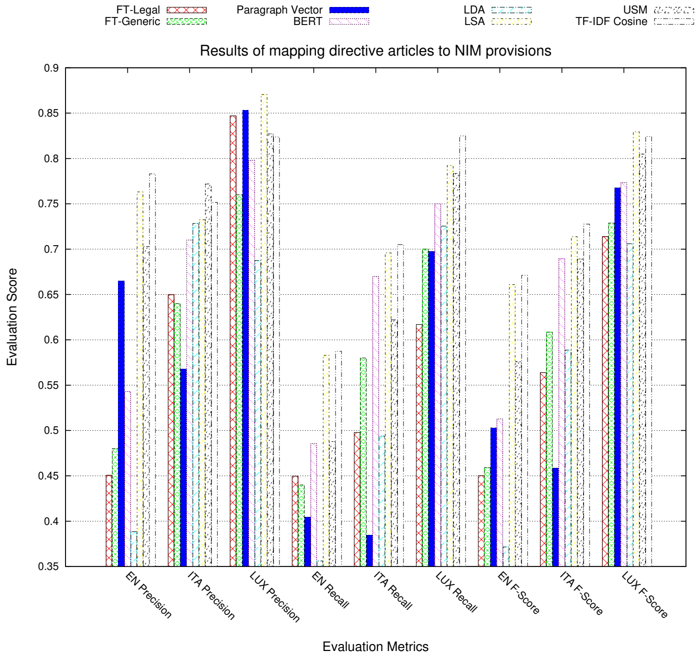


Figure 3. Comparison of the semantic textual similarity techniques for mapping directive (sub-)articles and NIM provisions

Table 3. Similar directive recital, directive sub-article and NIM provision

32002L0044 Directive Recital 8: Pour les secteurs de la navigation maritime et aérienne, dans l'état actuel de la technique, il n'est pas possible de respecter, dans tous les cas, les valeurs limites d'exposition relatives aux vibrations transmises à l'ensemble du corps. Il y a donc lieu de prévoir des possibilités de dérogations dûment justifiées.
32002L0044 Directive Article 10.1: Dans le respect des principes généraux de la protection de la sécurité et de la santé des travailleurs, les États membres peuvent, pour les secteurs de la navigation maritime et aérienne, dans des circonstances dûment justifiées, déroger à l'article 5, paragraphe 3, en ce qui concerne les vibrations transmises à l'ensemble du corps, lorsque, compte tenu de l'état de la technique et des caractéristiques spécifiques des lieux de travail, il n'est pas possible de respecter la valeur limite d'exposition malgré la mise en œuvre de mesures techniques et/ou organisationnelles.
72002L0044LUX_142436 NIM Provision 9.1: Le ministre ayant le travail dans ses attributions peut donner une dérogation à l'article 5, paragraphe 3, dans le respect des principes généraux de la protection de la sécurité et de la santé des travailleurs, pour les secteurs de la navigation maritime et aérienne, dans des circonstances dûment justifiées, en ce qui concerne les vibrations transmises à l'ensemble du corps, lorsque, compte tenu de l'état de la technique et des caractéristiques spécifiques des lieux de travail, il n'est pas possible de respecter la valeur limite d'exposition malgré la mise en œuvre de mesures techniques et/ou organisationnelles.

provision triplet from the French corpus (common texts higlighted). The best performing measure, TF-IDF Cosine was able to identify all three mappings.

There are some cases where the directive recital and NIM provision that are mapped to a directive (sub-)article do not share similar content. An example from the English corpus is shown in table 4 (the similar content between directive recital 9 and directive article 8 are highlighted in yellow, while the similar content between directive article 8 and NIM provision 5.3 are highlighted in green). TF-IDF Cosine identified the link between recital 9 and article 8 (of the directive), and also between directive article 8 and provision 5.3 of the NIM. However, due to the lack of semantic overlap between directive recital 9 and NIM provision 5.3, no similarity link was identifiedby TF-IDF Cosine. We did not find evidence of direct influence of directive recitals on NIM provisions, which

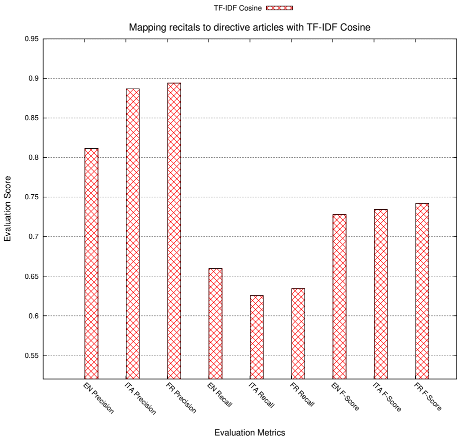


Figure 4. Evaluation metrics for mapping directive recitals to directive articles with TF-IDF Cosine (best performing text similarity measure)

Table 4. Directive article sharing different similar content with directive recital and NIM provision

31999L0092 Directive Recital 9 The establishment of a coherent strategy for the prevention of explosions requires that organisational measures complement the technical measures taken at the workplace; Directive 89/391/EEC requires the employer to be in possession of an assessment of the risks to workers' health and safety at work; this requirement is to be regarded as being specified by this Directive in that it provides that the employer is to draw up an explosion protection document, or set of documents, which satisfies the minimum requirements laid down in this Directive and is to keep it up to date; the explosion protection document includes the identification of the hazards, the evaluation of risks and the definition of the specific measures to be taken to safeguard the health and safety of workers at risk from explosive atmospheres, in accordance with Article 9 of Directive 89/391/EEC; the explosion protection document may be part of the assessment of the risks to health and safety at work required by Article 9 of Directive 89/391/EEC.
31999L0092 Directive Article 8 In carrying out the obligations laid down in Article 4, the employer shall ensure that a document, hereinafter referred to as the "explosion protection document", is drawn up and kept up to date. The explosion protection document shall demonstrate in particular: that the explosion risks have been determined and assessed, that adequate measures will be taken to attain the aims of this Directive, those places which have been classified into zones in accordance with Annex I, those places where the minimum requirements set out in Annex II will apply, that the workplace and work equipment, including warning devices, are designed, operated and maintained with due regard for safety, that in accordance with Council Directive 89/655/EEC(10), arrangements have been made for the safe use of work equipment. The explosion protection document shall be drawn up prior to the commencement of work and be revised when the workplace, work equipment or organisation of the work undergoes significant changes, extensions or conversions. The employer may combine existing explosion risk assessments, documents or other equivalent reports produced under other Community acts.
71999L0092IRL 111679 NIM Provision 5.3 The risk assessment shall be reviewed by the employer regularly so as to keep it up to date and particularly if— there is reason to suspect that the risk assessment is no longer valid; or there has been a significant change in the matters to which the risk assessment relates including when the workplace, work processes, or organisation of the work undergoes significant changes, extensions or conversions; and where, as a result of the review, changes to the risk assessment are required, those changes shall be made.

would have been evidenced for any triplets by text present in the directive recital and NIM provisions that is absent from the directive article. However, we did find examples of directive recitals containing additional information to related directive (sub-)articles, which can aid purposive interpretation - see table 5 (additional information is in bold).

Table 5. Similar directive recital, directive article and NIM provision, with additional information in the directive recital

32003L0010 Directive Recital 7: Come secondo passo, si ritiene opportuno introdurre misure di protezione dei lavoratori contro i rischi derivanti dal rumore a causa dei suoi effetti sulla salute e sulla sicurezza dei lavoratori, in particolare per quanto riguarda i danni all'udito. **Tali misure mirano non solo ad assicurare la salute e la sicurezza di ciascun lavoratore considerato individualmente, ma anche a creare per tutti i lavoratori della Comunità una piattaforma minima di protezione che eviti possibili distorsioni di concorrenza.**
32003L0010 Directive Article 1: La presente direttiva, che è la diciassettesima direttiva particolare a norma dell'articolo 16, paragrafo 1, della direttiva 89/391/CEE, stabilisce prescrizioni minime di protezione dei lavoratori contro i rischi per la loro salute e sicurezza che derivano, o possono derivare, dall'esposizione al rumore e, segnatamente, contro il rischio per l'udito.
72003L0010ITA_132468 NIM Provision 49_bis: Il presente titolo determina i requisiti minimi per la protezione dei lavoratori contro i rischi per la salute e la sicurezza derivanti dall'esposizione al rumore durante il lavoro e in particolare per l'udito.

4. Related work

Most work on links between norms in legislation e.g. [13][16] focus on the discovery and classification of explicit citations. Amantea et al. [3] proposed a model for classifying different kinds of implicit links between directive recitals and directive (sub-)articles including Conceptually Similar, Constitutive, Motivation and Impact. The authors suggested that different kinds of algorithms are required to identify each kind of link, but none were tested. Humphreys et al. [6] mapped recitals to legal articles (but not sub-articles) in EU legislation based on conceptual similarity. Norms were modeled as TF-IDF vectors and similarity was computed based on Cosine Similarity. The system achieved a high recall but low precision. The high accuracy achieved was due to the unbalanced dataset, with a great number of true negatives. Nanda et al. [11][12] investigated automated mapping of directive (sub-)articles to NIM provisions using a variety of similarity algorithms suited for short text including matching common words, common sequences of words and approximate string matching. The relevance of directive recitals was not considered for this work. The work of Lau [8] concerns finding similar provisions in different legislation in the US. A list of the most similar pairs of provisions are produced based on the similarity of parsed norms as well as associated features including legislative definitions and glossaries from reference books. Kumar et al. [7] also used a range of factors to find similar judgments from the Supreme Court of India including headnote, citation and case citation. The most important features were legal terms and citations. Legal-term cosine similarity performed better than all-term cosine similarity.

5. Conclusions and future work

This paper was concerned with ‘hidden’ links between norms in EU directives and national implementing measures. Automated identification of such links could facilitate purposive interpretation and monitoring of implementation. We focussed on identifying conceptually similar norms, and evaluated the performance of suitable text similarity techniques. Since the preparation of fine-grained provision mappings is time-consuming and expensive, we limited our experiments to five directives and their corresponding NIMs. Out of many text similarity techniques, the best performing model was TF-IDF Cosine Similarity. This is consistent with other research in the legal domain [12]. The semantic text representation methods in particular performed adequately for Luxembourg but poorly for the other countries. We found conceptually similar directive recitals, directive (sub-)articles and NIM provisions in all five directives and related NIMs in Ireland, Luxembourg and Italy. However, there was less similarity between directive recitals and NIM provisions, since NIMs are meant to transpose (sub-)articles, not recitals. We did not find evidence of direct influence of directive recitals on NIM provisions. However, we did find directive recitals that contain additional information that can facilitate purposive reasoning. The degree of similarity between directive recitals and NIM provisions varied according to country in exactly the same way as for directive (sub-articles), with NIMs from Luxembourg bearing the highest similarity, and NIMs from Ireland the lowest. The similar F-score in mappings between directive recitals and directive (sub-)articles for different language versions of the same directive shows that our approach is generally sound. However, one reason for the low F-score for mappings to Irish NIMs could be imprecision and inconsistency in EU English legal language due to the legal drafting being carried out by non-native English speakers who are unfamiliar with Common Law systems and terminology [17]. Our future work will investigate whether mapping equivalent terms in EU directives and NIMs (through an ontology [10]) can improve the performance of the text similarity system and facilitate the detection of conceptually similar normative provisions.

References

[1] Maarten den Heijer, Teun van Os van den Abeelen, and Antanina Maslyka. On the use and misuse of recitals in European Union law. *Amsterdam Law School Research Paper*, (2019-31), 2019.

[2] European Commission. Joint Practical Guide of the European Parliament, the Council and the Commission for Persons Involved in the Drafting of European Union Legislation. 2016.

[3] Ilaria Angela Amantea, Luigi Di Caro, Llio Humphreys, Rohan Nanda, and Emilio Sulis. Modelling norm types and their inter-relationships in EU directives. In *ASAIL@ ICAIL*, 2019.

[4] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[5] Thomas F Gordon. The role of exceptions in models of the law. *Formalisierung im Recht und Ansätze juristischer Expertensysteme*, pages 52–59, 1986.

[6] Llio Humphreys, Cristiana Santos, Luigi Di Caro, Guido Boella, Leon Van Der Torre, and Livio Robaldo. Mapping recitals to normative provisions in EU legislation to assist legal interpretation. In *JURIX*, pages 41–49, 2015.

[7] Sushanta Kumar, P Krishna Reddy, V Balakista Reddy, and Aditya Singh. Similarity analysis of legal judgments. In *Proceedings of the Fourth Annual ACM Bangalore Conference*, pages 1–4, 2011.

[8] Gloria T Lau, Kincho H Law, and Gio Wiederhold. Similarity analysis on government regulations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 711–716, 2003.

[9] Koen Lenaerts and José A Gutiérrez-Fons. To say what the law of the EU is: methods of interpretation and the European Court of Justice. *Colum. J. Eur. L.*, 20:3, 2013.

[10] Gianmaria Ajani, Guido Boella, Luigi Di Caro, Livio Robaldo, Llio Humphreys, Sabrina Praduroux, Piercarlo Rossi, and Andrea Violato. The European legal taxonomy syllabus: a multi-lingual, multi-level ontology framework to untangle the web of European legal terminology. *Applied Ontology*, 11(4):325–375, 2016.

[11] Rohan Nanda, Luigi Di Caro, Guido Boella, Hristo Konstantinov, Tenyo Tyankov, Daniel Traykov, Hristo Hristov, Francesco Costamagna, Llio Humphreys, Livio Robaldo, et al. A unifying similarity measure for automated identification of national implementations of European Union directives. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 149–158, 2017.

[12] Rohan Nanda, Giovanni Siragusa, Luigi Di Caro, Guido Boella, Lorenzo Grossio, Marco Gerbaudo, and Francesco Costamagna. Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives. *Artificial Intelligence and Law*, 27(2):199–225, 2019.

[13] Ali Sadeghian, Laksshman Sundaram, D Wang, W Hamilton, Karl Branting, and Craig Pfeifer. Semantic edge labeling over legal citation graphs. In *Proceedings of the workshop on legal text, document, and corpus analytics (LTDCA-2016)*, pages 70–75, 2016.

[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*, 2019.

[15] Daniela Tiscornia and Fabrizio Turchi. Formalization of legislative documents based on a functional model. In *Proceedings of the 6th international conference on Artificial intelligence and law*, pages 63–71, 1997.

[16] Bernhard Waltl, Jörg Landthaler, and Florian Matthes. Differentiation and empirical analysis of reference types in legal documents. In *JURIX*, pages 211–214, 2016.

[17] Aleksandra Čavoški. Interaction of law and language in the EU: Challenges of translating in multilingual environment." *Perspectives: The journal of specialised translation* 27:58–74. 2017

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200857

Extracting Outcomes from Appellate Decisions in US State Courts

Alina PETROVA [1], John ARMOUR and Thomas LUKASIEWICZ
University of Oxford, UK

Abstract. Predicting the outcome of a legal process has recently gained considerable research attention. Numerous attempts have been made to predict the exact outcome, judgment, charge, and fines of a case given the textual description of its facts and metadata. However, most of the effort has been focused on Chinese and European law, for which there exist annotated datasets. In this paper, we introduce CASELAW4 — a new dataset of 350k common law judicial decisions from the U.S. Caselaw Access Project, of which 250k have been automatically annotated with binary outcome labels of AFFIRM or REVERSE by our hybrid learning system. To our knowledge, it is the first attempt to perform outcome extraction (a) on such a large volume of English-language judicial opinions, (b) on the Caselaw Access Project data, and (c) on US State Courts of Appeal cases, and it paves the way to large-scale outcome prediction and advanced legal analytics using U.S. Case Law. We set up baseline results for the outcome extraction task on the new dataset, achieving an F-measure of 82.32%.

Keywords. legal analytics, outcome extraction, legal reasoning, outcome prediction

1. Introduction

Legal analytics – the application of computational methods to legal materials – has recently become a topic of global research interest. It offers potential to improve access to justice, automate repetitive administrative tasks, reduce legal costs, and bring transparency to judicial procedures [4]. Considerable research effort has recently been devoted to case outcome prediction — the task of predicting the outcome of a court's decision in a particular case (i.e., verdict, sentence, charge, or fine) given the factual background of the case [13,12,17,22,24]. Legal analytics requires a sufficiently large-scale dataset of case information, including facts and outcomes. However, legal data is usually stored in textual form with limited metadata. In particular, the outcome of a case is rarely stated explicitly in the case report and has to be extracted from text manually or (semi-)automatically.

In this paper, we investigate the problem of large-scale outcome extraction from common law judicial decisions. We introduce CASELAW4 — a novel dataset of 350k U.S. decisions from state Courts of Appeal, sourced from the Caselaw Access Project [8] that are annotated with outcomes. The annotation has been done in part manually but primarily with a hybrid outcome extraction model that reaches an F-measure of 82.32%.

[1] Corresponding Author: Alina Petrova; E-mail: alina.petrova@cs.ox.ac.uk.

Both the annotated data and the model are publicly available, and they act as baselines for outcome extraction both for opinions from US state Courts of Appeal cases and for the U.S. case law more generally.

2. Related Work

2.1. Legal Information Extraction

The works on legal information extraction are limited, and they adopt techniques from general-domain NLP. CAIL2018 [10], the largest publicly available dataset on Chinese Criminal Law, spurred works in legal event extraction and named entity recognition [26,30]. Few works focused on extracting particular types of clause sentences, e.g., sentences containing statutory terms [27], confidentiality clauses,[2] or even outcome sentences.[3] Unfortunately the latter proved to work poorly on appeal outcomes. For U.S. court data, prior work on outcome extraction has been done manually or semi-manually using dockets of US Federal Courts [28,29].

2.2. Legal Outcome Prediction

Legal outcome prediction is one of the most actively researched tasks in legal natural language processing. Previous works focused mostly on European and Chinese law. They include predicting outcomes in the French Supreme Court [18], in the European Court of Justice [14,19], and in the European Court of Human Rights [13,12,15,16], as well as predicting outcomes of criminal cases from the Supreme People's Court of China [10,20,21,22,23,24,25]. However, very limited work focused on the U.S. and U.K. law systems [9,17], and to our knowledge, no attempt has yet been made to predict outcomes for cases from the CAP dataset [8].

2.3. RNNs and LSTMs

In this section, we motivate our choice of the machine learning algorithm that we used in Section 4 in order to train a baseline outcome prediction model on the CASELAW4 dataset. Textual documents, such as court proceedings and case reports, are a type of sequential data. Sequential inputs have two important properties: (1) they do not have fixed size, and (2) later input typically depends on earlier one. For example, a word at position t in a sentence may depend on various other words at positions $t-n$ and even at positions $t+m$, with $n,m > 0$.

While in general deep learning models are successfully applied to natural language-related tasks, one type of deep learning models — recurrent neural networks (RNNs) — is specifically tailored to handle sequential input. Among various RNN architectures, long short-term memory (LSTM) models [1] perform particularly well, as they mitigate the problem of vanishing and exploding gradients in the network. The key component of an LSTM is the memory cell that contains self-recurrent connection as well as three gates (input, output, and forget), that regulate which information is kept in the cell, which

[2] `https://github.com/LexPredict/lexpredict-lexnlp`

[3] `https://github.com/ICLRandD/Blackstone`

is passed further, and which is ignored, respectively, while the model reads the input text word by word. Finally, bidirectional LSTMs (bi-LSTMs) [2] are a variation of LSTMs that read the input twice, in the original and in the reversed order, which allows them to take into account not only the preceding information, but the information further in time; this ability is typically beneficial when processing textual data. LSTMs and bi-LSTMs are considered to be the state-of-the-art models for numerous natural language processing tasks [5,6,7], including those in the legal domain [19,21,20]. It is reasonable to expect (bi-)LSTMs to efficiently capture key phrases and words that manifest legal outcomes in the appeal setting, such as *we reverse* or *is therefore affirmed*, hence we chose bi-LSTMs as baseline models for outcome extraction task (see Section 4).

3. Dataset

The Caselaw Access Project (CAP) is the largest publicly available dataset of U.S. court decisions [8]. It is maintained by the Harvard Law School. CAP consists of nearly 7 million case reports from all US state, federal, and territorial courts and covers the time period of 1658−2018. Each report contains metadata on the hearing, court of hearing, jurisdiction, judges and attorneys, as well as the full text of the court's decision. Each report typically contains a review of key facts and previous court rulings, the legal reasoning applied by the court, and the verdict; it may also contain corrections and dissenting opinions. The reports are in unstructured form, but occasionally may contain section headings, e.g., *Facts* or *Conclusion*.

We have used a subset of CAP, CASELAW4, that consists of over 350,000 court case reports from New Mexico, North Carolina, Illinois, and Arkansas Courts of Appeal. These Courts hear appeals exclusively from lower courts within their respective states, on matters of domestic state law. The data for these jurisdictions are freely downloadable from the CAP website.[4] An example of a case report from CASELAW4 is presented in Figure 1. Since each case in CASELAW4 appeals some lower court ruling, the possible outcomes of each case are as follows:

- the previous ruling is kept as is (AFFIRM);
- the previous ruling is changed/annulled (REVERSE);
- some parts of the previous ruling are kept and some are changed (MIXED);
- the appeal is dismissed (a type of AFFIRM).

We intentionally treat cases with a clear-cut decision (AFFIRM and REVERSE) separately from more complex ones (MIXED), as it is common to establish outcome prediction baseline results first on simpler cases and only then move on to more complex, non-binary ones [13,17], and we foresee this as an avenue for future work.

Table 1 summarizes the dataset statistics, and Table 2 shows the distribution of cases depending on their length (as measured by the word count of the main body of the case, without dissenting opinions). As can be seen from Table 2, the cases vary a lot in length. We assume that the length of a case report is a fair estimation of the case's complexity: shorter case reports tend to either reinstate the decision of the first instance court (AFFIRM) or to give a clear reason why the existing ruling should be reversed

[4] `https://case.law/download/bulk_exports/20200604/by_jurisdiction/case_text_open/`

Table 1. Overview of CASELAW4

	New Mexico	Arkansas	North Carolina	Illinois	**Total**
Number of cases	18326	59696	97583	182771	**358376**
Avg length	2471.67	1545.98	1114.71	1812.33	-
Median length	1940	1262	672	1413	-

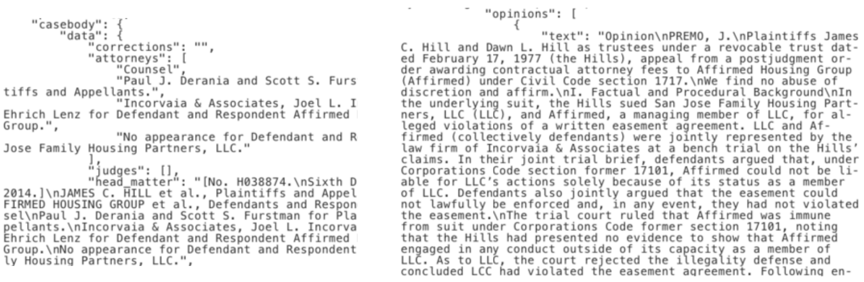
Table 2. Number of cases per word count in CASELAW4

Case length	New Mexico	Arkansas	North Carolina	Illinois	**Total**
< 200	952	2225	28439	29466	61082
200−500	738	5725	13037	12290	31790
500−1000	2466	14628	19103	25928	62125
1000−2000	5288	21862	20362	52144	99656
2000−5000	7036	14230	14312	53333	88911
> 5000	1846	1026	2330	9610	14812

(REVERSE). On the other hand, longer case reports usually indicate that the decision of the judges is non-binary (MIXED) and includes multiple sub-orders, or that more complex legal reasoning is involved.

The data in CASELAW4 are stored in JSON format. In addition to the original metadata about the case name, date, court, judges, cases cited etc., we annotated *a subset* of the cases with the AFFIRM or REVERSE outcome label (see Section 4). Finally, 500 cases from the New Mexico Court of Appeals are manually annotated with the AFFIRM, REVERSE, or MIXED outcome label as well as with the outcome sentences (also in Section 4). The dataset is publicly available on GitHub.[5]

Figure 1. Example of a case from CASELAW4



```
    "casebody": {
        "data": {
            "corrections": "",
            "attorneys": [
                "Counsel",
                "Paul J. Derania and Scott S. Furs
tiffs and Appellants.",
                "Incorvaia & Associates, Joel L. I
Ehrich Lenz for Defendant and Respondent Affirmed
Group.",
                "No appearance for Defendant and R
Jose Family Housing Partners, LLC."
            ],
            "judges": [],
            "head_matter": "[No. H038874.\nSixth D
2014.]\nJAMES C. HILL et al., Plaintiffs and Appel
FIRMED HOUSING GROUP et al., Defendants and Respon
sel\nPaul J. Derania and Scott S. Furstman for Pla
pellants.\nIncorvaia & Associates, Joel L. Incorva
Ehrich Lenz for Defendant and Respondent Affirmed
Group.\nNo appearance for Defendant and Respondent
ly Housing Partners, LLC.",
```

```
            "opinions": [
                {
                    "text": "Opinion\nPREMO, J.\nPlaintiffs James
C. Hill and Dawn L. Hill as trustees under a revocable trust dat-
ed February 17, 1977 (the Hills), appeal from a postjudgment or-
der awarding contractual attorney fees to Affirmed Housing Group
(Affirmed) under Civil Code section 1717.\nWe find no abuse of
discretion and affirm.\nI. Factual and Procedural Background\nIn
the underlying suit, the Hills sued San Jose Family Housing Part-
ners, LLC (LLC), and Affirmed, a managing member of LLC, for al-
leged violations of a written easement agreement. LLC and Af-
firmed (collectively defendants) were jointly represented by the
law firm of Incorvaia & Associates at a bench trial on the Hills’
claims. In their joint trial brief, defendants argued that, under
Corporations Code section former 17101, Affirmed could not be li-
able for LLC’s actions solely because of its status as a member
of LLC. Defendants also jointly argued that the easement could
not lawfully be enforced and, in any event, they had not violated
the easement.\nThe trial court ruled that Affirmed was immune
from suit under Corporations Code former section 17101, noting
that the Hills had presented no evidence to show that Affirmed
engaged in any conduct outside of its capacity as a member of
LLC. As to LLC, the court rejected the illegality defense and
concluded LCC had violated the easement agreement. Following en-
```

4. Outcome Extraction

The first step towards outcome prediction is to extract outcome labels from case reports. Unfortunately, the original CAP dataset does not formally store the outcome in the case metadata; the outcome is only mentioned in the text of the hearing. Therefore, one needs

[5] https://github.com/chinmusique/outcome-prediction

to extract the outcomes from text manually or automatically. In this section, we outline the methodology for automatic outcome extraction, explain how sentences containing the outcome can affect subsequent outcome prediction, and delve into the details of how the annotated parts of CASELAW4 were achieved.

4.1. Manual Outcome Annotation

As the data in CAP do not contain any outcome labels, we are faced with the so-called "cold start" problem: to train a model that extracts outcomes from case reports, one needs to get some labeled data first. For this reason, we randomly selected 500 cases from the New Mexico Court of Appeals and manually annotated them with one outcome label. In total, we collected 240 AFFIRM cases (among which 12 are dismissed cases), 159 REVERSE cases, and 101 MIXED cases.

In addition to case-level labels, we annotated each case at the sentence level, identifying sentences that contain the outcome information (e.g., *Judgment is affirmed* or *We affirm in part and reverse in part*). Such outcome sentences usually appear in the summary and conclusion sections of a report, but may as well appear in the main body of the case text. Outcome sentences are needed for two reasons: on the one hand, at extraction time, pre-filtering outcome sentences leads to more accurate outcome extraction in our setting (more on it below); on the other hand, at prediction time, it is important to remove explicit mentions of the outcome from the case report, so that the results are not biased.

The annotation process was performed using the web-based annotator system from Cognitiv+ [11].[6] All annotations are made available on GitHub[7].

4.2. Outcome Extraction Methodology

We split the process of extracting the outcome from a case into two steps. First, we select all the sentences in the case report that contain the outcome description (e.g., *The chancellor's order for alimony will be continued until final decree is entered on remand of the cause. In other respects the decree will be affirmed.*), then we decide on the final outcome based on the pre-filtered sentences only (e.g., AFFIRM). The first step uses a deep learning model, while the second step uses simple keyword matching. The choice for such architecture is motivated by the following.

- We could not use a deep learning model to accurately extract outcomes from the full case report (as opposed to outcome sentences only), since there is simply not enough training data: 500 annotated cases are too few to train all the weights and parameters of a complex RNN.

- We could not perform simple keyword matching on the full texts either: since the primary purpose of outcome extraction is to label cases before outcome prediction, the labels must be sufficiently accurate, so as not to propagate the annotation error further to the predictor. As discussed in Section 4.4, the keywords and patterns need to be quite generic, so that we account for different writing styles in multiple jurisdictions, and those vary a lot. If we use a set of more specific patterns (e.g., *we accordingly affirm* or *defendant's conviction is reversed*), a lot of outcomes are omitted. While experiment-

[6] https://cognitivplus.com/graybox

[7] https://github.com/chinmusique/outcome-prediction

	Precision	Recall	F-measure
OUTCOME	97.90	95.89	96.89
NON-OUTCOME	96.49	98.21	97.35
Total	97.15	97.13	97.13

Table 3. Performance of the sentence classification model (%)

ing with sets of phrase-based patterns, we were unable to reach the F-measure higher than 81.32%, due to low recall. Conversely, as we reduced the set of patterns towards the outcome keywords *affirm*, *reverse*, *remand*, and *dismiss*, the percentage of outcomes matched by the patterns increased. However, the precision drops: keyword patterns tend to also match legal facts and reasoning, such as in *In Ark. S&L Bd. v. Grant Cty. S&L, supra, the issue was not presented and we affirmed*, or *It is contended by appellant that the judgment should be reversed*, or *Under Rule 6(c), this court shall not affirm or revert a case based on an abbreviated record*. Simple keyword matching over full reports would have inferred that the outcomes for the above examples are AFFIRM, REVERSE, and MIXED, respectively, although it is not the case.
The two-step procedure of first selecting the outcome sentences and then inferring the outcome aims to balance the precision and recall of outcome extraction, while reaching a near perfect annotation quality (see Section 4.3).

- Finally, we are unable to use complex statistical models in the second step of the extraction process for an already familiar reason: 500 annotated cases are still not enough to train an accurate deep learning model.

4.3. Sentence Classification

In order to develop a sentence-level classifier that identifies whether a given sentence contains the outcome information, we split the 500 annotated cases into individual sentences and labeled all non-outcome sentences with the NON-OUTCOME class. For example, in the following excerpt from a CAP case report, the first sentence is non-outcome, while the second sentence mentions the outcome:
*The sole question raised on appeal is whether the district court erred in determining that Defendant was subject to being sentenced as a fourth-time DWI offender instead of a third-time offender (*NON-OUTCOME*). For the reasons discussed herein, we affirm the district court's judgment and sentence (*OUTCOME*).*

In total, we got 92k sentences, including 1455 outcome sentences and 90.8k non-outcome sentences. We then re-balanced the dataset by limiting the NON-OUTCOME class to 1455 randomly selected sentences with the corresponding label; the final sentence-level dataset consisted of 2910 sentences.

We formulated the task of identifying the outcome sentences as a binary classification problem, split the sentences into training, validation, and test sets by the 8:1:1 ratio, and trained a series of bi-LSTM models. The hyperparameters were chosen from embedding size $\{200, 300, 2000\}$, input size $\{100, 300\}$, and hidden layer dimensions $\{50, 100, 128\}$. The top performing classification model is a bi-LSTM with a single hidden layer of size 50 that uses Adam optimiser [3] and has the following parameters: learning rate 0.001, embedding size 200, input size 100, and 10 epochs. It achieves an F-measure of 97.13%. Performance details of the sentence classification model are outlined in Table 3.

All experiments were implemented in PyTorch and performed on a MacBook Air laptop with macOS 10.14, 1.6 GHz Intel Core i5 processor, and 16 GB 2133 MHz LPDDR3 memory.

4.4. Outcome Extraction

Once the outcome sentences are extracted, we apply simple keyword-based patterns to identify the final outcome contained in the sentences, since it does not make sense to use data-hungry deep learning models on such a small sample of hand-annotated data. The patterns are straightforward and function as follows: if the pre-filtered sentences contain a token *affirm* or *dismiss*, the outcome is AFFIRM; if they contain a token *reverse*, the outcome is REVERSE; if both *affirm/dismiss* and *reverse* are present, the outcome is MIXED.

The above patterns prove to work extremely well, once the outcome sentences are filtered out correctly (although they are not able to work on their own, as they would not differentiate between outcomes like *Judgment affirmed on all accounts* and recitals of previous decisions of the appeal like *Judgment affirmed by the previous court ruling*; see Section 4.2). The sentence classification model is easily trained on data coming from the same jurisdiction. However, the precision and recall drop when we transfer the model to cases from other jurisdictions. Most mistakes in annotation stem from the fact that different jurisdictions use different wordings and writing styles to record the same thing. This might involve the out-of-vocabulary problem: New Mexico judges do not typically use phrases like *motion allowed* or *petition denied* to pinpoint the outcome. Errors might as well stem from grammatical variability: while in our training set, most outcomes are expressed through constructs like *we affirm/reverse* and not through *order will be affirmed/reversed*, the LSTM model did not have enough training data to generalize beyond the writing style of one jurisdiction, i.e., New Mexico. As a result, AFFIRM and REVERSE cases are not recognized, but are automatically assigned to the MIXED category, and we were not able to achieve an F-measure higher than 60% in our empirical evaluation.

Since our outcome extraction procedure is used primarily for the purpose of annotating large volumes of cases from CASELAW4, the accuracy of outcome extraction must be the highest possible, and 60% is not enough. Therefore, we augmented the sentence classification model with one additional step, also pattern-based. The idea is simple: to help deep learning generalize across jurisdictions in the absence of enough labeled data, we pair its predictions with unambiguous patterns that univocally signal the final outcome but might not have yet been captured by the model. We can easily come up with these patterns from domain expertise. If such a sentence-level pattern is matched, the sentence is labeled with the respective outcome disregarding the statistically predicted label. The sentence-level patterns that we used are: *The trial/district court's order/judgment/decision/conviction is affirmed/reversed*, *The order/judgment/decision/conviction of the district/trial court is affirmed/reversed*, *We affirm the order/judgment/decision/conviction of the district/trial court*, and *Affirmed/Reversed/Dismissed/Error/No error*.

We validated the hybrid outcome extraction model by manually checking the labels of 100 randomly selected cases, 25 per jurisdiction. The weighted average F-measure of the outcome extraction model is 82.32%. Tables 4 and 5 outline the detailed results

	Precision	Recall	F-measure
Affirm	93.18	78.85	85.42
Reverse	100.00	80.77	89.36
Mixed	54.29	86.36	66.67
Total	86.40	81.00	82.32

Table 4. Performance of the outcome extraction model (%)

		Predicted label		
		Affirm	Reverse	Mixed
True label	Affirm	41	0	11
	Reverse	0	21	5
	Mixed	3	0	19

Table 5. Confusion matrix

Outcome type	New Mexico	Arkansas	North Carolina	Illinois	**Total**
Affirm	8707	33202	44022	85706	171637
Reverse	4961	14912	16694	47933	84500
Not annotated	4658	11582	36867	49132	102239

Table 6. Number of cases per outcome type

of model validation and the confusion matrix, respectively. They demonstrate that the single most important source of errors is the Affirm cases that are classified as Mixed, which in turn affects the overall performance of the model. While the average precision of 86.4% would be sub-optimal for large-scale outcome extraction and case annotation, if we only focus on the Affirm and Reverse classes, the weighted average precision will be 95.45%. This is the reason why our final annotations only contain Affirm and Reverse, which we consider reliable.

4.5. *Automated* Caselaw4 *Annotation*

Finally, we used the outcome extraction model to annotate cases in Caselaw4. Since we aim for reliable, high-quality annotations, and precision is much more important than recall, we only keep the predicted labels Affirm and Reverse, and we leave unlabeled the cases for which the predicted outcome is Mixed. In total, the number of labeled classes in Caselaw4 are 171637 for Affirm and 8450 for Reverse; 102239 cases are left without outcome annotation. Table 6 outlines the distribution of outcome types per jurisdiction.

4.6. *Lessons Learned*

Outcome extraction from cases of appeal proves to be a non-trivial task. While at first glance it seems that the ways outcomes are manifested in text are quite repetitive and pattern-like (*a judgment/order/conviction/sentence is affirmed/reversed/dismissed*), there is no one straightforward way to extract outcomes automatically with high quality, for two reasons. Patterns may work well on a coherent, homogeneous set of cases, i.e., those coming from the same court. However, the language in general and the outcome sentences in particular vary a lot across courts, judges, and jurisdictions. This variability may be captured with patterns or statistical models—but for that, considerable amounts of cases from diverse sources need to be analyzed and annotated manually. The labelled data bottleneck is one of the reasons why legal outcome prediction for English language is not yet as developed as the one for Chinese language [10]. The current work aims to remedy this problem with a combination of pattern- and deep learning-based approaches, as well as to open the discussion about the value of structured legal data.

5. Summary and Outlook

This paper presents the baseline for extracting legal outcomes from the US Case Law. The main contributions of this work are the annotated dataset of English language court cases with the outcomes explicitly stated in the metadata, as well as the baseline model for outcome extraction for state Courts of Appeal cases using the Caselaw Access Project data. The new dataset CASELAW4 contains both automatic and manual annotations, and acts as the first step towards outcome prediction and advanced legal analytics for the English language legal documents, and for US state Courts of Appeal in particular.

Additionally, the work provides valuable insights into the problem of automatic annotation of legal cases. In the absence of large numbers of hand-annotated data, high-quality information extraction such as outcome extraction requires a combination of statistical learning and pattern matching. While deep learning models can typically generalize patterns appearing in texts, in the setting of labeled data deficiency, they work best when they (a) are paired with keyword- and phrase-based patterns, and (b) "mimic" keyword matching by utilizing few parameters and a small encoding size. The intuition behind it is to make them learn outcome patterns quicker. This way, the models are still versatile and are able to account for linguistic ambiguity and variability, while learning the key outcome features from little data.

The current work can be advanced in several directions. Firstly, the CASELAW4 dataset could be used in a number of prediction models, from more complex LSTMs to transformers to pre-trained language models. Since the problem of legal outcome prediction is a highly complex problem that relies on numerous factors, sophisticated deep learning models show promising results [12,19,25]. Secondly, it is important to further improve outcome extraction, to go beyond the binary system of AFFIRM and REVERSE labels and to move to more granular MIXED cases. Lastly, it is essential to further improve the outcome extraction quality by handling the linguistic variance in writing styles across courts and jurisdictions .

Acknowledgments. This work was supported by the Industrial Strategy Challenge Fund's (ISCF) Next Generation Services Research Programme and UK Research and Innovation (UKRI), through the project "Unlocking the Potential of AI for English Law".

References

[1] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov;9(8):1735-80.

[2] Wang C, Yang H, Bartz C, Meinel C. Image captioning with deep bidirectional LSTMs. In Proceedings of the 24th ACM International Conference on Multimedia 2016 Oct (pp. 988-997).

[3] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980. 2014.

[4] Ashley KD. Artificial intelligence and legal analytics: New tools for law practice in the digital age. Cambridge University Press; 2017 Jul 10.

[5] Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. In Proceedings of the 19th International Conference on Artificial Intelligence and Soft Computing 2017 Jun (pp. 553-562).

[6] Liang Q, Wu P, Huang C. An efficient method for text classification task. In Proceedings of the 2019 International Conference on Big Data Engineering 2019 Jun 11 (pp. 92-97).

[7] Lim CG, Choi HJ. LSTM-based model for extracting temporal relations from Korean text. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing 2018 (pp. 666-668).

[8] The President and Fellows of Harvard University. Caselaw Access Project. 2018, `https://case.law`.

[9] Spaeth HJ, Epstein L, Martin AD, Segal JA, Ruger TJ, Benesh SC. 2016 Supreme Court Database, Version 2016 Legacy Release v01. (SCDB_Legacy_01) http://supremecourtdatabase.org.

[10] Xiao C, Zhong H, Guo Z, Tu C, Liu Z, Sun M, Feng Y, Han X, Hu Z, Wang H, Xu J. CAIL2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478. 2018 Jul 4.

[11] Cognitiv+ blog. Cognitiv+ is offering our NLP annotator free of charge to all Covid-19 researchers. 2020, https://cognitivplus.com/cognitiv-offers-nlp-annotator-free-of-charge-to-all-research-projects-fighting-to-find-the-cure-against-covid-19.

[12] Chalkidis I, Androutsopoulos I, Aletras N. Neural legal judgment prediction in English. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019 Jul (pp. 4317-4323).

[13] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016.

[14] Chalkidis I, Fergadiotis E, Malakasiotis P, Androutsopoulos I. Large-scale multi-label text classification on EU Legislation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019 Jul (pp. 6314-6322).

[15] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. 2020 Jun;28(2):237-66.

[16] Medvedeva M, Vols M, Wieling M. Judicial decisions of the European Court of Human Rights: Looking into the crystal ball. In Proceedings of the Conference on Empirical Legal Studies 2018.

[17] Katz DM, Bommarito MJ, Blackman J. A general approach for predicting the behavior of the Supreme Court of the United States. PloS one. 2017 Apr;12(4):e0174698.

[18] Şulea OM, Zampieri M, Vela M, van Genabith J. Predicting the law area and decisions of French Supreme Court cases. In Proceedings of the International Conference Recent Advances in Natural Language Processing, 2017 Sep (pp. 716-722).

[19] Lim C. An evaluation of machine learning approaches to Natural Language Processing for legal text classification [master thesis]. Imperial College London; 2019.

[20] Zhong H, Guo Z, Tu C, Xiao C, Liu Z, Sun M. Legal judgment prediction via topological learning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing 2018 (pp. 3540-3549).

[21] Yang W, Jia W, Zhou X, Luo Y. Legal judgment prediction via multi-perspective bi-feedback network. In Proceedings of the 28th International Joint Conference on Artificial Intelligence 2019 Aug 10 (pp. 4085-4091).

[22] Luo B, Feng Y, Xu J, Zhang X, Zhao D. Learning to predict charges for criminal cases with legal basis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing 2017 Sep (pp. 2727-2736).

[23] Ye H, Jiang X, Luo Z, Chao W. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 2018 Jun (pp. 1854-1864).

[24] Hu Z, Li X, Tu C, Liu Z, Sun M. Few-shot charge prediction with discriminative legal attributes. In Proceedings of the 27th International Conference on Computational Linguistics 2018 (pp. 487-498).

[25] Chen H, Cai D, Dai W, Dai Z, Ding Y. Charge-based prison term prediction with deep gating network. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing 2019 Nov (pp. 6363-6368).

[26] Li C, Sheng Y, Ge J, Luo B. Apply event extraction techniques to the judicial field. In Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers 2019 Sep 9 (pp. 492-497).

[27] Šavelka J, Ashley KD. Extracting case law sentences for argumentation about the meaning of statutory terms. In Proceedings of the Third Workshop on Argument Mining 2016 Aug (pp. 50-59).

[28] Copus R, Hübert R. Detecting Inconsistency in Governance. Working Paper, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2812914.

[29] Vacek T, Teo R, Song D, Nugent T, Cowling C, Schilder F. Litigation Analytics: Case outcomes extracted from US federal court dockets. In Proceedings of the Natural Legal Language Processing Workshop 2019 Jun 7 (pp. 45-54).

[30] Li Q, Zhang Q, Yao J, Zhang Y. Event extraction for criminal legal text. In Proceedings of the 2020 IEEE International Conference on Knowledge Graph 2020 Aug (pp. 573-580).

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200858

Legal Knowledge Extraction for Knowledge Graph Based Question-Answering

Francesco Sovrano [a], Monica Palmirani [b] and Fabio Vitali [a]
[a] *DISI, University of Bologna*
[b] *CIRSFID-Alma AI, University of Bologna*

Abstract. This paper presents the Open Knowledge Extraction (OKE) tools combined with natural language analysis of the sentence in order to enrich the semantic of the legal knowledge extracted from legal text. In particular the use case is on international private law with specific regard to the Rome I Regulation EC 593/2008, Rome II Regulation EC 864/2007, and Brussels I bis Regulation EU 1215/2012. A Knowledge Graph (KG) is built using OKE and Natural Language Processing (NLP) methods jointly with the main ontology design patterns defined for the legal domain (e.g., event, time, role, agent, right, obligations, jurisdiction). Using critical questions, underlined by legal experts in the domain, we have built a question answering tool capable to support the information retrieval and to answer to these queries. The system should help the legal expert to retrieve the relevant legal information connected with topics, concepts, entities, normative references in order to integrate his/her searching activities.

Keywords. Legal Knowledge Extraction; Question-Answering; Ontology Design Pattern Alignment.

1. Introduction and Problem Statement

The legal ontology modelling method [4, 2] is a relevant instrument for defining the legal concepts and relationships included in legal texts (e.g., hard law, judgment, soft law, etc.) but it is extremely expensive, it depends to the hermeneutic approach adopted by each scholar or community (e.g., common law vs. civil law), it is influenced by a strong localization due to the local jurisdiction (e.g., domestic regulation and local court action), by the cultural and social norms (e.g., concept of gender) and furthermore every time there is a new modification in the legal framework (e.g., new legislation) a refinement or (even worse) a whole extension of the ontology is required. On the other hand, the semantic web techniques are very useful in legal domain to detect relevant texts according to situations and concrete cases, or to filter the connected legislation among wide corpora. Legal ontologies are also important, in the legal rule modelling, for providing a common vocabulary of predicates and thus to permit interoperability between different legal knowledge engineers belonging to different institutions. Other important applications of legal ontologies are legal design and smart contract domains in order to refer to the same legal concepts and axioms inside of a community. For these reasons a hybrid solution is necessary in order to take benefits from the legal ontology information, but with a reasonable balance between information granularity and human effort. A light ontology, based on the language analysis of the text and on simple relationships between classes, is a very poor instrument for the legal domain, where the legal norms often include exceptions and odd situations (e.g., derogation, retroactivity, suspension). In fact, the core of a domain legal ontology is very detailed and accurate but it takes years for an accurate modeling and during this period too much modifications can change the legal scenario. In the meantime the Ontology Design Patterns (ODPs) [8, 13] method could help to maintain a good methodological connection between light and foundational approaches [6] (bottom-up and top-down approaches). In our work we show an effective way to combine the two approaches by mapping to Ontology Design Pattern (ODP) a Knowledge Graph (KG) automatically extracted for performing Question Answering (QA). The results are provided in the form of a Knowledge Graph (KG) of templates aligned to well-known legal ODPs, and structured in a way that would enhance the selection of relevant material for a specific case-law or legal situation. The aforementioned KG is extracted from regulations such as: Rome I Regulation EC 593/2008, Rome II Regulation EC 864/2007, and Brussels I bis Regulation EU 1215/2012. The alignment of the KG to the legal ODPs, plus the fact that the KG is extracted from regulations, makes the KG a sort of light ontology. This light ontology will be structured in a way that would be possible for the legal end-users (e.g., lawyer, judge, scholar, students) to easily query and explore the extracted information through a QA algorithm. In the future we intend to integrate our approach in existing tools [1] for legal document analysis, thus allowing to the legal experts to formulate relevant queries in order to orient the KG during the modelling phase and to correct/disambiguate some edges/nodes of the extracted KG.

2. Background Information

2.1. International Private Law

The International Private Law (PIL) is a complex legal domain that presents frequent conflicting norms between the hierarchy of legal sources (e.g., national vs. European level), between legal domains (e.g., consumer law vs. labor law), between the procedures adopted (e.g., criminal law vs. civil law). After the Treaty of Amsterdam (1 May 1999), the legislative powers for judicial cooperation in civil and commercial matters were transferred to EU institutions with the aim to harmonize the following issues: i) which state court has jurisdiction in private matters having cross-border implications; ii) which domestic law is applicable in such matters, iii) and under which conditions can a foreign decision be recognized and enforced in another Member State. Scientific research on PIL reveals the need to create a bridge between European and national laws on this domain, accessing heterogeneous legal sources. The European project Interlex [2] intended to investigate this domain and to use technology to fill the gap between different legal sources. The need to fill such a gap between legal sources is so frequent in the PIL domain that the legal experts need to recall all the norms and to combine them using the theory of interpretation principles and the case-law based approach. For this reason, the classical

[1] https://interlex-portal.eu/FindLaw/Doc/LegalAct/6573821

[2] http://www.interlexproject.eu/index.html

databases and information systems based on full-text search or document classification or document clustering seem not to be effective. This is because the terminologies are different (e.g., “consumer” in consumer law, and “data subject” in data protection law), the normative references are consistent only inside the same domain, the mapping of concepts is difficult because they are not perfectly equivalent. For this reason our approach is to discover correlation between different terms and parts of the legal documents, and to use the ODP main classes for structuring a KG that can be queried by the experts.

2.2. *Legal Ontology Design Patterns*

In the legal domain, different researches [10, 1] identified some basic ontology design patterns regularly used for modelling norms. i) Agent-role-time [3]; ii) Event-time-place-jurisdiction [4] ; iii) Agent-action-time [7]; iv) Object-document [12]; v) Legal deontic ontology [5][10]. These patterns, combined with linguistic taxonomies, could provide a good solution for creating a bridge between the variants of the legal definitions and the conceptualization level [11].

3. Proposed Solution

We can extract Knowledge Graphs (KGs) from legal documents by exploiting the grammatical dependencies of their content, through an automated dependency parser. In order to make sense of the extracted information, making these KGs useful for exploration and Question Answering (QA), we should be able to guarantee some properties that would facilitate the interoperability of the KG with state-of-the-art deep-learning based QA algorithms. Considering that modern state-of-the-art QA algorithms have several limitations in terms of input and output size, the challenge is not trivial. Furthermore these QA algorithms are trained with natural language and not Resource Description Framework (RDF) triples.

Assuming that serialising natural language into RDF triples is a challenging open-problem, the simplest solution appears to be to abandon RDF serialisation in favour of natural language, but natural language is not as structured as RDF and performing QA over large natural language corpora is too expensive. This is why our proposed solution consists in ad-hoc KG extraction of triples in the form of textual templates rather than classical RDF, in order to preserve the natural language while structuring it into a proper graph aligned to external resources such as WordNet or ODPs. We think that effective abstract querying can be possible by structuring the KG as an ontology, giving it a solid backbone in the form of a taxonomy. In fact, being able to identify the type/class of a concept would allow to perform queries with a reasonable level of abstraction, making possible to refer to all the sub-types (or to some super-types) of a concept without explicitly mentioning them.

Our proposed solution, for extracting and making sense of complex information stored into natural language documents, is defined by the following steps: [i)] KG ex-

[3] https://sparontologies.github.io/pro/current/pro.html

[4] https://sparontologies.github.io/tvc/current/tvc.html

traction, Taxonomy Construction, Ontology Design Pattern Alignment, KG question answering. [5]

3.1. KG extraction

KG extraction is the extraction of concepts and their relations, from a natural language text, in the form of a graph where concepts are nodes and relations are edges. As mentioned before, we are looking for a way to extract KGs that somehow preserve the original natural language, preferring them over classical RDF graphs. This way we can easily make them inter-operate with deep-learning based QA algorithms and language models.

More in detail, we perform KG extraction by:

1. Analysing the grammatical dependencies of tokens extracted by Spacy's Dependency Parser, therefore identifying noun syntagms (concepts): the possible objects and subjects of the triples to extract.
2. Using the dependency tree to extract all the tokens connecting two different target concepts in a sentence, thus building a template composed by these connecting tokens (the order of the tokens is preserved) together with the target concepts (replaced with the placeholders "{subj}" and "{obj}", in accordance with their grammatical dependencies).
3. Creating a graph of triples where target concepts are subjects/objects and templates are predicates.

The resulting triples are a sort of function, where the predicate is the body and the object and the subject are the parameters. Obtaining a natural language representation of these template-triples is straightforward by design, by replacing the instances of the parameters in the body. An example of template-triple (in the form subject, predicate, object) is: "the applicable law", "Surprisingly {subj} is considered to be clearly more related to {obj} rather than to something else.", "that Member State".

Furthermore, to increase the interoperability of the extracted KG with external resources, we performed the following extra steps: i) We assigned a URI and a RDFS label to every node of the graph. The URI is obtained by lemmatising the label. ii) We added special triples to keep track of the snippets of text (a.k.a. the sources) from which the concepts and the relations are extracted. iii) We added sub-class relations between composite concepts (syntagms) and the simplest concepts (if any) composing the syntagm. Because of the adopted extraction procedure, the resulting KG is not perfect, thus it may contain some mistakes caused by wrong grammatical dependencies or other issues. But, due to the fact that the original natural language is practically preserved, one would expect that such imperfection would not significantly impact on QA if the adopted neural networks are robust enough (e.g. being trained on very large corpora of real text).

[5] The graph, the taxonomies and the ontological hinge, extracted from the 3 EU's regulations we mentioned, can be found here: https://github.com/Francesco-Sovrano/Legal-Knowledge-Extraction-for-Knowledge-Graph-Based-Question-Answering

3.2. Taxonomy Construction

In order to efficiently use, query and explore the extracted KG, we need to structure it in a proper way. We believe that effective abstract querying can be possible by structuring the KG as a light ontology, giving it a solid backbone in the form of a taxonomy. In fact, being able to identify the types/classes of a concept would allow to perform queries with a reasonable level of abstraction, making possible to refer to all the sub-types (or to some super-types) of a concept without explicitly mentioning them. The taxonomy construction phase consists in building one or more taxonomies, through Formal Concept Analysis (FCA).

In order to build a taxonomy via FCA, one simple approach consists in exploiting (as FCA's properties) the hypernyms relations of the concepts in the KG. We found that a naive way to extract such relations is through the alignment of the extracted KG to WordNet[6], via a Word-Sense Disambiguation algorithm. Applying FCA, to the hypernyms of the aligned Wordnet concepts, produces a forest of taxonomies. Every taxonomy in this forest is a cluster of concepts rooted into very abstract concepts that we can use as label/identifier for the respective taxonomies.

The results we obtained for the three EU's regulations are quite interesting. In fact FCA is able to identify very few concepts clusters (taxonomies), and these clusters resemble the same core concepts our domain experts previously (and independently) identified for the regulation under study: person, claim and contract. More in detail, the main clusters obtained through FCA are about:

- **Legal Documents**: a document that states some contractual relationship or grants some right.
- **Acts**: something that people do or cause to happen.
- **Organizations**: a group of people who work together.
- **Causal Agents**: any entity that produces an effect or is responsible for events or results.
- **States**: the way something is with respect to its main attributes.

3.3. Legal Ontology Design Pattern Alignment

With rich enough taxonomies we can improve the quality of the KG structure by aligning it to known legal Ontology Design Patterns (ODPs). We can perform this alignment easily, by manually mapping the roots of every taxonomy obtained via FCA (see Section 3.2) to relevant concepts of the design patterns we identified in section 2.2. This is feasible because the number of relevant concepts in the ODPs is very small (in the order of 10). The KG extraction is said to be a bottom-up approach (from concrete documents, to abstract ontologies), while the design of ontologies through patterns is said to be a top-down approach (from abstract legal concepts identified by experts, to their concretization in the legal documents under examination). The top-down approach is more complicated to accomplish, because it requires a domain expert. Furthermore, the level of abstraction required for top-down ontologies in legal domain may be challenging and

[6] We are aware that WordNet is not designed for the legal domain, but at this stage of the work we are less interested in extracting more formal knowledge (e.g. RDF graphs). A better alternative to WordNet might consist in a combination of [9] with other existing resources such as Eurovoc, IATE and BabelNet.

time-consuming even for the best legal experts. On the other hand, the bottom-up approach is much easier to automatise, but it is prone to mistakes and redundancy, often producing worse results with respect to the other approach.

This is why we propose to exploit the best of these two approaches by using a sort of *ontological hinge* that should be able to connect a bottom-up KG with top-down ODPs. In order to obtain this *ontological hinge*, we have to abstract new relations between the concepts of the ODPs and those of the extracted KG.

It appears that only a few of the concepts in the ODPs defined in Section 2.2 are reasonably useful to specialize into more concrete concepts: “pro:RoleInTime”, “foaf:Organization”, “ti:TimeInterval”, “InformationObject”, “Place”, “pwo:Workflow”. Surprisingly, we can see that every cluster obtained through FCA can be quickly mapped into one the aforementioned concepts. The fact that the concepts to align are only 6 allows us to perform the mapping manually, with ease. In our case, a sufficient mapping/hinge function would be:

- “Causal Agent” (employee, consumer, etc..) mapped to “pro:RoleInTime”.
- “Organization” mapped to “foaf:Organization”.
- “Time Period” mapped to “ti:TimeInterval”.
- “Written Communication” (legal document, etc..) and “Information” (database, etc..) mapped to “pro:InformationObject”.
- “Location” (country, region, address, etc..) mapped to “pro:Place”.
- “Action” (legalization, protest, litigation, etc..) mapped to “pwo:Action”.
- “Obligation” mapped to “pro:Obligation”.

3.4. Question Answering

KG-based question answering consists in answering natural language questions about information contained in the KG. Let C be the set of concepts in a question Q. We perform KG question answering by:

1. **Extracting**: extract C from Q, using the same procedure adopted for extracting concepts during the KG extraction in section 3.1.
2. **Matching**: find the most syntactically similar KG concepts to C, and retrieve all their related template-triples including those of the sub-classes of C.
3. **Selecting**: among the natural language representations of both the retrieved triples and their respective subjects/objects[7], select those snippets of text that are sufficiently likely to be an answer to Q.
4. **Answering**: return as set of answers the contexts (the source paragraphs) of the selected snippets of text (triples or simple concepts).

More in detail, the *matching* phase is performed by computing the similarity between the labels of every concept in the KG and every concept in C; we do it by using the algorithm described in [14]. Similarly to the *matching* phase, the *selection* phase is performed by means of a variation of [14], that combines Term Frequency–Inverse Document Frequency (TFIDF) with a version of the Universal Sentence Encoder (USE) for

[7] Some questions can be succinctly answered through a single concept, while others require a more elaborated sentence (therefore a template-triple).

QA [16]. The main difference of the *selection* phase with the *matching* phase is that the similarity is computed between the questions and the contextualized triples/concepts in the KG. Every triple/concepts is represented in natural language (as in the matching phase) and its context is the snippet of text (the paragraph) from which the template has been originally extracted.

4. Related Work

In literature we found many works on Question Answering (QA), only few of them [18, 17, 3] were on Knowledge Graphs (KGs) and all of these were about RDF or similar technologies. As comparison to our work, we point to the many state-of-the-art deep-learning based QA algorithms implemented by Wolf et al. [15]. With these algorithms, using the whole Rome II Regulation EC 864/2007 as input context would require an impractical amount of time[8] for every posed question, in order to obtain very short (e.g. 2-3 words) answers which quality heavily depends on the selected linguistic model. The practical advantage of our approach over the others is that it is capable of selecting the most relevant text fragments in the context, limiting the search for an answer to very few paragraphs rather than the entire corpus. Furthermore the matching criterion [14] we adopted for answer selection combines both statistical and deep learning approaches trying to take the best from both, making the answering process a little bit more transparent.

5. Evaluation

We are interested in evaluating the usefulness of the resulting Knowledge Graph (KG), extracted from contract regulations, with respect to the legal user's needs. The goal in this specific domain is to extract knowledge according to specific situations and to detect the useful legal sources capable to help the expert to interpret them and to find a solution. A user can interact with the KG through the Question Answering (QA) tool, posing natural language questions and expecting useful answers from the system. Some frequently asked questions, in these cases, might be related to where a legal trial is celebrated (e.g., the pertinent jurisdiction and court), because there are many nuances and conflicting rules depending to the typology of actors, the country of residence (e.g., habitual residence), the country where the activity is performed (e.g.,country where the employee habitually carries out his work). For this evaluation we focus our attention on the jurisdiction and the judge *a quo* [9]. The adopted methodology comprises a team of legal experts selecting 8 relevant questions and evaluating the correctness of the answers provided by our algorithm. As shown in table 1, the resulting answers are in some cases imprecise, but overall the algorithm we described in this paper achieved an average top5-recall[10] of 34.91%. Considering we have been using generic QA models (not fine-tuned on the specific domain) and a naive approach to paragraph matching (see Section 3.4;

[8] Try it: `https://huggingface.co/models?filter=question-answering`

[9] The judge *a quo* is the judge that is pertinent in the first grade of judgment and consequently the definition of this element is related with the jurisdiction applicable and the normative system valid

[10] Let n be the number of strictly-correct produced answers, let $|E|$ the number of expected answers for a question, then the top5-recall is $\frac{n}{min(|E|,5)}$.

Table 1. Questions, expected answers and the top5 produced answers and their top5-recall - "B" stands for Brussels, "RI" for Rome I and "RII" for Rome II. "Rec." stands for Recital and "Art." for Article. The percentages shown are an estimate of the answer pertinence (the answer similarity defined in Section 3.4).

Question	Produced Answers - Top5	Expected Answers	Top5-Recall
Who determines disputes under a contract?	RI: Rec. 12 (63.31%) B: Art. 17.2 (36.70%) RI: Rec. 24 (35.39%)	B: Art. 7.1, 8.3, 8.4, 17	25%
What factors should be taken into account for conferring the jurisdiction to determine disputes under a contract?	RI: Rec. 12 (65.71%) B: Art. 25 (39.84%) B: Art. 17 (35.89%) B: Art. 25.5 (36.82%) B: Rec. 15 (36.15%)	B: Art. 7.1, 17, 20, 25	50%
Which parties of a contract should be protected by conflict-of-law rules?	RI: Rec. 23 (53.53%) B: Rec. 18 (43.06%) RI: Rec. 24 (36.97%) RI: Art. 25.1 (36.42%) RI: Rec. 27 (36.28%)	RI: Rec. 23 RI: Art. 6, 8, 13	25%
In which case claims are so closely connected that it would be better to treat them together in order to avoid irreconcilable judgments?	B: Art. 8.1 (47.75%)	B: Art. 8, 30, 34	33.3%
What kind of agreement between parties are regulated by these regulations?	B: Art. 73.3 (45.51%) B: Rec. 12 (43.90%) B: Rec. 36 (42.83%) B: Art. 71.2.a (38.51%) B: Art. 71.1 (38.03%)	B: Rec. 6, 10, 12 B: Art. 1 RI: Rec. 7 RI: Art. 1	20%
In which court is celebrated the trial in case the employer is domiciled in a Member State?	B: Art. 21.1.a (68.67%) B: Art. 22.1 (62.17%) B: Art. 21.2 (56.25%) B: Art. 21.1.b.i (44.90%) B: Art. 20.2 (44.07%)	B: Art. 21, 22, 23	66%
How should a contract be interpreted according to this regulation?	RI: Art. 10.1 (39.77%) RI: Rec. 17 (35.02%)	RI: Rec. 22, 12, 26, 29 RI: Art. 12	0%
Which law is applicable to a non-contractual obligation?	RI: Art. 8.1 (54.21%) RII: Art. 15.g (51.84%) RII: Art. 16 (50.06%) RII: Art. 8.1 (49.33%) RII: Rec. 22 (48.43%)	RII: Rec. 17, 18, 26, 27, 31 RII: Art. 4-20	60%

it could be improved by integrating information coming from an external reasoner), we believe the results are promising.

As we can see the QA system is able to identify plausible answers for all the questions, even if they are clearly limited to the knowledge explicitly mentioned in the regulations. In many real-case scenarios we need to codify also implicit rules that are coming from the legal experts, in order to include also non-written relationships coming from the theory of law. In any case this approach compared with pure a full-text method is producing better results. Despite this, the results show that the QA algorithm is poor in reasoning (especially multi-hop reasoning), being trained to solve tasks related to commonsense, hence pointing to future developments.

6. Conclusions

This paper presents a hybrid and innovative approach to model legal knowledge extracted from heterogeneous legal sources, using ontology design patterns as skeleton for mapping the information deducted using OKE and linguistic NLP analysis. In a legal domain, with multiple conflicting norms and a large number of multiple definitions for the same concept, our proposed approach gives interesting results, providing a KG where the legal expert can easily retrieve the relevant information via critical queries. The KG provides a useful instrument for information navigation, that could be integrated in traditional information systems and legal databases. The confidence scores of the preliminary results are not optimal, but in the light of conflicting norms this approach could be an interesting outcome, in any case, because it integrates the legal interpretation methodology provided by the legal experts. We definitely need more testing with the help of the legal experts in order to tune the resulting pipeline defined in Section 3, but we believe that the proposed approach is correct especially in the domain where there is no unique accredited interpretation and the application of the norms depends too much on the hierarchy of sources. In the future we intend to integrate our approach in existing tools[11] for legal document analysis, as the first part of a sophisticated explanatory tool for making sense of complex legal documents, facilitating the process of representing legal knowledge in machine-compatible ways (e.g. ontologies, taxonomies, thesauri, etc..).

Acknowledgements

This paper is was conducted thanks the InterLex project funded by the European Union's Justice programme under Grant Agreement No.800839 and with the contribution of CIRSFID-Alma AI and DISI University of Bologna funds. The evaluation test was conducted by Giorgia Bincoletto, Salvatore Sapienza, Biagio Distefano of the Law, Science and Technology International PhD.

References

[1] Guido Boella, Luigi Di Caro, and Valentina Leone. Semi-automatic knowledge population in a legal document management system. *Artificial Intelligence and Law*, 27(2):227–251, 2019.

[2] Pompeu Casanovas, Monica Palmirani, Silvio Peroni, Tom Van Engers, and Fabio Vitali. Semantic web for the legal domain: the next step. *Semantic Web*, 7(3): 213–227, 2016.

[11] https://interlex-portal.eu/FindLaw/Doc/LegalAct/6573821

[3] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: learning question answering over qa corpora and knowledge bases. *arXiv preprint arXiv:1903.02419*, 2019.

[4] Meritxell Fernández-Barrera and Giovanni Sartor. The legal theory perspective: doctrinal conceptual systems vs. computational ontologies. In *Approaches to Legal Ontologies*, pages 15–47. Springer, 2011.

[5] Fabien Gandon, Guido Governatori, and Serena Villata. Normative requirements as linked data. In *Legal Knowledge and Information Systems: JURIX 2017. The Thirtieth Annual Conference*. IOS Press, 2017.

[6] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer, 2002.

[7] Aldo Gangemi, Silvio Peroni, David Shotton, and Fabio Vitali. The publishing workflow ontology (pwo). *Semantic Web*, 8(5):703–718, 2017.

[8] Pascal Hitzler, Aldo Gangemi, and Krzysztof Janowicz. *Ontology engineering with ontology design patterns: foundations and applications. Studies on the semantic web*, volume 25. IOS Press, 2016.

[9] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. Glossbert: Bert for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*, 2019.

[10] Monica Palmirani, Michele Martoni, Arianna Rossi, Cesare Bartolini, and Livio Robaldo. Pronto: Privacy ontology for legal compliance. In *Proc. 18th Eur. Conf. Digital Government (ECDG)*, pages 142–151, 2018.

[11] Monica Palmirani, Giorgia Bincoletto, Valentina Leone, Salvatore Sapienza, and Francesco Sovrano. Pronto ontology refinement through open knowledge extraction. In *JURIX*, pages 205–210, 2019.

[12] Silvio Peroni, Monica Palmirani, and Fabio Vitali. Undo: The united nations system document ontology. In *International Semantic Web Conference*, pages 175–183. Springer, 2017.

[13] Valentina Presutti, Giorgia Lodi, Andrea Nuzzolese, Aldo Gangemi, Silvio Peroni, and Luigi Asprino. The role of ontology design patterns in linked data projects. In *International Conference on Conceptual Modeling*, pages 113–121. Springer, 2016.

[14] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. Deep learning based multi-label text classification of unga resolutions. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance (ICEGOV2020)*, 2020.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[16] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*, 2019.

[17] Weiguo Zheng, Hong Cheng, Jeffrey Xu Yu, Lei Zou, and Kangfei Zhao. Interactive natural language question answering over knowledge graphs. *Information Sciences*, 481:141–159, 2019.

[18] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over rdf: a graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 313–324, 2014.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200859

Natural Language Processing Applications in Case-Law Text Publishing

Francesco TARASCONI [a,1], Milad BOTROS [a], Matteo CASERIO [a],
Gianpiero SPORTELLI [a], Giuseppe GIACALONE [b], Carlotta UTTINI [b],
Luca VIGNATI [b] and Fabrizio ZANETTA [b]

[a] *CELI Language Technology, Turin, Italy*
[b] *Giuffrè Francis Lefebvre, Milan, Italy*

Abstract. Processing case-law contents for electronic publishing purposes is a time-consuming activity that encompasses several sub-tasks and usually involves adding annotations to the original text. On the other hand, recent trends in Artificial Intelligence and Natural Language Processing enable the automatic and efficient analysis of big textual data. In this paper we present our Machine Learning solution to three specific business problems, regularly met by a real world Italian publisher in their day-to-day work: recognition of legal references in text spans, new content ranking by relevance, and text classification according to a given tree of topics. Different approaches based on BERT language model were experimented with, together with alternatives, typically based on Bag-of-Words. The optimal solution, deployed in a controlled production environment, was in two out of three cases based on fine-tuned BERT (for the extraction of legal references and text classification), while, in the case of relevance ranking, a Random Forest model, with hand-crafted features, was preferred. We will conclude by discussing the concrete impact, as perceived by the publisher, of the developed prototypes.

Keywords. natural language processing, applications, transfer learning, language models, text classification, information extraction, publishing industry, machine learning, BERT fine-tuning, random forest, Italian language

1. Introduction

Processing case-law contents, such as court judgements, for electronic publishing purposes is a time-consuming activity that encompasses several sub-tasks and usually involves adding annotations to the original text. Some operations, such as ranking new documents by their relevance, are required to determine which ones are worthy of publication. Other annotations are incorporated in products or services for the final customers, for example to facilitate search and exploration of related contents. Annotating legal texts requires specific knowledge, usually provided by domain experts or coded in a software component. On the other hand, recent trends in Artificial Intelligence and Natural Language Processing enable the automatic and efficient analysis of big textual data. These methods usually must be adapted for a specific domain. We will present our solution to three different business problems in the context of an Italian publisher of legal texts

[1] Corresponding Author: Francesco Tarasconi, CELI Language Technology, Via San Quintino 31, 10121 Turin, Italy; E-mail: tarasconi@celi.it.

and related products, in particular concerning the automatic annotation of Italian court judgements (mostly common or criminal law), originally provided in XML format :

1. **recognizing legal references**, distinguishing between references to legislation or to other judgements (Section 4);
2. **ranking by potential relevance** for the editors, to help assessing whether the content should be published or not (Section 5);
3. **labeling according to given topics**, described by a hierarchy of three classification levels, containing nodes such as ”personal freedom” or ”extortion” (Section 6).

For each problem, we developed a Machine Learning prototype that was deemed viable by the Business (i.e. the publisher's managers and decision-makers), and successfully deployed in a controlled production environment for inference on new data and further fine-tuning. The availability of high-quality training data, collected by the Business over the course of the years, enabled the successful experimentation of supervised methods. Before describing in detail the developed prototypes, we will summarize some previous work to better contextualize our research (Section 2); we will also provide essential details about the pre-existing annotation process of the publisher (Section 3). We conclude by discussing the business impact of the developed prototypes, together with their limitations and further work (Section 7).

2. Related work

A problem we will investigate in Section 4 is the automatic extraction of legal references, which can been solved without the help of Machine Learning through top-down approaches, as shown in [1] and [2]. However, our goal is also to classify different types of references according to their roles in the examined judgement (see [3] for a similar business case); we will frame the problem as a Named Entity Recognition one and solve it with Machine Learning methods, in order to better use context information and generalize. Named Entity Recognition for Italian language using Deep Learning is tackled with interesting results in [4]. Similar applications in role classification, that involve a Machine Learning approach, can be found in [5]. Text classification methods are within the scope of our research in Section 5 (binary classification problem) and Section 6 (multi-class); they have been successfully applied to a number of use cases ranging from plagiarism [6] to estimating the period in which a text was published [7]. Overall, Machine Learning overcomes the limits of manually compiling classification rules, when enough training data are available. Successful experiments in predicting law areas from text, using the Support Vector Machine model class, are described in [8]. Deep Learning approaches for the legal domain, using Convolutional Neural Networks, are described in [9]. More context to the problem of Extreme multi-label text classification (XMTC) and relative applications of Deep Learning techniques is provided in [10]. A larger amount of training examples was traditionally required in order to reach satisfying results through Deep Learning. Human-labeled data, domain-specific, are still necessary to conduct successful experiments, but in smaller amounts, thanks to transfer learning and pre-trained language models. One the most effective architectures developed over the last few years is Google BERT [11], a transformer model that leverages upon the self-attention mechanism. BERT can be fine-tuned for specific tasks such as Named Entity Recognition and text classification. Chalkidis and Kampas [12] noted that self-attention does not only lead to performance improvements in legal text classification, but might also provide useful evidence for the predictions. However, Deep Learning models can be computationally expensive and sometimes the apparent performance gain over other Machine Learning methods is negligible or spurious, as discussed for example in [13]. NLP–based metadata extraction for Italian legal texts is described in [14] and [15], but they are focused on the legislative act life-cycle and consolidation.

3. Business Context: A Real-World Publishing Process of Legal Texts

We will briefly describe in this section the business context where our research took place, in particular the electronic publishing workflow where NLP was applied. We won't provide information on other operations that are outside the scope of these applications. The original contents are judgements released by Italian courts and, after a pre-publishing phase, provided in XML format (*documents*). Each document is assigned a unique *ID* and stored in a database with its metadata, such as an identifier of the corresponding source, called *Authority*, and the *Date* of the judgement. XML documents are divided in three different sections: an introductory *Preamble* providing contextual information to the judgement; a main part containing factual and legal information (called *FactsLaw*); a final part containing the verdict (called *PQM*, acronym for the Italian expression "per questi motivi", meaning "for these reasons"). Each section is further divided into *Paragraphs*, of variable length (from hundreds to thousands of characters).
The following Steps are performed on each document, enriching the original XML:

1. **extraction of legal references**: contiguous spans within the same Paragraph, that contain a reference, are tagged. Prior to this work, it was accomplished through top-down rules and regular expressions. See Section 4 for more details;
2. **linking of legal references**: hyperlinks to external documents are added, containing the judgement or legislation mentioned in the text. This is accomplished through a custom search engine that is outside the scope of this paper;
3. **relevance classification**: documents are labeled as relevant or irrelevant. Relevant ones are considered for further editing and publication. This operation is historically performed by domain experts and content curators. See Section 5 for more details;
4. **topic classification**: each relevant document is labeled by domain experts, according to what the examined judgement is about and a pre-existing topic tree. See Section 6;
5. **holding formulation**: one or more holdings are compiled by domain experts, summarizing the law principles expressed in the judgement. Through adoption of attention-based models, this task is related to the topic classification one step and briefly discussed in Section 6.
6. **reference classification**: references to other judgements that were previously extracted are classified by domain experts as "according to" / "different from" / "related to", based on the relation between the two verdicts; errors in reference extraction are also manually corrected.

Topics, holdings and legal references form the backbone of several of the publisher's electronic products, for attorneys and other Law professionals. Given the current state-of-the-art, outlined in Section 2, A.I. potential and limitations, the following best practices were agreed upon with the Business:

(i) to carefully frame the use cases/business problems;
(ii) to identify meaningful datasets for Machine Learning model development, together with the appropriate error metrics;
(iii) to evaluate different models according to chosen metrics, and also in terms of computational cost and explainability, so that an informed decision can be taken by the Business;

(iv) to perform error analysis of each prototype, educating the Business on the limits of A.I. and understanding where the human must intervene.

4. Application: Recognition of Legal References

Our goal is to identify in a judgement all the spans of text that refer to a specific law or to another judgement. References to other judgements must also be classified as ”according to” / ”different from” / ”related to” the examined judgement. Developing a single Machine Learning system that performs both operations allows to automate Steps 1 and 6 described in Section 3. This simple distinction between reference roles is used downstream in several publishing products.

4.1. Methodology

The proposed solution is based on a fine-tuned version of multi-language BERT[2] for Named Entity Recognition [11]. Our setup is similar to the one for Portuguese language described in [16], but we do not use the CRF layer that is described in the paper. The final layer performs token-level classification with one predicted class among the following target list, defined in manner consistent with common IOB practices in NER [4]:

1. O: the token is outside / not part of a reference;
2. B-L: the token is the beginning of a legislative reference e.g. to a specific law article;
3. B-J-ACC: the token is the beginning of a reference to a judgement, that is in accordance with the examined judgement;
4. B-J-DIF: as B-J-ACC, but the referred verdict was different from the examined one;
5. B-J-REL: as B-J-ACC, but the two judgements are simply related; from a legal standpoint, it’s a weaker relation compared to B-J-ACC and B-J-DIF;
6. I-R: the token is the continuation of a reference (any kind).

The chosen metrics to evaluate the system, agreed upon with the Business, are the F1-Scores of ”proper” reference classes, excluding the O class from the list above.
Original input comes in the form of XML Paragraphs where free text references (i.e. spans of text) are tagged accordingly. Through a custom version of the standard BERT *wordpiece* tokenizer, a preprocessing phase prepares each Paragraph for analysis, associating target classes to BERT tokens, and removing all XML markup. Data are split in a Training Set (70%), Development Set (15%) and Test Set (15%). BERT fine-tuning is conducted by adding a final feedforward layer with softmax, and minimizing cross-entropy loss function over training data. Development data are used to perform model evaluation and selection by maximizing the weighted average of F1-Scores, calculated over all target classes, barring the O class. A postprocessing function, used for integration with the publisher’s pipeline, is made available for re-aligning BERT output to the original text. At the moment of inference on new documents, all Paragraphs are classified separately, in conformity with model training.

Implementation Details. The described methodology was implemented using TensorFlow 1.12, in particular the *estimator* API for training, evaluation, prediction and export for serving [17].

[2] BERT original code from: https://tfhub.dev/google/bert_multi_cased_L-12_H-768_A-12/1

4.2. Prototype Data

When our research started, the publisher's information concerning the type of reference to other judgements (necessary to discriminate between B-J-ACC, B-J-DIF, B-J-REL classes) was not available at the level of text spans, but stored only at document level. Therefore, domain experts were involved to further annotate, add the precise classes to text spans, and provide the required input. For this reason, only a small subset of the publisher's documents could be used, for the development of this application. We worked on criminal and common law judgements of the Italian Highest Courts of Appeal. The resulting dataset is composed of 6,133 Paragraphs from 150 documents, with 13,657 total references.

4.3. Results

Table 1. Breakdown of Test error metrics for fine-tuned BERT model in legal reference recognition.

Type	Test Cases	Precision	Recall	F1-Score
B-L	692	0.940	0.957	0.948
B-J-ACC	77	0.535	0.494	0.514
B-J-DIF	15	0.200	1.000	0.333
B-J-REL	776	0.883	0.930	0.906
I-R	16,118	0.969	0.985	0.977

Breakdown of performance on Test Set is reported in **Table 1**. The system achieved a weighted F1-Score on classes of interest of 0.970 (including continuations I-R), 0.900 (counting only beginnings of references B).

Error Analysis. Several errors were in delimiting text spans containing references, exactly as the original data, but the model proposals were found to be often acceptable as well. Only in 6 cases serious errors were committed: confusing laws with judgements, or B-J-ACC references with B-J-DIF. Despite lower performances on less frequent classes, the prototype was considered viable by the Business, given also the partially subjective nature of the task; more experiments will be conducted with additional data.

Other Experiments. Different setups, for solving the problem with BERT, were experimented with, such as breaking down the problem into related subtasks (e.g. distinguishing B-L and B-J, plus distinguishing between B-J-ACC, B-J-DIF and B-J-REL). These approaches yielded slightly lower performances (between 0.01 and 0.02 drop in weighted F1-Score) and found more difficult to correctly assign the less frequent labels. Other experiments, without pre-training for the Italian language (e.g. analyzing windows of texts as shown in [4]), saw a larger performance drop, especially in discriminating between B-J-ACC, B-J-DIF and B-J-REL.

5. Application: Ranking by Relevance

The goal of this application is to identify the potential relevance of documents, in order to select the ones that will be annotated further and eventually published (see Step 3 in Section 3). A model that formulates such predictions should implement, explicitly or implicitly, the criteria employed by humans; a supervised approach, based upon preclassifed relevant documents, seems therefore promising . Because the output of Machine Learning models can usually be expressed as a probability or a score, our idea, agreed upon with the Business, was to provide the end-user with a ranking of documents, to review model suggestions in order of relevance.

5.1. Methodology

Our solution is based on a Random Forest model [18] that uses hand-crafted features, defined together with the editors, and is trained on a binary classification problem, to distinguish between relevant and irrelevant documents. The probability of belonging to the relevant class is provided as output and it's used as relevance ranking. The features are:

a) number of references to legislation (see Section 4) in the document;
b) number of references to other judgements (see Section 4) in the document;
c) length (number of characters) of FactsLaw XML section (see Section 3), after removing XML markup;
d) number of legal quotes, delimited by quotation marks and containing more than one word;
e) binary features corresponding to presence or absence or specific expressions in the PQM XML section.

Coding these features involves an NLP preprocessing step, not only to remove XML markup, but also to perform lemmatization and be able to match variants of the original expression, e.g. ”**declares** the **appeals** inadmissible” should match the given expression ”**declare** the **appeal** inadmissible”.
Data are split in a Training Set (60%), Development Set (20%) and Test Set (20%). A grid search is performed in order to maximize the weighted F1-Score on the development set and identify the optimal number of estimators, minimum samples in each leaf and maximum depth of each tree. According to the importance of listed variables in the resulting model, calculated through permutations [18], they are all useful to the task.

Implementation Details. The procedure was coded in Python and implemented using Scikit Learn 0.22.1 [19].

5.2. Prototype Data

The dataset, that was determined in accordance with the Business, represents a sample of stored data from all the Authorities which are currently managed. The dataset is composed of 4,958 documents: 64% relevant and 36% irrelevant. It is largely composed (70%) of judgements from the Highest Courts of Appeal (criminal and common law), but also contains documents from the T.A.R. Administrative Regional Tribunal (5%), Italian Constitutional Court (4%) and E.U. courts (4%). Remaining documents come from other Italian courts. Irrelevant documents are likely to be *more frequent* in the real-world execution of this task, as not all the historical ones were stored and available. At the same time, it was not possible to determine an average distribution of ”relevant vs irrelevant” documents. This fact will be considered in analyzing the performance of the optimal solution; strong bias towards the relevant class should be avoided.
Finally, working on this dataset, through Machine Learning methods, allowed us to find human mistakes in the original classification.

5.3. Results

Table 2. Breakdown of Test error metrics for Random Forest model in relevance classification.

Class	Precision	Recall	F1-Score
Relevant	0.84	0.90	0.87
Irrelevant	0.81	0.70	0.75
Weighted average	0.83	0.83	0.83

The model achieves a weighted F1-Score of 0.83 in Test. Breakdown between relevant and irrelevant classes is reported in **Table** 2.
As we have seen in the *Prototype Data* Subsection, irrelevant data are likely underrepresented in our dataset, so it's important that the performance on the irrelevant class is checked carefully, as its weight in the real-world application is higher. We will evaluate further fine-tuning of the model and re-balancing of the training data, as information from the production environment is collected.

Error Analysis. Human analysis of 50 errors showed that, in 64% of cases (32 documents), the model picked the wrong class, but in a borderline situation; several irrelevant documents were considered ”acceptable” (i.e. relevant) by some of the domain experts. The remaining 18 documents, actual mistakes, had a lower ranking associated with them, indicating lower model confidence. There were cases, difficult to treat with this approach, where a judgement was labeled as ”irrelevant”, because the annotator knew pertained a topic, well covered by the publisher, and with very similar judgements already analyzed.

Other Experiments. A single Classification Tree, based upon the same features, achieved a weighted F1-Score of 0.78 on the same task. Adding features, based on frequent words or specific references, found in the document, didn't improve the performance of Random Forest or Classification Tree models.
An implementation of BERT for binary classification of judgements, similar to the one described in Section 6, was used to test an approach entirely based on free text analysis, and achieved a weighted F1-Score of 0.75.

6. Application: Classification by Topic

Our goal is to label each document as related to none, one or more topics. Topics belong to a proprietary resource of the publisher's: a classification tree of three levels, with 12,066 nodes. The majority of documents (75%) are associated to a single topic; more than 99% documents possess between 1 and 5 labels.
After conducting an exploratory analysis, the original problem was transformed in a more tractable one; for what concerns the prototype, object of this research, target topics must possess a minimum number $F = 200$ of training examples. In case a node is discarded because of its frequency, lower than F, documents belonging to that node are assigned to the parent node (corresponding to a more generic topic), if possible.
This restriction allowed us to build a working prototype and show its usefulness to the Business. Adding data, reducing F and managing more topics will be treated as further evolution of the developed system.

6.1. Methodology

The proposed solution is based on a fine-tuned version of multi-language BERT [11] for multi-label text classification. Our setup is similar to the one proposed in [20] for multi-label text classification on EU Legislation and we exploit the multi-label attention mechanism through an architecture similar to the one described in [21]. The main obstacle in adapting BERT to this application is the limitation of the length of documents that the model can analyze (512 tokens). We fix a constant N, and, for each document, N different Paragraphs are randomly sampled from the FactsLaw XML section and processed individually through the attention layers. The N different outputs from these layers are combined to produce a unified document representation, passed to the final fully connected (and output) layer. Random sampling is more effective, on this dataset, than considering the first N Paragraphs. Data are split in a Training Set (80%) and a Test Set (20%). Fine-tuning is conducted on Training data, by minimizing sigmoid cross-entropy loss function.
Output is provided in two formats: all labels with score > 0.5 or the top K labels, regardless of their minimum probability. While the first format is used to evaluate and compare different models through F1-Scores and their weighted average, the second format is used in production environment for end-users (domain experts and editors), when performing inference on new data.

Implementation Details. The described approach was implemented in the same framework employed in Section 4, using TensorFlow 1.12. N was fixed at 40 for computational reasons. K was fixed at 5 after evaluating the prototype's performance.

6.2. Prototype Data

The dataset is composed of 44,413 documents from the Highest Courts of Appeal (Criminal and Common Law), collected by the publisher over the last five years.
After a preliminary analysis, having fixed F at 200, 81 topics were considered during development. In spite of considering a small subset of the full classification tree, 64% of documents have at least one valid (i.e. frequent) topic associated. The most frequent topic is *contracts and obligations*, with 1,248 examples.

6.3. Results

The described solution achieves a weighted F1-Score of 0.505 over the 81 examined Topics. It was verified that the correct (i.e. originally assigned by human) labels are found 90% of the times in the first 5 predictions.
The output of attention layers, as suggested in [12], is currently being examined by domain experts to assess its usefulness in highlighting the most important Paragraphs and in the holding definition phase (Step 5 of Section 3).

Error Analysis. Examining the top K predictions for some documents, domain experts verified that they are usually related and that there was in fact a certain degree of freedom in choosing the original classification itself.

Other Experiments. The best performing Bag-of-Words, no pre-training, experiment, was an XGBoost ensemble model [22], using a combination of frequent words and frequent legislation references as features. It achieved a weighted F1-Score of 0.370.

7. Conclusions and Future Work

We have first introduced the annotation process of court judgements by a real-world Italian publisher, highlighting areas where amount of human effort and availability of training data motivated the experimentation of Machine Learning automatic approaches. We then described the developed solutions to three specific problems, showing how Natural Language Processing could in fact reach satisfying performances where training data was sufficient. Employing a model architecture based on BERT, fine-tuned for the specific tasks of Named Entity Recognition and Extreme Multi-label Text Classification, provided the best results in the most complex problems, where free text understanding was crucial. In the case of ranking by relevance, the importance of hand-crafted features (in capturing the differences between relevant and irrelevant documents) explains why a simpler, faster Random Forest model obtained better results and was chosen for deployment.

7.1. Business Impact

Working on the described prototypes required several skills, ranging from Natural Language Processing development to in-depth knowledge of the legal domain for problem framing, data selection and error analysis. The resulting team-mix was deemed successful and can be adopted in new projects. Communication between the Business and the developers was constant during the research and effective: the added value of Deep Learning was shared and understood, not taken for granted. The developed prototypes are performing inference on a subset of new real-world data, in a controlled production environment, before further fine-tuning and integration. The current integration model is asynchronous and employs Apache Kafka (`kafka.apache.org`) for handling data feeds. Each Machine Learning module is exposed as a synchronous RESTful Service. A JSON data exchange format was agreed for integration in the rest of the publishing pipeline. This system currently helps the editors and reduces the amount of human effort by pre-annotating documents which can then be reviewed more quickly by the domain expert. The model for relevance ranking mirrors closely human decision-making and actually allows to correct some mistakes in the original classification.

7.2. Limits and Further Developments

The models for extracting legal references and topic classification will require new cycles of annotated data gathering, training and test, in order to increment the coverage of less frequent classes. Instead, the main limit of ranking by relevance is its being based upon intrinsic features of the documents. Adding features based on the similarity to previous judgements could help in dealing with particular or difficult cases.
Once the users have acquired trust in the system and the machine behavior mirrors more closely the human's in edge cases, a deeper integration in the publishing process will be possible. To this end, advances in zero-shot learning should also be followed closely and tested. Finally, monitoring how these modules work on new data and carefully reviewing user's feedback will help in identifying unknown issues and making the solution more robust over time.

References

[1] Agnoloni T, Bacci L, Peruginelli G, van Opijnen M, van den Oever J, Palmirani M, Cervone L, Bujor O, Lecuona AA, García AB, Di Caro L, Siragusa S. Linking European Case Law: BO-ECLI Parser, an Open Framework for the Automatic Extraction of Legal Links. In: Frontiers in Artificial Intelligence and Applications. Volume 302: Legal Knowledge and Information Systems. 2017. Pages 113-118.

[2] Gheewala A, Turner C, Maistre JR. Extraction of Legal Citations using Natural Language Processing. In: Proceedings of the 15th International Conference on Web Information Systems and Technologies. 2019. Pages 202-209.

[3] Winkels R, Boer A, de Maat E, van Engers T, Breebaart M, Melger H. Constructing a semantic network for legal content. In: Gardner, A (ed.) Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL 2005). ACM Press, New York (2005). Pages 125-140.

[4] Bonadiman D, Severyn A, Moschitti A. Deep Neural Networks for Named Entity Recognition in Italian. Italian Conference on Computational Linguistics (CLiC it). 2015.

[5] Winkels R, Boer A, Vredebregt B, Someren, A. Towards a Legal Recommender System. JURIX. 2014.

[6] Barrón-Cedeño A, Vila M, Martí MA, Rosso P. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. In: Computational Linguistics 39, 4 (2013). Pages 917–947.

[7] Niculae V, Zampieri M, Dinu L, Ciobanu AM. Temporal Text Ranking and Automatic Dating of Texts. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers. Association for Computational Linguistics. 2014. Pages 17–21.

[8] Șulea OM, Zampieri M, Malmasi S, Vela M, Dinu LP, van Genabith J. Exploring the Use of Text Classification in the Legal Domain. In: Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL). London, United Kingdom. 2017.

[9] Wei F, Qin H, Ye S, Zhao H. Empirical Study of Deep Learning for Text Classification in Legal document Review. 2018 IEEE International Conference on Big Data (Big Data). Seattle, WA, USA. 2018. Pages 3317-3320.

[10] Liu J, Chang WC, Wu Y, Yang Y. Deep Learning for Extreme Multi-label Text Classification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17). Association for Computing Machinery, New York, NY, USA. 2017. Pages: 115–124.

[11] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. Association for Computational Linguistics. 2019. Pages 4171–4186.

[12] Chalkidis I, Kampas D. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. In: Artificial Intelligence and Law 27. 2019. Pages 171–198.

[13] Niven T, Kao HY. Probing Neural Network Comprehension of Natural Language Arguments. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. Pages 4658-4664.

[14] Bolioli A, Dini L, Mercatali P, Romano F. For the Automated Mark-Up of Italian Legislative Texts in XML. Legal Knowledge and Information Systems. JURIX. 2002.

[15] Spinosa P, Giardiello G, Cherubini M, Marchi S, Venturi G, Montemagni S. NLP-based metadata extraction for legal text consolidation. In: The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference. 2009. Pages 40-49.

[16] Souza F, Nogueira R, Lotufo R. Portuguese Named Entity Recognition using BERT-CRF. Preprint on arxiv.org. Last revised 27 Feb 2020.

[17] Abadi M et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. White Paper. Software available from tensorflow.org.

[18] Breiman, L. Random Forests. In: Machine Learning 45. 2001. Pages 5–32.

[19] Pedregosa F et al. Scikit-learn: Machine Learning in Python, JMLR 12. 2011. Pages 2825-2830.

[20] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Ion Androutsopoulos Large-Scale Multi-Label Text Classification on EU Legislation In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. Pages: 6314–6322.

[21] You R, Zhang Z, Wang Z, Dai S, Mamitsuka H, Zhu S. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada. 2019.

[22] Friedman J. Greedy Function Approximation: A Gradient Boosting Machine. In: The Annals of Statistics, Vol. 29, No. 5. 2001.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200860

Sentence Embeddings and High-Speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents

Hannes WESTERMANN [a,1], Jaromír ŠAVELKA [b], Vern R. WALKER [c], Kevin D. ASHLEY [d] and Karim BENYEKHLEF [a]

[a] *Cyberjustice Laboratory, Faculté de droit, Université de Montréal*
[b] *School of Computer Science, Carnegie Mellon University*
[c] *LLT Lab, Maurice A. Deane School of Law, Hofstra University*
[d] *School of Computing and Information, University of Pittsburgh*

Abstract. Human-performed annotation of sentences in legal documents is an important prerequisite to many machine learning based systems supporting legal tasks. Typically, the annotation is done sequentially, sentence by sentence, which is often time consuming and, hence, expensive. In this paper, we introduce a proof-of-concept system for annotating sentences "laterally." The approach is based on the observation that sentences that are similar in meaning often have the same label in terms of a particular type system. We use this observation in allowing annotators to quickly view and annotate sentences that are semantically similar to a given sentence, across an entire corpus of documents. Here, we present the interface of the system and empirically evaluate the approach. The experiments show that lateral annotation has the potential to make the annotation process quicker and more consistent.

Keywords. Annotation, Language Models, Sentence Embeddings, Approximate Nearest Neighbour, Interactive Machine Learning

1. Introduction

A lot of AI & Law research is enabled by annotation of legal texts. The annotation can be performed on several levels of textual units, such as the entire document, the paragraph, or an individual sentence. In this work, we focus on annotations performed on the sentence level. AI & Law research has employed a variety of annotation schemes on the sentence level, such as the annotation of:

- the rhetorical roles sentences play in a legal case (such as factual circumstances, a legal rule or an application of a legal rule to factual circumstances);
- the presence or absence of a certain factual circumstance the sentence describes (such as whether a security measure was present in a trade-secret case);
- the type and attributes of contractual clauses (such as the kind of liability addressed in a certain clause); and

[1] Corresponding Author: Hannes Westermann, E-mail: hannes.westermann@umontreal.ca

- the relevance of a sentence retrieved from a legal case to interpret a statutory term.

An annotated corpus of documents has many useful applications. For instance, a classification algorithm may be trained to infer labels for new sentences in a larger corpus of documents. This may lead, for example, to insights about the distribution of clauses in a large data set of contracts, to improved predictions about the outcome of a legal case from factors, or to ranking documents by relevance to a particular search query.

Typically, annotation of documents is performed by one or several annotators using a tool that allows them to review one document at a time, and to sequentially assign a label for each sentence as it occurs in that document. This approach has significant drawbacks. First, it is inefficient because annotating large corpora in this way takes a long time and is expensive. The label has to be determined from scratch for each sentence, causing significant cognitive overhead. Second, the annotations might be inconsistent across similar sentences. Since annotators often work through thousands of sentences, they may not remember how a certain sentence type was annotated the last time they saw it. If multiple annotators are involved, this problem may be exacerbated, as similar sentences are reviewed by different annotators.

In this paper, we investigate an alternative approach which we call "lateral annotation." Similarly to the traditional approach, annotators use a tool to view documents and label sentences. However, given any sentence there is an option to see sentences across the entire corpus (or from the rest of the document) that are semantically similar to the focused sentence. This feature uses sentence encoders based on deep learning models and libraries to quickly deliver semantically similar sentences using approximate nearest neighbour searches. The annotator then has the option of reviewing these similar sentences and assigning labels to one or more of them. Although the computer system assists the user by showing similar sentences, the choice of how to label a sentence ultimately rests with the annotator. It is therefore a hybrid approach, using machine learning to support human annotators with their task.

Legal language is often formalized and uses recurring linguistic structures. This means that identical, or very similar, sentences often appear in many documents. For example, contract clauses specifying a certain type of liability might often use the same words, syntax and sentence structure. Lateral annotation makes use of this attribute of legal language by allowing the annotator to label all similar sentences at one time. This can increase the speed of annotation. Since all similar sentences can be labelled at once, the consistency of the annotations is also likely to increase. Consequently this approach can significantly ease the important task of labelling large data sets in the legal domain.

2. Related Work

Branting et al. [4,5] proposed a semi-supervised approach to annotation of case decisions. The approach is based on several observations about the consistency of language across separate cases and within different sections of the same case. The researchers annotated a small set of decisions and calculated the mean of the semantic vectors [2,17] of all the spans annotated by a given tag (the "tag centroid"). The annotations were then projected to semantically similar sentences in the entire corpus to enable explainable prediction. In our work, we describe a hybrid method where we show the semantically similar sentences to an annotator for rapid and reliable annotation.

A steady line of work in AI & Law focuses on making the annotation effort more effective. Westermann et al. [26] proposed and assessed a method for building strong, explainable classifiers in the form of Boolean search rules. Employing an intuitive interface, the user develops Boolean rules for matching instead of annotating the individual sentences. Here, we replace the Boolean matching rules with sentence semantic similarity. Instead of developing the rules, the user confirms that the semantically similar sentences should be labeled as instances of the same types. Šavelka and Ashley [21] evaluated the effectiveness of an approach where a user labels the documents by confirming (or correcting) the prediction of a ML algorithm (interactive approach). The application of active learning has been explored in the context of classification of statutory provisions [25] and eDiscovery [8,9]. Hogan et al. [13] proposed and evaluated a human-aided computer cognition framework for eDiscovery. Tools to retrieve and rank text fragment by similarity for coding have further been implemented in qualitative data analysis tools, such as QDA Miner[2] and Nvivio.[3]

3. Proposed Framework

We investigate a system that enables an annotator to perform lateral annotations on a corpus of documents. We use sentence embeddings to capture the meaning of sentences, and then use approximate nearest neighbour search to find sentences that are semantically similar to a source sentence. This enables us to provide the annotators with viable sentence candidates for annotation in sub-second time.

3.1. Sentence Embeddings

In order to search for similar sentences based on an original sentence, we need a way to store sentences in a vector format that makes comparison easy. There are several ways of representing sentences in this way.

A bag of words representation (e.g., TF-IDF) is a simple but effective way to encode the meaning of a sentence. It has, however, at least two notable disadvantages: an enormously large feature space and the inability to account for the relatedness of meaning in different words. This means that sentences with the same meaning may be deemed to be completely different if they use largely non-overlapping vocabulary (e.g., synonyms). This is problematic for applications where sentence similarity is a key component (as in this work). In our experiments, we include the representation as a strong baseline due to its simplicity and wide usage.

More recently, pre-trained word embeddings and language models have gained popularity in creating word embeddings. These representations are motivated by the so-called distributional hypothesis: words that are used and occur in the same contexts tend to have similar meanings. [12] The idea that "a word is characterized by the company it keeps" was popularized by Firth. [11] The gist of the method is that words with similar meanings are projected onto similar vectors, by analyzing massive corpora of text to learn the distributions. There are several ways of combining these word vectors to produce sentence vectors, of which we have chosen three:

[2] provalisresearch.com/products/qualitative-data-analysis-software

[3] www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

1. Cer et al. [6] use the transformer architecture [22] and Deep Averaging Network [15] trained on the SNLI dataset. We work with the implementation released by the authors as the Google Universal Sentence Encoder (**GUSE**).[4]
2. Reimers et al. [19] build on top of BERT [10] and RoBERTa [18], which have been shown to be remarkably effective on a number of NLP tasks. Specifically, they use siamese and triplet network structures to derive semantically meaningful sentence embeddings. The authors released the models as Sentence Transformers (**ST**). We use this implementation in our work.[5]
3. Conneau et al. [7] demonstrate the effectiveness of models trained on a natural language inference task (Stanford NLI dataset [3]). They propose a BiLSTM network with max pooling trained with fastText word embeddings [2,17] as the best universal sentence encoding method. We adapt the implementation released by the authors which is commonly referred to as **InferSent**.[6]

3.2. Efficient Similarity Search over High-dimensional Vectors

Document search traditionally relies on a combination of relational databases built on structured data (metadata search) and inverted indexes (full-text search). These cannot deal efficiently with the vectors that represent documents' meaning. The brute-force approaches to indexing (i.e., store the results for all documents) or querying (i.e., compare the query to each data point) do not scale well beyond fairly small data sets.

In order to achieve semantic similarity search with the desired properties, one has to tackle the problem of preprocessing a set of n data points $P = \{p_1, p_2 \ldots, p_n\}$ in some metric space $\mathbb{X}$ (e.g., the d-dimensional Euclidean space $\mathbb{R}^d$) so as to efficiently answer queries by finding the point in P closest to a query point $q \in \mathbb{X}$. Solutions to this well-studied problem in other domains can be readily applied in our context. [14] For example, images and videos have become a massive source of data for indexing and search. And since it is often not practical to manually annotate the data to enable the use of relational databases, a search in some sort of a vector space remains the only option.

Johnson et al. [16] proposed a system that allows efficient indexing and search over collections containing around 1 billion documents. This shows that the current state-of-the-art is capable of supporting virtually any practical scenario in a legal annotation domain. For example, Šavelka et al. [20] segmented the whole corpus of US case-law into 0.5 billion sentences. While the technique in [16] would allow for efficient semantic similarity search over such a collection, it is most likely several magnitudes larger than any realistic legal annotation task. We utilize the Annoy similarity search library released by Spotify[7] for its ease of use and minimal system requirements. Annoy is an efficient implementation of the Approximate Nearest Neighbors algorithm proposed in [14].

3.3. Lateral Annotation

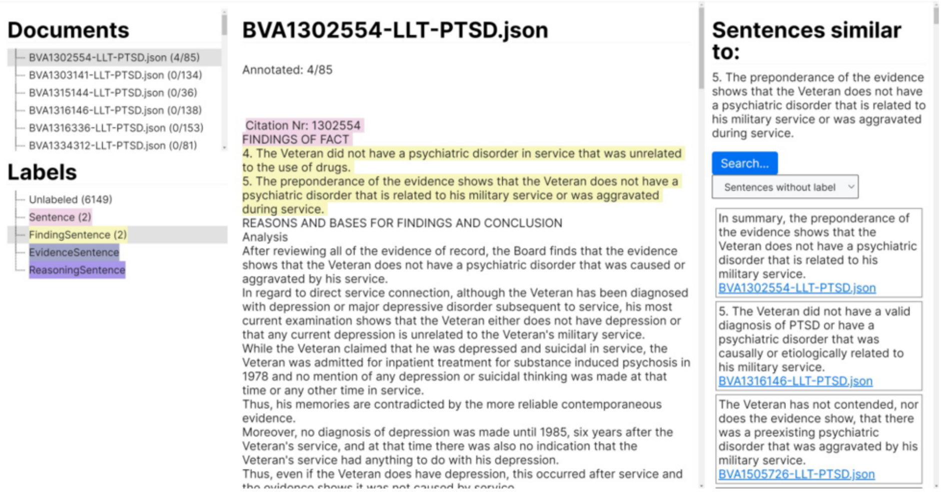
Semantic sentence embeddings and efficient vector similarity search are combined to enable lateral annotation. We have developed a prototype interface called CAESAR,

[4] https://tfhub.dev/google/universal-sentence-encoder/4

[5] github.com/UKPLab/sentence-transformers

[6] github.com/facebookresearch/InferSent

[7] github.com/spotify/annoy


Documents
BVA1302554-LLT-PTSD.json (4/85)
BVA1303141-LLT-PTSD.json (0/134)
BVA1315144-LLT-PTSD.json (0/36)
BVA1316146-LLT-PTSD.json (0/138)
BVA1316336-LLT-PTSD.json (0/153)
BVA1334312-LLT-PTSD.json (0/81)
Labels
Unlabeled (6149)
Sentence (2)
FindingSentence (2)
EvidenceSentence
ReasoningSentence
BVA1302554-LLT-PTSD.json
Annotated: 4/85
Sentences similar to:
Search...
Sentences without label

Citation Nr: 1302554
FINDINGS OF FACT
4. The Veteran did not have a psychiatric disorder in service that was unrelated to the use of drugs.
5. The preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service or was aggravated during service.
REASONS AND BASES FOR FINDINGS AND CONCLUSION
Analysis
After reviewing all of the evidence of record, the Board finds that the evidence shows that the Veteran does not have a psychiatric disorder that was caused by or aggravated by his service.
In regard to direct service connection, although the Veteran has been diagnosed with depression or major depressive disorder subsequent to service, his most current examination shows that the Veteran either does not have depression or that any current depression is unrelated to the Veteran's military service.
While the Veteran claimed that he was depressed and suicidal in service, the Veteran was admitted for inpatient treatment for substance induced psychosis in 1978 and no mention of any depression or suicidal thinking was made at that time or any other time in service.
Thus, his memories are contradicted by the more reliable contemporaneous evidence.
Moreover, no diagnosis of depression was made until 1985, six years after the Veteran's service, and at that time there was also no indication that the Veteran's service had anything to do with his depression.
Thus, even if the Veteran does have depression, this occurred after service and the evidence shows it was not caused by service

5. The preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service or was aggravated during service.

In summary, the preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service.
BVA1302554-LLT-PTSD.json

5. The Veteran did not have a valid diagnosis of PTSD or have a psychiatric disorder that was causally or etiologically related to his military service.
BVA1316146-LLT-PTSD.json

The Veteran has not contended, nor does the evidence show, that there was a preexisting psychiatric disorder that was aggravated by his military service.
BVA1505726-LLT-PTSD.json

Figure 1. A screenshot of the prototype Computer-Assisted Efficient Semantic Annotation & Ranking (CAESAR) Interface

Computer-Assisted Enhanced Semantic Annotation & Ranking, to demonstrate this capability. The sentence embedding frameworks are used to create semantic embeddings of all sentences in a corpus. These are then used to create an index for fast similarity search.

The capability is provided to the user through an annotation interface (see Figure 1). The interface allows the annotator to define a schema of labels in a hierarchical structure (i.e. a type system), and to tag individual sentences with these labels. For each sentence, it is possible to retrieve semantically similar sentences, using the methods described above. These are shown in the sidebar to the right, sorted by similarity in descending order. This panel allows the annotator to perform lateral annotations by quickly annotating the sentences shown, or to see the context of the shown sentences before annotation.

We envision the following procedure for labeling sentences using CAESAR. An annotator starts with the first document, and labels the first sentence. Then, he asks to be shown similar sentences in the sidebar. He then labels sentences in the sidebar until sentences are no longer similar enough to quickly allow the annotator to determine that they are of the same class. The annotator then returns to the full text of the case and labels the next sentence. As the annotator moves to the next documents, many of the sentences may already be labelled, and can be skipped.

This method of improving annotation efficiency is completely unsupervised. It can be implemented before having started any kind of annotation, by relying on the sophisticated neural models trained on huge datasets of general texts (e.g., news corpora, Wikipedia). Despite this, the method seems to perform very well on legal annotation tasks, as demonstrated in Section 4.

4. Evaluation

4.1. Datasets

In order to evaluate the lateral annotation method, we use three existing data sets:

1. Walker et al. [23] analyzed 50 fact-finding decisions issued by the U.S. Board of Veterans' Appeals (**BVA**) from 2013 through 2017, all arbitrarily selected cases dealing with claims by veterans for service-related post-traumatic stress disorder (PTSD). For each of the 50 BVA decisions in the PTSD dataset, the researchers extracted all sentences addressing the factual issues related to the claim for PTSD, or for a closely-related psychiatric disorder. These were tagged with the rhetorical roles the sentences play in the decision [24]. We conducted our experiments on this set of 6,153 sentences.[8]
2. Šavelka et al. [20] studied methods for retrieving useful sentences from court opinions that elaborate on the meaning of a vague statutory term. To support their experiments they queried the database of sentences from case law that mentioned three terms from different provisions of the U.S. Code. They manually classified the sentences in terms of four categories with respect to their usefulness for the interpretation of the corresponding statutory term. In [20] the goal was to rank the sentences with respect to their usefulness; here, we classify them into the four value categories (**StatInt**).[9]
3. Bhattacharya et al. [1] analyzed 50 opinions of the Supreme Court of India. The cases were sampled from 5 different domains in proportion to their frequencies (criminal, land and property, constitutional, labor and industrial, and intellectual property). From each of the 50 decisions the sentence boundaries were detected using an off-the-shelf general tool.[10] Then the 9,380 sentences were manually classified into one of the seven categories according to the rhetorical roles they play in a decision. Our experiments were conducted on this set of sentences (**IndSC**).[11]

4.2. Experiments

In order to evaluate the effectiveness of lateral annotation and compare different embedding methods, we use our system to retrieve the closest sentences to a query sentence, and investigate how many of them have the same label as the source sentence. Assuming that lateral annotation is more efficient than sequential annotation, the more retrieved sentences that have the same label, the more efficiently the annotator will be able to annotate the data set.

We report several metrics. First, we investigate the length of chains of annotations by traversing the retrieved sentences, from the most similar to the least, until we arrive at a label that does not match the label of the source sentence. We calculate the longest encountered chain (Max) and the average chain length (Avg) for each data set and em-

[8] Dataset available at `github.com/LLTLab/VetClaims-JSON`

[9] Dataset available at `github.com/jsavelka/statutory_interpretation`

[10] `spacy.io`

[11] Dataset available at `github.com/Law-AI/semantic-segmentation`

bedding method. Second, we determine how many of the top 20 closest sentences have the same label as the source sentence (P@20) - a measure of precision.

Third, we visualize the high-dimensional GUSE embeddings of all sentences in a dataset, reduced to 2 dimensions using a Principal Component Analysis. The colors in the resulting visualization correspond to the gold standard labels for the individual sentences. The most important feature of an embedding space for our purpose is that each sentence should be surrounded by multiple sentences with the same label, e.g. the same color.

4.3. Results

Table 1 presents the Max, Avg, and P@20 statistics for each data set and for each embedding method. Overall, the sentence embeddings seem to capture enough linguistic information to achieve improvement in all three metrics, without any training on the domain-specific data sets. The neural models perform much better than the random baseline, and perform better or equal to the TF-IDF baseline.

Looking at the individual data sets, it seems like the **Board of Veterans' Appeals data set** benefits significantly from lateral annotation. On average, 70% of the closest 20 sentences to each sentence have the same label, meaning an annotator can likely annotate these sentences laterally. This could offer a significant speed-up in the annotation of such a data set.

Looking at the individual labels in Table 2, the "Citation" label seems by far the easiest to annotate laterally. 94% of the top 20 closest sentences to a citation sentence are also citation sentences. This is likely due to the special tokens (such as parentheses, year numbers and special words such as "See") in these sentences. "Rule" also performs very well, which might be due to the same rule being cited in multiple cases. The embeddings capture these distinctions well, which can be seen in Figure 2, where sentences of the same type seem clearly concentrated in certain areas.

In the **Statutory Interpretation data set** the technique appears most suitable for the sentences labeled as "No value." This makes sense since these are mostly sentences that fully or partially quote or paraphrase a statutory provision. Hence, these sentences are often very similar to each other. The middle graph in Figure 2 confirms this observation. Three compact red clusters clearly correspond to the "No value" sentences associated with the three terms of interest. The sentences with the other three labels are somewhat more challenging. Yet, even for the more challenging categories, a significant amount of sentences could still be annotated laterally, as seen in Table 2.

The **Indian Supreme Court data set** is where lateral annotation gives the least advantage, with our models retrieving under 40% of matching sentences in the top 20 positions. On average, each sentence seems to be next to only 2.1-2.4 sentences of the same class in the embedding space. This can also be seen in the PCA visualization in figure 2. Unlike the other data sets, the sentence embeddings do not seem to result in clearly separate classes. The comparative difficulty of separating this data set might be explained by the fact that the sentences are selected from five different domains, and assigned seven labels—more than the other two data sets. The "Ratio" and "Facts" sections seem slightly easier to annotate in a lateral fashion, which might be due to a consistent structure or content of these sentences. It is surprising that the "Ratio" class has a high precision, while the "Ruling of lower court" has low precision, but this matches the findings of the authors in [1] for classification difficulty.

	Statistics								
	BVA			StatInt			IndSC		
	Max	Avg	P@20	Max	Avg	P@20	Max	Avg	P@20
Random	9	1.35	.24	18	2	.45	10	1.36	.24
TF-IDF	195	13.73	.59	197	24.40	.70	27	2.16	.36
GUSE	696	55.92	**.70**	257	25.80	**.73**	**45**	**2.43**	.37
ST	710	48.62	.68	**277**	**30.16**	.69	30	2.22	.35
InferSent	**863**	**83.92**	**.70**	204	22.5	.66	**45**	2.41	**.38**

Table 1. Statistics for different sentence embedding methods, including evaluation of chains of lateral annotation (Max, Avg) and how many of the 20 closest sentences on average have the same label (P@20).

BVA		SID		ISC	
Label	P@20	Label	P@20	Label	P@20
Sentence	0.60	No Value	0.89	Facts	0.40
Finding	0.49	Potential	0.58	Ruling (lower)	0.08
Evidence	0.76	Certain	0.15	Argument	0.18
Rule	0.73	High	0.33	Ratio	0.46
Citation	0.94			Statute	0.25
Reasoning	0.27			Precedent	0.32
				Ruling (present)	0.32

Table 2. The ratio of matching labels among the top 20 most similar sentences, per label

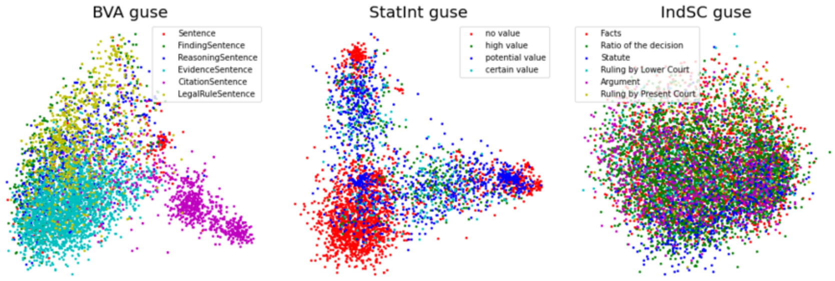


Figure 2. Visualizations of the sentences across the data sets, embedded using the GUSE and reduced to two dimensions using a Principal Component Analysis. The colors correspond to different labels.

5. Discussion

We have introduced and evaluated a lateral annotation framework. We experimented with four types of sentence embeddings and compared them against the random baseline. All of the embeddings showed significant improvements in selecting sentences that are of the same class compared to the random baseline. The neural models show strong performance across the three data sets. In general, they perform similarly, although the Google Universal Sentence Encoder and InferSent seems to have a slight edge. In the BVA data set, the neural models clearly outperform the TF-IDF baseline, while the performance is more balanced in the StatInt and IndSC data sets. Even in the most challenging data set, almost 40% of the 20 closest sentences had the same label as a source sentences. For the other data sets, this number was 70%. This indicates a significant potential benefit for using lateral annotation.

Different areas might benefit from the use of lateral annotations. The assumption behind the method is that sentences that have similar semantic embeddings are likely to belong to the same class that an annotator is aiming to label. This should work better for labeling schemas and data sets where the semantic properties of a sentence are linked to its label, and where the homogeneity of sentences with the same label is high. This can be seen in the per-class analysis of precision, showing that citation sentences and recitation of previous rules and cases are more suitable for lateral annotation. Sentences with less inherent structure and similarity, such as reasoning sentences, seem to perform slightly worse. The Indian Supreme Court data set, which draws from five different domains and uses seven classes, performs worse in a lateral annotation context, which could indicate that more diverse data sets are more difficult to annotate laterally.

Further, the method benefits from data sets where the set of sentences with a particular label is made up of several clusters of semantically similar sentences that the annotator can efficiently scope. For each sentence that is part of such a cluster, the annotator can efficiently label a large number of sentences. Outlier sentences, which do not belong to any larger cluster of sentences with the same label, are less likely to benefit from the method, as they do not assist the annotator in finding other sentences of the same label.

We hypothesize that the legal domain is well-suited for the lateral annotation method. Legal practitioners often use stereotypical language to describe certain facts, including a shared vocabulary and sentence structure. This shared language is more likely to be suited for annotation supported by semantic similarity search, and could significantly speed-up annotating large data sets with per-sentence labels.

6. Future Work

There are multiple ways of building upon this research. First, it is important to investigate how lateral annotation performs in additional real-world scenarios, and compare it to traditional methods of annotation. Second, finding ways to expand our framework by extending vectors with relevant properties or by combining vectors could increase the system's performance. Third, augmenting the method to integrate active learning approaches (where a machine learning model suggests which sample to label next to the annotator) could help to discover more relevant sentences. Furthermore, the approach of annotating sentences laterally could be used at an earlier stage, to support the exploration of data sets and the creation of type systems.

7. Conclusions

In this paper, we have explored a method for the efficient annotation of sentences, by leveraging sophisticated sentence embedding models and approximate nearest neighbour searches. Using these technologies, we designed a method and an interface that allow annotators to label similar sentences in one go across documents, rather than having to episodically label similar sentences as they come up in new documents. We investigated some properties of different possible embeddings and demonstrated the benefits of using the method on three legal data sets.

References

[1] Bhattacharya, P., S. Paul, K. Ghosh, S. Ghosh, and A. Wyner. "Identification of Rhetorical Roles of Sentences in Indian Legal Judgments." arXiv preprint arXiv:1911.05405 (2019).

[2] Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. "Enriching word vectors with subword information." *Transactions of the Association for Computational Linguistics* 5 (2017): 135-146.

[3] Bowman, S., G. Angeli, C. Potts, and C. Manning. "A large annotated corpus for learning natural language inference." *arXiv preprint arXiv*:1508.05326 (2015).

[4] Branting, L. K., C. Pfeifer, B. Brown, L. Ferro, J. Aberdeen, B. Weiss, M. Pfaff, and B. Liao. "Scalable and explainable legal prediction." Artificial Intelligence and Law (2020): 1-26.

[5] Branting, L.K., B. Weiss, B. Brown, C. Pfeifer, A. Chakraborty, L. Ferro, M. Pfaff, and A. Yeh. "Semi-supervised methods for explainable legal prediction." In *Proc. ICAIL 2019*, pp. 22-31. 2019.

[6] Cer, D., Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. St John, N. Constant et al. "Universal sentence encoder." *arXiv preprint arXiv*:1803.11175 (2018).

[7] Conneau, A., D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. "Supervised learning of universal sentence representations from natural language inference data." *arXiv preprint arXiv*:1705.02364 (2017).

[8] Cormack, G., and M. Grossman. "Scalability of continuous active learning for reliable high-recall text classification." In Proc. 25th ACM Int'l Conf. on Info. & Knowledge Management, pp. 1039-1048. 2016.

[9] Cormack, G., and M. Grossman. "Autonomy and reliability of continuous active learning for technology-assisted review." *arXiv preprint arXiv*:1504.06868 (2015).

[10] Devlin, J., M. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv*:1810.04805 (2018).

[11] Firth, J. "A synopsis of linguistic theory, 1930-1955." *Studies in linguistic analysis* (1957).

[12] Harris, Z. "Distributional structure." *Word* 10, no. 2-3 (1954): 146-162.

[13] Hogan, C., R. Bauer, and D. Brassil. "Human-aided computer cognition for e-discovery." In *Proc. 12th Int'l Conf. on Artificial Intelligence and Law*, pp. 194-201. 2009.

[14] Indyk, P., and R. Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality." In *Proc. 30th Annual ACM Symposium on Theory of Computing*, pp. 604-613. 1998.

[15] Iyyer, M., V. Manjunatha, J. Boyd-Graber, and H. Daumé III. "Deep unordered composition rivals syntactic methods for text classification." In *Proc. 53rd ann. meet. ACL* (Vol 1), pp. 1681-1691. 2015.

[16] Johnson, J., M. Douze, and H.é Jégou. "Billion-scale similarity search with GPUs." IEEE Transactions on Big Data (2019).

[17] Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov. "Bag of tricks for efficient text classification." *arXiv preprint arXiv*:1607.01759 (2016).

[18] Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv*:1907.11692 (2019).

[19] Reimers, N., and I. Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." *arXiv preprint arXiv*:1908.10084 (2019).

[20] Šavelka, J., H. Xu, and K. Ashley. "Improving Sentence Retrieval from Case Law for Statutory Interpretation." *Proc. 17th Int'l Conf. on Artificial Intelligence and Law*, pp. 113-122. 2019.

[21] Šavelka, J., G. Trivedi, and K. Ashley. "Applying an interactive machine learning approach to statutory analysis." In Proc. 28th Ann. Conf. on Legal Knowledge & Info. Systems (JURIX'15). IOS Press. 2015.

[22] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

[23] Walker, V., K. Pillaipakkamnatt, A. Davidson, M. Linares, and D. Pesce. "Automatic Classification of Rhetorical Roles for Sentences: Comparing Rule-Based Scripts with Machine Learning." In ASAIL@ ICAIL. 2019.

[24] Walker, V., J. Han, X. Ni, and K. Yoseda. "Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans' claims dataset." *Proc. ICAIL '17*. ACM, 2017.

[25] Waltl, B., J. Muhr, I. Glaser, G. Bonczek, E. Scepankova, and F. Matthes. "Classifying Legal Norms with Active Machine Learning." In JURIX, pp. 11-20. 2017.

[26] Westermann, H., J. Šavelka, V. Walker, K. Ashley, and K. Benyekhlef. "Computer-Assisted Creation of Boolean Search Rules for Text Classification in the Legal Domain." In *JURIX*, pp. 123-132. 2019.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200861

Integrating Domain Knowledge in AI-Assisted Criminal Sentencing of Drug Trafficking Cases

Tien-Hsuan WU [a], Ben KAO [a], Anne SY CHEUNG [b], Michael MK CHEUNG [b], Chen WANG [c], Yongxi CHEN [b], Guowen YUAN [a] and Reynold CHENG [a]

[a] *Department of Computer Science, The University of Hong Kong*
[b] *Faculty of Law, The University of Hong Kong*
[c] *College of Computer Science and Software, Shenzhen University*

Abstract. Judgment prediction is the task of predicting various outcomes of legal cases of which sentencing prediction is one of the most important yet difficult challenges. We study the applicability of machine learning (ML) techniques in predicting prison terms of drug trafficking cases. In particular, we study how legal domain knowledge can be integrated with ML models to construct highly accurate predictors. We illustrate how our criminal sentence predictors can be applied to address four important issues in legal knowledge management, which include (1) discovery of model drifts in legal rules, (2) identification of critical features in legal judgments, (3) fairness in machine predictions, and (4) explainability of machine predictions.

Keywords. judgment prediction, prison term prediction, domain knowledge, fairness, explainability

1. Introduction

With recent advances in machine learning (ML) and AI technology, one of the fastest growing areas in legal technology is the adoption of AI to assist lawyers and judiciaries in handling, processing, and discovering legal knowledge that is embedded in various legal documents such as judgments and ordinances. Works in this area have led to much interesting research, notably in *judgment prediction*, which is the task of predicting or determining various aspects of a legal case given a textual description of a litigation. Early works in judgment prediction (e.g., [1,2]) aim at predicting a certain outcome of a judgment by finding statistical correlations between a set of variables and possible outcomes from historical judgments. In recent years, researchers apply natural language processing (NLP) techniques and tackle the judgment prediction problem by formulating it as various text classification problems. Among them, the following four tasks have attracted much attention lately: **[Outcome prediction]**: to predict the outcome (e.g., settled, dismissed, etc.) of a case [3,4]; **[Article prediction]**: to identify relevant articles in law for a case [5,6,7]; **[Criminal charge prediction]**: to predict the charges of which a defendant should be convicted based on a description of the defendant's criminal activities [5,6,7,8]; and **[Prison term (sentence) prediction]** (or **PTP** for short): to predict the prison term to which a defendant should be sentenced [5,6,7,9]. In this paper we focus on PTP, which is the most challenging one of the four tasks due to the fact that criminal sentencing, as a form of judicial decision, may involve discretionary reasoning that is hard to specify as rules.

Most existing works in PTP build prediction models that take a textual description of a case as input and output a predicted sentence (prison term)[1]. These existing works generally suffer from the following inadequacies.

(Limited accuracy). Many of the works (e.g., [9,7,6]) formulate the PTP problem as a text classification problem in which (a textual description of) a case is classified into a group, each being associated with a prison-term range (e.g., "1-to-2 years", "10 years or above"). The prediction is therefore imprecise. For works that predict a numerical value (e.g., prison term in months), the accuracy is generally not very high. For example, Zhong et al. [10] survey a number of predictors that participated in the Chinese AI and Law Challenge (CAIL2018). The predicted sentences made by the best predictor are on average about 38% off compared with the actual sentences.

(Little use of domain knowledge). Most existing works represent a case as unstructured text and apply NLP and neural networks to construct a sentence predictor. Although there are works that take legal domain knowledge into consideration, the application of which is very limited. For example, in Liu and Chen [9], only the average and the maximum imprisonment terms as stated in related law articles are used as domain knowledge in constructing the prediction models. We remark that legal domain knowledge can potentially help build more accurate models and thus should be effectively exploited.

(Non-explainability). The black-box model of neural networks employed by existing works does not provide sufficient information to explain sentencing decisions. Legal reasoning, however, is an important element in judgments. A sentence predictor should be able to explain the primary logic based on which a final sentence is made.

In this paper we study the PTP problem in the context of drug trafficking cases in Hong Kong. The reasons of our choice are twofold. First, Hong Kong has a common law system. Lower-level courts are bound to follow the rulings of appellate courts and the Court of Final Appeal, and to apply the law consistently. This makes prediction modeling based on historical judgments very applicable. Second, the sentencing of drug trafficking cases generally follows guidelines established in "tariff cases". As we will discuss later, we take these guidelines as domain knowledge and show how they can be integrated into the construction of highly-accurate models. In our study, we take as input a judgment with sentencing information removed. The task of the predictor is to predict a final sentence of the case described in the judgment. Our major contributions are:

- We propose to tackle the PTP problem by integrating legal *domain knowledge* (DK) into ML modeling.
- We show how to apply ML techniques to (1) extract feature values from cases' textual descriptions given a feature set that is specified in the DK; and (2) construct various prediction models that consider the computational elements as specified in the DK.
- We show how our predictors can be applied to address a number of interesting legal/technical issues of PTP.

[1] We use the terms "sentence" and "prison term" interchangeably.

2. Related Work

Recent works formulate judgment prediction problems as text classification tasks, which take a case's textual description as input and predict an outcome using advanced NLP and ML techniques. For example, Vacek et al. [4] predict the outcomes of U.S. Federal Court judgments (such as dismissal by motion, settlement, etc.) and Hu et al. [8] predict the charges against a defendant. Both works use word embeddings to encode cases' descriptions which are then used to train various neural network models. Zhong et al. [6] and Yang et al. [7] use Long Short-Term Memory (LSTM) model to perform judgment prediction. In their works, they solve multiple judgment prediction tasks together to achieve synergetic effect. For example, the result of *article prediction* helps determine a defendant's *charge* as well as a range of the possible *prison term*.

Chen et al. [5] predict a prison term given a defendant's charges. In their work, a case's description and the charges are first encoded using embedding techniques. These embeddings are then fed into a Deep Gating Network (DGN) to determine a criminal sentence. One limitation of their approach is that numerical features, such as the value of stolen properties, are not sufficiently captured by their model. Moreover, the prediction model is trained using cases' textual descriptions. Hence, little legal domain knowledge is used. Liu et al. [9] also use embedding to encode cases. However, their predictor is informed with the ranges (min and max) of imprisonment terms stated in law articles.

3. Methodology

Given a textual description of a drug trafficking case, our task is to predict the prison term in number of months. In this section we describe our drug trafficking sentence predictors. In particular, we discuss the domain knowledge used and how it is integrated into our prediction modeling.

3.1. Data and Domain Knowledge (DK)

We start with a description of the legal data we use in training and evaluating our prediction models. We collected 3,172 English judgments on drug trafficking sentencing from Hong Kong courts. These judgments were handed down from 1998 to 2019. We consulted academic legal experts to identify two kinds of domain knowledge, namely, *substantive domain knowledge* (SDK) and *argumentative domain knowledge* (ADK). For SDK, our experts identified 6 categories of features that represent the most salient facts of a case. These categories are (1) *charge information* (e.g., name of charge and the related ordinances); (2) *drug information* (e.g., kind and weight of drugs involved); (3) *defendant background* (e.g., age, gender, nationality); (4) *mitigating factors* (e.g., guilty plea); (5) *aggravating factors* (e.g., persistent offender); (6) *sentence* (e.g., starting point sentence, final sentence). There are altogether 82 features. Our experts further identified 11 features (out of the 82) that are typically the determining factors of a sentence. These features are listed in Table 1[2]. We employed 11 law students to manually extract the values of all 82 features from each of the 3,172 judgments. As quality assurance, each

[2] We distinguish *guilty-plea* from the other mitigating factors because the sentence reduction for guilty-plea is quite standard.

Category	Feature Description
Drug Weights ($\mathcal{W}$)	weights of drugs (cocaine, heroin, ketamine, methamphetamine) involved
Plea ($\mathcal{P}$)	whether the defendant pleads guilty
Mitigating Factors ($\mathcal{M}$)	defendant shows **remorse**, drugs are mostly **self-consumed**, defendant assists in **controlled delivery**, defendant **gives testimony** in court, defendant has a **good character**
Aggravating Factors ($\mathcal{A}$)	defendant is a **refugee claimant**, defendant is **on bail**, defendant is a **persistent offender**, drugs are trafficked **internationally**

Table 1. Key factors in drug trafficking case sentencing

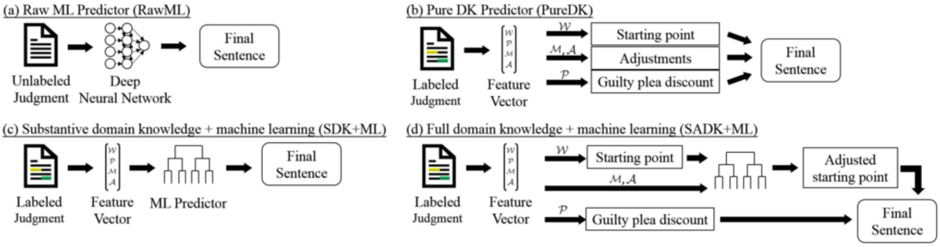


Figure 1. Predictors

judgment is processed by two workers to cross validate the extracted feature values. This "labeling task", which effectively transforms each piece of unstructured judgment text into structured data, took about 6 months to complete. Since our objective is to predict the prison term of a defendant given his/her offense, we remove cases that involve multiple defendants (so that the textual description given in a judgment focuses only on a given defendant). We also remove cases that involve very rare elements (such as rare drugs) because there are insufficient prior judgments to build reliable predictors for those cases. Our final set of judgments consists of 1,641 cases with an average prison term 91.4 months. The lengths of the judgments ranges from 115 words to 2,668 words, with the average being 475 words. ADK refers to a set of procedural rules to calculate the length of the prison term, as explained in detail in Section 3.2. Note that the manually labeled data serves as training data and test data for us to evaluate our prediction models. In Section 3.2.1 we will discuss how we train a machine to automatically extract features from legal documents, thus reducing the high cost of manual labeling.

3.2. Predictors

We consider four predictor models, which are illustrated in Figure 1. These predictors differ in whether and how ML and/or SDK/ADK are used.

Raw ML Predictor (RawML). The first predictor (Figure 1 (a)) does not use any domain knowledge. It takes as input a judgment (with sentencing information masked) and predicts the prison term using a deep neural network. Specifically, we follow the general architecture documented in Zhong et al. [10] to build the predictor — we encode each judgment as a sequence of word embeddings using word2vec [11]. The word embeddings are passed into Stacked Gated Recurrent Unit [12] to obtain a document-level vector, which is then passed to a fully connected layer with sigmoid activation for sentence prediction. We call this baseline approach *RawML* (for "Machine-learning-based with only raw data").

Pure DK Predictor (PureDK). The second predictor (Figure 1 (b)) follows closely how a human judge would determine a prison term. Both SDK and ADK are applied. For SDK, all the features listed in Table 1 are used in the computation. For ADK, the following 3-step procedure, as advised by legal experts, are carried out:
(1) *Starting point calculation:* For popular drugs, there are tariff cases in which sentencing guidelines are established to determine starting point penalty[3]. A typical guideline for a drug gives a list of *weight ranges* and for each range a *prison term range* (e.g., 10-50g of heroin → 5-8 years of imprisonment). Given a drug weight w that falls within a weight range $[w_1, w_2]$ with the corresponding starting point penalty range $[t_1, t_2]$, we assume a linear model in determining a starting point penalty sp, i.e., $sp = t_1 + (w - w_1)/(w_2 - w_1)$.

In many drug trafficking cases, however, more than one drug is involved. In this case, judges apply the *absurdity test*, the *conversion test*, and the *ratio test* to cross-check the appropriate sentence. We apply the *ratio test*[4] in our model to determine an aggregated starting point, which is computed as follows. Let w_1, ..., w_n be the weights of n types of drugs involved, and let $w_T = \sum_i w_i$ be the total weight. A judge would first compute the starting point penalty (sp_i) for type-i drug *as if all the drugs dealt were type-i* (i.e., sp_i is determined by the guideline of type-i drug with weight = w_T). The overall starting point sp is then given by a weighted sum of the sp_i's. Specifically, $sp = \sum_i (sp_i \times w_i / w_T)$.
(2) *Guilty plea discount*: If a defendant pleads guilty, a judge usually assesses a 1/3 discount to the penalty. We thus set a *guilty-plea-factor* (*gpf*) to $(1 - 1/3) = 2/3$ if the defendant pleads guilty; or 1 otherwise.
(3) *Adjustments*: Let $\mathcal{F} = \mathcal{M} \cup \mathcal{A}$ be the set of mitigating and aggravating factors (see Table 1). Note that we model each such factor $f_i \in \mathcal{F}$ as a binary feature, i.e., f_i is either present or absent in a case. For each f_i, a judge would assess a sentence adjustment (adj_i) if f_i is present. We estimate the adjustment adj_i by the average sentence reduction (if f_i is mitigating) or increment (if f_i is aggravating) observed in historical judgments due to factor f_i. An *adjustment factor* (*af*) is estimated by: $af = \prod_{f_i \text{is present}} (1 + adj_i)$.

The final predicted sentence (*fps*) is given by $fps = sp \times gpf \times af$. We call this predictor *PureDK* (for "Pure domain-knowledge-based without machine learning").

Substantive domain knowledge + machine learning (SDK+ML). The third predictor (Figure 1 (c)) is constructed by building regression trees with gradient boosting using the features listed in Table 1 as input and a prison term as output. We call this predictor *SDK+ML* as it utilizes substantive domain knowledge and ML techniques. Note that its construction is completely data-driven. In particular, it is not given any argumentative knowledge such as the *starting point guidelines*, *ratio test*, or *the guilty-plea discount*.

Full domain knowledge + machine learning (SADK+ML). The last predictor (Figure 1 (d)) uses all domain knowledge (SDK and ADK) plus machine learning techniques. We call it *SADK+ML*. It is essentially a hybrid of PureDK and SDK+ML. Similar to PureDK, SADK+ML uses the features specified in the SDK and it follows the procedure given in

[3] *R v Lau Tak-ming & Others* [1990] 2 HKLR 370; *HKSAR v Abdullah Anwar Abbas* [2009] 2 HKLRD 437; *Attorney General v Pedro Nel Rojas* [1994] 1 HKC 342; *HKSAR v Tam Yi-chun* CACC524/2011; *Secretary for Justice v Hii Siew Cheng* [2009] 1 HKLRD 1

[4] *HKSAR v Chan Yuk Leong* [2014] HKLRD (Yrbk) 325

the ADK in determining a starting point penalty, *sp*, and a guilty plea factor, *gpf* (see Steps (1) and (2) under PureDK). While PureDK assesses the adjustment of each mitigating and aggravating factor independently (based on the average adjustments observed in historical judgments), SADK+ML uses ML techniques to learn an overall adjustment model. Specifically, under SADK+ML, we build regression trees with gradient boosting that take starting point *sp*, mitigating factors ($\mathcal{M}$) and aggravating factors ($\mathcal{A}$) as input and predict an adjusted starting point penalty (*asp*). The reason for applying ML in determining adjustments is that occasionally judges do not articulate the adjustment of each factor, but determine an overall adjustment after considering all the factors.

3.2.1. Automatic Feature Extractor (AFE)

Our predictors, except for RawML, assume that features given in the SDK (Table 1) are extracted from judgments. These judgments are said to be "labeled" with the feature values identified. The feature values can be obtained manually, for example, by asking a user of the predictor to input drug types and weights, and to answer 10 yes/no questions for the 10 binary features. Alternatively, if the information of a case is given by a textual description (such as a plaintext judgment), we can apply machine comprehension to automatically extract feature values from text. We have implemented an automatic feature extractor (AFE). In this section we briefly describe the design of the AFE.

We use a combination of *regular expression* (RE) and *deep neural networks* (DNN) methods to extract features from judgments. Specifically, we use regular expressions to find drug types and weights, and guilty plea as the descriptions of these are relatively standard. As an example, from the text *"Defendant ... unlawfully trafficked in 3.56 kilograms of a solid* ***containing 2.29 kilograms of cocaine****"*, our RE extractor recognizes the pattern in boldface and correctly retrieves 2,290g (drug weight) and *cocaine* (drug type) from the text.

We use deep recurrent neural network [13] to classify text to determine the presence/absence of mitigating and aggravating factors. Specifically, a judgment is converted to a sequence of vectors (x_1, ..., x_n) using word2vec. The vectors are then sent to a network that consists of two stacked Bi-LSTM layers [14]:

$$y_i^{(1)} = [\overrightarrow{\mathrm{LSTM}}(x_i);\overleftarrow{\mathrm{LSTM}}(x_i)]; \quad y_i^{(2)} = [\overrightarrow{\mathrm{LSTM}}(y_i^{(1)});\overleftarrow{\mathrm{LSTM}}(y_i^{(1)})], \tag{1}$$

where $y_i^{(j)}$ denotes the output of the i-th vector in layer j. The output of the last unit in the Bi-LSTM layer is passed to a fully connected layer with sigmoid activation.

4. Performance

We conducted experiments to evaluate the four predictors. To recap, RawML takes a plaintext judgment (with sentencing masked) as input and returns a predicted prison term. It is oblivious to any legal guidelines or logic that a judge would usually follow. It serves as a baseline to illustrate how well a pure machine learning approach performs, which is also a typical approach taken by many existing works. PureDK mimics a human judge's decision by considering both substantive and argumentative knowledge. It calculates a prison term based on a feature vector (Table 1) and the three steps we previously described. PureDK thus provides the decision of a (pseudo) human judge as another base-

Predictor	RawML	PureDK	SDK+ML	SADK+ML	SDK+ML (AFE)	SADK+ML (AFE)
Accuracy (%)	73.03	91.52	91.23	92.12	88.04	88.90
$M_{0.3}$ (%)	33.19	4.33	5.91	5.55	9.69	7.86

Table 2. Predictors' accuracies

Factor	(a) # of Cases	(b) Sentence Adjustment	(c) Gini Importance
Show remorse	138	-0.62%	.0853
Self-consumption	222	-8.61%	.2301
Controlled delivery	36	-5.37%	.1605
Give testimony	18	-11.84%	.2243
Good character	24	-3.33%	.0125
Refugee claimant	23	+5.95%	.0392
On bail	29	+5.11%	.0019
Persistent offender	81	+5.91%	.0261
International	468	+5.19%	.2201

Table 3. Impact of factors on sentence prediction

line for comparison. SDK+ML and SADK+ML apply ML techniques and use DK to different extent — SDK+ML uses SDK to obtain a set of important features. The rest of the sentence prediction pipeline is completely done by ML. SADK+ML uses ADK to determine starting point and guilty-plea discount. However, it uses ML to learn a sentence adjustment model, which is not well promulgated in laws. For each of SDK+ML and SADK+ML, we consider two versions: one with feature values provided (by human labelers) and another one with feature values extracted by our AFE. The latter version represents the scenario of a fully automatic predictor that predicts a sentence by comprehending a plaintext case description.

All predictors are trained and evaluated with our corpus of 1,641 drug trafficking judgments (see Section 3.1) using 5-fold cross validation. Given a case, we measure a predictor's accuracy by $1 - (|\hat{y} - y|/y)$, where $\hat{y}$ is the predicted prison term and y is the prison term given in the corresponding judgment (i.e., ground truth). Also, we count the fraction of cases, denoted by M_a, in which a predictor's error $(|\hat{y} - y|/y)$ is at least a. For large a, say 30%, we consider the prediction a "big miss" (because of the substantial error). Table 2 shows the predictors' average accuracies and $M_{0.3}$. From the table, we make some observations.

- RawML's accuracy (73.03%) is much smaller than those of the other three predictors (which are in the 90's). Moreover, about 1/3 of RawML's predictions are big misses (predicted sentences are at least 30% off from ground truths). This shows that it is challenging for a pure machine learning approach to learn the models of hidden logic such as sentencing guidelines, ratio tests, and sentence adjustment.
- PureDK gives a very high accuracy (91.52%) and the lowest big-miss rate ($M_{0.3}$ = 4.33%). Recall that PureDK acts as a (pseudo) human judge by modeling the sentencing procedure based on the knowledge provided to us by human legal experts. The excellent performance of PureDK shows that the model is consistent with the decisions made by different human judges of the historical judgments. This infers that the human judges are very consistent in their judicial decisions on sentencing, closely following guidelines and common principles. We remark that our analysis of PureDK provides a data-driven scientific approach to studying the consistency issue in legal system, which would be otherwise difficult to perform considering the large number and big variety of cases.
- SDK+ML's accuracy (91.23%) is comparable to that of PureDK (91.52%) and it gives a bigger (5.91%) but still small big-miss rate. This shows that it is possible and effective to use ML techniques to perform PTP on drug trafficking cases. The fact that SDK+ML performs much better than RawML shows that substantive domain knowledge is critical to solving the PTP problem. Feature engineering is thus a necessary step in developing an effective solution.

• SADK+ML gives the best accuracy (92.12%), which is even slightly better than PureDK's (pseudo human judge). Recall that SADK+ML uses ML to learn a sentence-adjustment model. As we have previously mentioned, sentence adjustments are sometimes not perfectly articulated in judgments and that judges exercise discretion in determining an overall adjustment when there are multiple mitigating/aggravating factors. Our results show that ML techniques can be effectively applied to learn an aggregated adjustment model.
• Finally, the versions of SDK+ML and SADK+ML that use AFE to automatically extract feature values give very good accuracies and reasonably low $M_{0.3}$. Like RawML, the AFE versions of the predictors comprehend plaintext case descriptions. In particular, they significantly outperform RawML. This shows that our AFE is very effective.

5. Applications

We apply our sentence predictors to perform a number of interesting legal analytics studies. We discussed judgment consistency evaluation in the last section. In this section we briefly discuss four other applications of our predictors.

Model Drift. Our ML-based predictor learns a sentencing model from historical judgments. An implicit assumption is that those judgments follow the same hidden model. An interesting application of our predictor is to detect "model drift", which is a change of the hidden model, by detecting *outliers*. An outlier is a case whose sentencing deviates much from our model's prediction. Among the outlier cases, we find a specific case *C*, which is one of the most early cases in our corpus. It turns out that in case *C*, an unprecedentedly large amount of cocaine was dealt. Shortly after case *C*, a new guideline was laid down that has a binding effect on future similar cases. The model our predictor learns fits the new guideline well as most of the cases in our corpus are after case *C*. Since case *C* does not follow the new guideline, it is an outlier of our predictor. Hence, by outlier detection, our sentence predictor can detect model drifts. It also helps us identify appropriate sets of historical data for constructing models that are valid through different periods of time.

Factor Impact. With SDK, factors that would impact the final sentencing are identified by legal experts. An interesting question is what these factors' relative impacts are. We apply our predictor to provide a quantitative impact analysis.

We first study the impacts of the 9 mitigating/aggravating factors on final sentencing. Table 3, Column (b) shows the average sentence adjustment due to each factor. We see that the adjustments vary from the highest impact (*give testimony*, -11.84%) to the lowest (*show remorse*, -0.62%). The *testimony* factor has the highest impact because a defendant's testimony is often crucial in convicting accomplices and the mastermind of an offense. On the contrary, *remorse* has low impact because drug trafficking is a serious crime and the court seldom reduces sentence for such a minor factor. We remark that this impact analysis helps legal professional to statistically review the key factors in judicial decisions.

We further investigate how the factors impact the predictor's confidence. Table 3, Column (c) shows the normalized *Gini Importance (GI)* of each factor. The GI of a factor *X* quantifies how well we improve the predictor's confidence if *X* is known. The GI of factor *X* is measured by the reduction in the data impurity of each regression tree node that is split based on *X*, weighted by the probability of reaching that node in a

Feature	Group	# of cases	$\bar{e}$	σ_e	p-value
Gender	Male	995	0.0355	0.1891	0.4495
	Female	209	0.0448	0.1558	
Nationality	Local	90	0.0095	0.1095	0.6010
	Foreigner	219	0.0023	0.1090	
Age	Youth (16–20)	170	0.0435	0.2733	0.7476
	Aged 21+	1,090	0.0366	0.1462	

Table 4. Predictor's fairness evaluation

```
[Starting point]
   Heroin (482.82g): 204.9 months;
   Meth (14.38g): 87.5 months;
   Combined (ratio test applied): 209.4 months
[Adjustments]
   Give testimony in court: -11.84%;
   International element: +5.19%;
   Adjusted sentence: 194.2 months.
[Guilty plea]
   Yes (early stage): 1/3 sentence discount.
[Final sentence (predicted)]
   129.5 months.
```

Figure 2. Explanation of the prediction

prediction. From the table, we see that the most important factors in terms of GI are *self-consumption*, *give testimony*, and *international*. Note that even the *international* factor has a mild sentence impact (column (b)), it has a high GI because it is involved in many cases (column (a)). This knowledge allows us to selectively deploy more resources on extracting the high-GI factors, e.g., by multiple manual validations. Low-GI factors, on the other hand, can be more economically extracted by automatic extraction.

Fairness. *Algorithmic bias* refers to systematic errors made by an algorithm that produce unfair, favorable outcomes for one group of users/subjects over other groups. For example, a PTP predictor that tends to over-estimate the prison terms of male offenders and under-estimate those of female is biased. These biases should be avoided [15]. Fairness (absence of bias) is an important quality in an algorithmic decision system. Previous studies (e.g., [16]) show that even if demographic features are excluded in model construction, biases may still occur due to feature correlations. For example, from our data, the aggravating factor *international trafficking* (which is used in sentence prediction) and the demographic feature *nationality* are positively correlated.

We evaluate our predictor in terms of fairness by comparing the errors it makes over different groups of a demographic feature; if there is no statistically significant difference in the errors, the predictor treats the groups similarity and is thus fair. Specifically, for each drug trafficking case, we measure an error $e = (\hat{y} - y)/y$ where $\hat{y}$ is the predicted sentence and y is the actual sentence. Note that e can be positive (overestimate) or negative (underestimate). Table 4 shows the mean ($\bar{e}$) and standard deviation (σ_e) of e for different groups under three demographic features, namely, *gender*, *nationality*, and *age*[5]. These three features are studied because there are prior rulings that forbid biases with respect to them in sentencing[6]. We determine if there is a significant difference between the error means of two groups by Welch's t-Test [17]. A p-value > 0.05 indicates that the evidence of two groups having different means is weak; hence, the predictor is unbiased. From Table 4, we see that the average errors of different groups under each demographic feature are small and are sufficiently similar. Moreover, the p-values are all much larger than 0.05. The predictor is therefore fair.

Prediction Explainability. Besides improving prediction accuracy, the integration of domain knowledge (both SDK and ADK) allows us to design explainable models, which is very important in the legal domain as judges often explain their decisions in judgments. Note that RawML does not apply any domain knowledge and it is difficult to explain

[5] Cases for which a demographic feature is not documented in the judgments are not included in the data. *Race* is also an important factor for fairness evaluation, but it is not included because such information cannot be found from the judgments.

[6] *R v Okuya and Nevaboi* (1984) 6 Cr App R (S) 253, *HKSAR v Hong Chang Chi* [2002] 1 HKC 295, *R v Lau Tak-ming & Others* [1990] 2 HKLR 370

its logic from a legal perspective. In contrast, PureDK models the decision elements that judges generally follow and thus it provides *explainability by design*. To illustrate, Figure 2 is an example output of a sentence predictor we have developed. In this example, the predictor is given a case (*HKSAR v Kwan Yun-hang*) in which the defendant pleaded guilty for importing 482.82g of heroin and 14.38g of methamphetamine hydrochloride. The defendant helped the authority to convict other criminals by giving testimony in court. The sentence passed upon the defendant was 11 years (132 months). Note that our model's prediction of 129.5 months is very close to the actual sentencing.

6. Conclusion

In this paper we studied the prison term prediction (PTP) problem in the context of drug trafficking cases. We considered two kinds of domain knowledge, namely, substantive domain knowledge (SDK) and argumentative domain knowledge (ADK). We showed how the knowledge can be integrated with ML models to construct highly-accurate sentence predictors. Furthermore, we discussed a number of important applications of the predictors. Our study provides an example based on which similar techniques can be derived in other applications and legal domains.

References

[1] Nagel SS. Applying correlation analysis to case prediction. Texas Law Review. 1963;42:1006.

[2] Segal JA. Predicting Supreme Court cases probabilistically: The search and seizure cases, 1962-1981. American Political Science Review. 1984;78(4):891–900.

[3] Sulea OM, Zampieri M, Malmasi S, Vela M, Dinu LP, van Genabith J. Exploring the use of text classification in the legal domain. Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Text. 2017.

[4] Vacek T, Teo R, Song D, Nugent T, Cowling C, Schilder F. Litigation Analytics: Case outcomes extracted from US federal court dockets. In: Proceedings of the Natural Legal Language Processing Workshop 2019; 2019. p. 45–54.

[5] Chen H, Cai D, Dai W, Dai Z, Ding Y. Charge-Based Prison Term Prediction with Deep Gating Network. EMNLP. 2019.

[6] Zhong H, Guo Z, Tu C, Xiao C, Liu Z, Sun M. Legal judgment prediction via topological learning. In: EMNLP; 2018. p. 3540–3549.

[7] Yang W, Jia W, Zhou X, Luo Y. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019.

[8] Hu Z, Li X, Tu C, Liu Z, Sun M. Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics; 2018. p. 487–498.

[9] Liu YH, Chen YL. A two-phase sentiment analysis approach for judgement prediction. Journal of Information Science. 2018;44(5):594–607.

[10] Zhong H, Xiao C, Guo Z, Tu C, Liu Z, Sun M, et al. Overview of CAIL2018: Legal Judgment Prediction Competition. arXiv preprint arXiv:181005851. 2018.

[11] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.

[12] Chung J, et al. Gated feedback recurrent neural networks. In: ICML; 2015. p. 2067–2075.

[13] Pascanu R, Gulcehre C, Cho K, Bengio Y. How to construct deep recurrent neural networks. International Conference on Learning Representations. 2014.

[14] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks. 2005;18(5-6):602–610.

[15] Tonry M. Legal and ethical issues in the prediction of recidivism. Federal Sentencing Reporter. 2014;26(3):167–176.

[16] Tolan S, Miron M, Gómez E, Castillo C. Why Machine Learning May Lead to Unfairness: Evidence from Risk Assessment for Juvenile Justice in Catalonia. ICAIL. 2019.

[17] Welch BL. The generalization of 'Student's' problem when several different population variances are involved. Biometrika. 1947;34(1/2):28–35.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200862

Using Argument Mining for Legal Text Summarization

Huihui Xu [a,1], Jaromír Šavelka [b], and Kevin D. Ashley [a,c,d,2]

[a] *Intelligent Systems Program, University of Pittsburgh*
[b] *School of Computer Science, Carnegie Mellon University*
[c] *Learning Research and Development Center, University of Pittsburgh*
[d] *School of Law, University of Pittsburgh*

Abstract. Argument mining, a subfield of natural language processing and text mining, is a process of extracting argumentative text portions and identifying the role the selected texts play. Legal argument mining targets the argumentative parts of a legal text. In order to better understand how to apply legal argument mining as a step toward improving case summarization, we have assembled a sizeable set of cases and human-expert-prepared summaries annotated in terms of legal argument triples that capture the most important skeletal argument structures in a case. We report the results of applying multiple machine learning techniques to demonstrate and analyze the advantages and disadvantages of different methods to identify sentence components of these legal argument triples.

Keywords. Information retrieval, legal analysis, relevant sentences, argument mining, summarization

1. Introduction

Case summaries can assist legal professionals more easily to identify relevant cases and assess whether to read their full texts. As a more accessible means for the general public to gain some insights into what legal cases contain, summaries may also increase access to justice. A good case summary should include some key information: 1) major **issues** a court addressed in the case, 2) the court's **conclusion** with respect to each issue, and 3) a characterization of the court's **reasons** for reaching the conclusion. We refer to this key information as *legal argument triples* (*IRC triples*). These triples capture a skeletal structure of the legal arguments in a case.

Our ultimate goal is to extract the most important legal argument triples and use them to create succinct, three-sentence summaries that could enable legal researchers to better and more quickly assess what a case is really about and whether it is worth studying in detail. As a step in that direction, we conducted an empirical study of whether a machine learning (ML) model can identify the components of legal argument triples in case summaries prepared by human experts. The human summarizers are legal professionals charged with capturing the most important information in the cases. While their

[1] huihui.xu@pitt.edu
[2] ashley@pitt.edu

summaries are still too long for our intended information retrieval (IR) use case, they appear to contain the most important issues raised, the conclusions reached, and a reason connecting them. Since the experts act as a well-informed filter on importance, it makes sense to capture this information by annotating the summaries rather than the full texts. A more ambitious goal, however, is to generate the triples automatically from the full case texts.

Having developed a detailed annotation scheme, we annotated a sizable set of case summaries in terms of argument triples and also used those annotated summaries to help annotate the corresponding sentences in the full texts. We then applied various traditional ML algorithms and deep neural network models to identify the sentence components of IRC triples in corpora of legal summaries and of the corresponding full text decisions. We explored the use of different sampling strategies with the algorithms. We report the results and compare the advantages and disadvantages of the different methods.

2. Related Work

In order to summarize texts automatically, researchers have applied either abstractive [1,2] or extractive techniques. In summarizing legal texts, researchers have applied mainly the latter. See [3] for a recent survey. These extractive techniques have included: graph-based methods to cluster sentences by topics [4], by similarity based on repetition of legal phrases [5], or by unsupervised learning [6], machine learning classification of rhetorical roles of sentences in legal cases (e.g., FACT, BACKGROUND) [7,8,9], thematic structure [10,6], the rhetorical status of sentences in judgments of the UK House of Lords, [11], or catchphrases [12].

In [13] Zhong, et al. used machine learning to select which sentences in the decision are predictive of the case outcome. The summarizer computes the relative importance of sentences in a legal case document, as measured by their predictiveness and chooses a subset to generate the summary. They partitioned acceptable sentences as classified by type (i.e., Reasoning or Evidential Support sentence) and chose a set of summary sentences using maximum marginal relevance. They concluded, based on a detailed error analysis, that argument mining techniques would be required to identify more conceptual aspects of the decisions. Our focus on identifying legal argument triples is intended to do exactly that. In recent work, Yamada, et al. [14] have applied a similar approach to summarizing Japanese judgments by extracting issues, conclusions, and framings. Our legal argument triples, however, are simpler types, not tailored to Japanese legal judgements. Here, we provide evidence that machine learning can extract the argument triple cases from case summaries and full case texts based on a training set of expert summaries, a resource not available in the cited work.

Argument mining is “the automatic discovery of an argumentative text portion, and the identification of the relevant components of the argument presented there.” [15]. Argument mining research has developed techniques to automatically identify argument components (e.g., premises, claims) in text and argumentative relations (e.g., support, attack) between components in contexts such as document summarization [16], legal information systems [17], and policy modeling platforms [18]. Argumentative relation mining involves determining if a relation holds among particular argument components and classifying the argumentative function of the relation (e.g., support vs. attack). Prior research

Table 1. Annotated data set summary.

			Sentence length (mean token count)			
	Count	# Sentences	Issue	Reason	Conclusion	non-IRC
Summaries	574	7,484	26.0	24.6	18.1	19.5
Full texts	109	23,653	35.5	26.7	25.7	16.8

has dealt with predicting argumentative relationship labels between pairs of argument components, e.g., attachment [19], support vs. non-support [20,21,22], {implicit, explicit}x{support, attack} [23,24] and verifiability of support [25]. In the legal domain, argument mining research has focused on extracting argumentative propositions, premises and conclusions, and nested arguments [26], arguments by example and other argument schemes [27], the rhetorical and other roles that sentences play in legal arguments [8,28], legal factors in domains like trade secret law [29], cited facts and principles (i.e., reasons or warrants) [30], functional and issue-related parts (including analysis and conclusions) [31], segments by topic [32] and segments by linguistic analysis [33,10,34].

3. Data Set

The Canadian Legal Information Institute (CanLII), a non-profit organization created and funded by the Federation of Law Societies of Canada[3] provided 28,733 paired cases and human-prepared summaries. The cases cover different kinds of legal claims and issues presented before Canadian courts. The summaries of those cases were prepared by members of Canadian legal societies.

Two annotators, both second year law students at the University of Pittsburgh, classified sentences from the summaries in terms of three types (i.e., issue, reason, conclusion), which together form “legal argument triples,” and a catch-all category (for all the other sentences):

1. **Issue** – Legal question which a court addressed in the case.
2. **Conclusion** – Court’s decision for the corresponding issue.
3. **Reason** – Sentences that elaborate on why the court reached the Conclusion.
4. **Non-IRC**– Sentences that do not qualify as either of the three types.

For annotation, we randomly selected 574 pairs from the 28,733 case/summary pairs. Annotators were asked to annotate all 574 summaries. After resolving all the annotation disagreements between annotators, we asked them to annotate the full texts corresponding to 109 summaries. Table 1 reports some key statistics about our annotated data set. The statistics of the mean sentence length reveal something of how the summaries are created. The IRC sentences are shorter in the summaries as compared to the corresponding sentences in the full texts. This likely reflects the fact that after selecting a sentence a human expert typically removes anything extraneous for the summary. The opposite holds for the non-IRC sentences, which suggests that the full texts have many short sentences not suitable for summaries (e.g., headings).

The third author, who is a law professor, provided a detailed annotation guideline for student annotators to identify sentences in the summaries that are instances of the

[3] CanLII’s website is https://www.canlii.org/en/.

Table 2. Mean and median Cohen's kappa scores for each sentence type in summaries and full texts. Different degrees of agreement strength correspond to ranges of kappa: values ≤ 0.00 as poor agreement; value $0.00 - 0.20$ indicates slight agreement; value $0.21 - 0.40$ as fair agreement; value $0.41 - 0.60$ as moderate agreement; value $0.61 - 0.80$ as substantial agreement and value $0.81 - 1.00$ as almost perfect agreement.

	Summary				Full text			
	Issue	Reason	Concl.	Overall	Issue	Reason	Concl.	Overall
Mean κ	0.698	0.602	0.698	0.709	0.598	0.591	0.616	0.773
Median κ	1.000	0.700	1.000	0.740	0.780	0.820	0.750	0.860

three categories (i.e., issue, conclusion, and reason). Both annotators attended all the sessions. During those sessions, we did not notice any problems with American law school students working with Canadian cases.

The student annotators employed an online tool, Gloss (developed by the second author), to facilitate the annotation procedure. The annotators then proceeded over a period of several weeks to annotate successive batches of twelve case summaries at a time. After annotating each batch of twelve, the annotators and the first and third authors met via Zoom to resolve any differences and assign the final labels via consensus of both annotators in consultation with the third author.

The procedure for annotating full texts is different from that for summaries. We leveraged the existing summary annotations by allowing the student annotators to quickly target the sentences that are most similar to the annotated summary sentences: for each annotated sentence in a summary, annotators pick up some keywords and utilize them as pointers to locate corresponding sentences in the full text. This process does not require the annotators to read and understand the whole full texts and expedited the process of full text annotation. We believe that finding the triples in the summaries is considerably easier than doing so in the full texts. We hope to develop a strategy for inexpensively annotating full texts of cases that will enable us to amass a sizeable data set (something nonexistent in legal text summarization as of now). Eventually, we hope to project the annotations from the summaries to the full texts automatically. At the moment, however, this is done manually by the annotators. This paper could be understood as a first step toward the desired automation.

We use Cohen's κ [35] to measure the degree of agreement between the two annotators after their independent annotations of each batch of twelve summaries. They annotated N items into C mutually exclusive categories. In our case, there are three mutually exclusive categories — issue, conclusion and reason and the number of items are the number of sentences of each summary. The results of the inter-annotator agreement study are presented in Table 2. The mean Cohen's κ coefficients across all types is 0.716 which indicates a substantial agreement about the nature of the sentence types according to [36]. As shown in the table, the mean of Conclusion agreement scores is the highest whereas the mean of Reason's is the lowest. Reasons are clearly the most challenging since they are always entwined with facts. The other two types are easier because the courts are more explicit in identifying their Issues and the Conclusions. The feedback from the student annotators confirms this observation.

Figure 1 reports the distribution of the final consensus labels of summaries and full texts. Non-IRC is the most frequent label across all the summaries. The reason label is the second most frequent label, while the issue and conclusion labels are less frequent. This result is attuned to our intuition since more valuable sentences (IRC triples in our

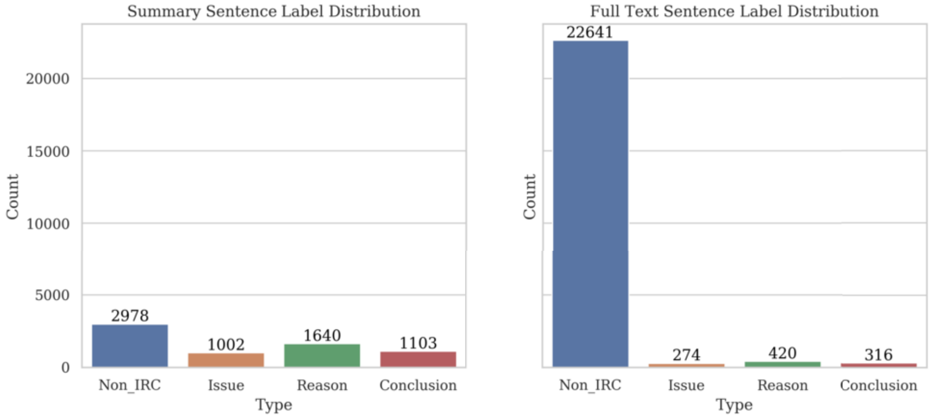


Figure 1. Distribution of annotated IRC type sentences in 574 summaries (left) and in 109 full texts (right). (The total number of full text IRC type sentences is lower because fewer full texts have been annotated than summaries.)

Table 3. Number of summaries that contain issue, reason and conclusion

	Issue	Reason	Conclusion
Count	557	531	574
Ratio	0.970	0.925	1.00

case) are much rarer than less valuable (Non-IRC) sentences. Table 3 reports the number of summaries that contain issue, reason and conclusion sentences. The statistics show that all summaries contain conclusions and well over 90% of summaries have issues (97%) and reasons (93%). This confirms our hypothesis that these sentence types are foundations for a good summary of a legal case.

4. Experiments

As discussed in Section 2, supervised learning techniques with labeled data are frequently used for argument mining. The performance of supervised learning techniques depends, among other things, on the quantity and quality of the annotated data. If the data set is too small, supervised learning algorithms will not have enough data to learn; if the quality of the annotation is not good, the algorithms will not be properly trained despite a large data set. As a result, a sizeable data set with consistent high quality annotation is fundamental for this study.

In order to fully assess the performance of our models, we undertook four types of experiments: Four-way classification on summary only (IRC types and the non-IRC type), four-way classification on full texts, binary classification on summaries (IRC vs non-IRC) and binary classification on full texts. Four-way classification on summaries takes annotated summaries as the training set and tests on unseen summaries. Four-way classification on full texts takes part of the annotated summaries as the training set and tests on full texts. Binary classification requires label transforming, which means we take annotated IRC labeled sentences as one group and the non-labeled sentences (Non-IRC) as the other group. After transforming the labels, we use label-transformed summaries for training and testing on full texts. For four-way and binary summary classifications, we took 50% of our annotated summaries as the training set, 25% of them as the validation set and the rest 25% as the test set. Since the full texts are used as the test set, 75% of annotated summaries are being used as the training set and the rest of them are treated as the validation set. We carefully designed four-way and binary classification experiments on the full texts by excluding corresponding summaries from the training set to prevent leakage of test data into our training.

4.1. Traditional Machine Learning

From traditional ML we work with random forest as one of the most successful algorithms. A random forest algorithm (composed of multiple decision trees) was proposed in [37]. Instead of dealing with a single tree classifier, the random forest is an ensemble of trees and the final result depends on the majority vote. This algorithm significantly improves the classification accuracy because of the randomness of feature selection.

We use TF-IDF values of unigrams, bigrams and POS tags as features for the classifier. We utilize grid search to find the best parameters for this model. Grid search picks the parameters with the best validation accuracy. As [37] mentioned, there is a high probability that random forest will under perform in an extremely imbalanced data set since a bootstrap sample will contain only few or no instances of the minority classes. We observe that the non-IRC sentences are the majority in both summaries and full texts. Some research shows that down-sampling the majority class or over-sampling the minority class are effective ways to boost the performance of tree classifiers. We investigated different sampling strategies along with the random forest classifier and compared the final results: naive over- and under-sampling, performing over-sampling using synthetic minority over-sampling (SMOTE) and down-sampling by using edited nearest neighbor (ENN) [38], and SMOTE and down-sampling using TomekLinks [39].

4.2. Deep Neural Networks

Deep neural network techniques have been widely used for the text classification task because of their high performance. We leverage the power of deep learning to pick the argument triple components. We performed experiments with a Recurrent Neural Network (RNN) based model and a Convolutional Neural Network (CNN) based model.

RNN-based models take text as a sequence of words and are intended to capture the dependencies between words and text structures [40]. We use a variation of this RNN architecture—Long Short-Term Memory (LSTM) network. LSTM performs generally better than vanilla RNNs since LSTM addresses the vanishing gradient problem by introducing multiple gates to control the information flow into and out of the neural cells [40]. In our experiments, we use glove pre-trained word embedding. Those vectors were trained on 6 billion tokens and have 100 dimensions.

CNN-based models are often used for analyzing images. They utilize several filters to extract important features across several convolutional layers [41]. In text classification problems, a CNN model can use different filters looking at different word lengths in a piece of text. We use glove pre-trained word embedding for these experiments, as well.

4.3. FastText

In [42] a computationally efficient method for text classification is proposed. This model has only two layers, the embedding layer and linear layer. It has fewer parameters than most deep learning models. The embedding layer is used for calculating the word embedding, and taking the average of all the word embebddings. The average is stored in a variable and fed to the linear layer. Glove pre-trained word embedding is used for calculating the average in our experiments.

5. Results

The results of experiments described in Section 4 are presented in Tables 4 and 5. Table 4 reports the results of four-way classification on summaries and full texts. Here, CNN achieves the highest F1 scores on IRC types. FastText performs best on picking up issue and conclusion sentences in full texts. We found that the neural models perform better than the random forest model in terms of identifying components of legal argument triples in summaries. Our results also suggest that the different filters of a CNN model pick up more semantic cues regarding the sentence types than RNN models.

Table 5 reports the results of binary classification on summaries and full texts. For summary-only binary classification, we combined all the annotated IRC type sentences into one group. This significantly increases the ratio of majority and minority classes of our training set. Even though the random forest algorithm achieves some highest scores, the neural models and FastText have more stable performances than random forest for picking IRC type sentences in summaries: they all achieve 0.75 or above while the random forest model with naive random under-sampling, SMOTTENN, and SMOTETomek score less than 0.75. The only exception is random forest with oversampling technique.

Since full texts are significantly longer than summaries, the Non-IRC sentences are still significantly more numerous than IRC sentences even though we combine all three types of sentences. Training on an extremely imbalanced data set, random forest with different sampling techniques has a slight competitive edge over neural networks and FastText in classifying unseen samples. Random forest with sampling techniques score over 0.83 on Non-IRC recognition while neural models and FastText score less than 0.82. The performance of the neural models, however, is on par with random forest in picking IRC sentences. This result suggests that random forest may be a better choice for retrieving components of legal argument triples.

6. Discussion

We confirmed that classification techniques are able to extract the components of legal argument triples from summaries and full texts. We performed experiments using random forest and several deep neural network models. Those classifiers performed well on summary-only data. We observed that issue, conclusion, and non-IRC sentences are easier to classify correctly than reason sentences. This observation is aligned with the experience from the annotation phase: issue and conclusion sentences were easier for the human annotators to identify. This indicates that legal common knowledge is embedded

Table 4. F1 scores for the four-way classification on only-summary by using random forest (RF), LSTM, CNN and FastText. The weighted F1 is the average of F1 scores of each type weighted by its support. The suffixes are -O(over-sampling), -U(under-sampling), -w.o.r.(without replacement), - w.r.(with replacement).

	Issue		Reason		Conclusion		Non-IRC		Weighted F1	
	Sum.	Full	Sum.	Full	Sum.	Full	Sum.	Full	Sum.	Full
RF	0.48	0.19	0.39	0.10	0.58	0.23	0.67	**0.95**	0.56	**0.91**
RF-O	0.56	0.23	0.46	0.08	0.61	0.20	0.65	0.91	0.58	0.88
RF-U(w.o.r.)	0.55	0.15	0.48	0.07	0.59	0.15	0.58	0.80	0.55	0.77
RF-U(w.r.)	0.52	0.10	0.50	0.08	0.63	0.17	0.56	0.81	0.55	0.77
RF-SMOTEENN	0.49	0.14	0.09	0.08	0.58	0.21	0.66	0.92	0.48	0.89
RF-SMOTETomek	0.57	0.23	0.46	0.09	0.61	**0.24**	0.66	0.92	0.59	0.89
LSTM	0.59	0.13	0.52	0.09	**0.67**	0.17	0.68	0.85	0.62	0.82
CNN	**0.64**	0.23	**0.54**	0.10	**0.67**	0.20	0.66	0.85	**0.63**	0.82
FastText	0.59	**0.27**	0.52	**0.14**	**0.67**	0.22	**0.69**	0.89	**0.63**	0.86

Table 5. F1 scores for the binary classification on summaries and full texts by using random forest (RF), LSTM, CNN and FastText.

	IRC		Non-IRC		Weighted F1	
	Sum.	Full	Sum.	Full	Sum.	Full
RF	**0.76**	0.16	0.59	0.80	0.69	0.78
RF-O	**0.76**	0.15	0.59	0.83	0.70	0.80
RF-U(w.o.r.)	0.71	0.16	0.63	0.87	0.68	0.84
RF-U(w.r.)	0.70	0.17	0.64	0.92	0.67	0.89
RF-SMOTEENN	0.17	0.05	0.62	**0.98**	0.48	**0.94**
RF-SMOTETomek	0.75	0.16	0.63	0.80	0.70	0.77
LSTM	0.75	0.16	0.58	0.82	0.68	0.79
CNN	0.75	0.16	0.62	0.78	0.69	0.72
FastText	**0.76**	**0.18**	**0.65**	0.82	**0.72**	0.79

in the usage of semantic tokens and that classifiers can recognize them by training on a sizeable data set.

The more challenging task is migrating semantic cues of these sentence types to the broader context of full texts. Performance drops significantly when classifying sentences in full texts. One reason could be that the number of IRC sentences is still too few for training an ML classifier. We discovered that some sampling techniques helped to address the problem of imbalance; the combination of over-sampling and under-sampling techniques in SMOTETomek performed better than the others.

7. Conclusions and Future Work

We experimented with several ML models to identify components of legal argument triples by utilizing annotated human-generated summaries. We confirmed that classification techniques can extract components of these triples in both summaries and full texts. Based on the detailed discussion and evaluation, we found that neural models and FastText show promising results and some sampling techniques could be useful for boosting the performance of random forest.

In the future, we plan to increase the size of the annotated data set. The total number of case summaries is 574. We have used only 465 of them as a training set for our full text sentence classification because we needed to prevent our models from getting any cues from annotations of corresponding summaries. The data set supported our experiments in evaluating if ML techniques could identify components of legal argument triples and in recognizing challenges faced in this task. The data size, however, is still not large enough to draw finer conclusions in terms of comparing performance of different models where they reach similar levels of performance. A larger data set will also be helpful for testing on models that require careful hyperparameter tuning.

After improving a system's ability to identify sentence components of IRC triples, we will explore how best to identify related issues, conclusions, and reasons and to combine and present them as effective extractive case summaries.

Acknowledgement

Grants from the Autonomy through Cyberjustice Technologies Research Partnership at the University of Montreal Cyberjustice Laboratory supported this work. The Canadian Legal Information Institute provided the corpus of paired legal cases and summaries.

References

[1] K. Ganesan, Ch. Zhai, and J. Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. *Coling 2010*, 2010.

[2] A. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[3] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh. A comparative study of summarization algorithms applied to legal case judgments. In *Eur. Conf. on Info. Retrieval*, pages 413–428. Springer, 2019.

[4] M. Kim, Y. Xu, and R. Goebel. Summarization of legal texts with high cohesion and automatic compression rate. In *JSAI Int'l Symposium on Artificial Intelligence*, pages 190–204. Springer, 2012.

[5] F. Schilder and H. Molina-Salgado. Evaluating a summarizer for legal text with a large text collection. In *3rd Midwestern Computational Linguistics Colloquium (MCLC)*, 2006.

[6] M. Moens. Summarizing court decisions. *Info. processing & management*, 43(6):1748–1764, 2007.

[7] C. Grover, B. Hachey, I. Hughson, and C. Korycinski. Automatic summarisation of legal documents. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 243–251, 2003.

[8] M. Saravanan and B. Ravindran. Identification of rhetorical roles for segmentation and summarization of a legal judgment. *Artificial Intelligence and Law*, 18(1):45–76, 2010.

[9] M. Yousfi-Monod, A. Farzindar, and G. Lapalme. Supervised machine learning for summarizing legal documents. In *Canadian Conference on Artificial Intelligence*, pages 51–62. Springer, 2010.

[10] A. Farzindar and G. Lapalme. Legal text summarization by exploration of the thematic structure and argumentative roles. In *Text Summarization Branches Out*, pages 27–34, 2004.

[11] B. Hachey and C. Grover. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345, 2006.

[12] Vu Tran, Minh Le Nguyen, Satoshi Tojo, and Ken Satoh. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Art. Int. and Law*, pages 1–27, 2020.

[13] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. Ashley, and M. Grabmair. Automatic summarization of legal decisions using iterative masking of predictive sentences. In *Proc. 17th Int'l Conf. AI & Law*, pages 163–172, 2019.

[14] H. Yamada, S. Teufel, and T. Tokunaga. Building a corpus of legal argumentation in japanese judgement documents: towards structure-based summarisation. *Art. Int. and Law*, 27(2):141–170, 2019.

[15] A. Peldszus and M. Stede. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.

[16] S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.

[17] R. Mochales Palau and M. Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proc. 12th int'l conference on artificial intelligence and law*, pages 98–107, 2009.

[18] E. Florou, S. Konstantopoulos, A. Koukourikos, and P. Karampiperis. Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 49–54, 2013.

[19] A. Peldszus and M. Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, pages 938–948, 2015.

[20] O. Biran and O. Rambow. Identifying justifications in written dialogs by classifying text as argumentative. *International Journal of Semantic Computing*, 5(04):363–381, 2011.

[21] E. Cabrio and S. Villata. Combining textual entailment and argumentation theory for supporting online debates interactions. In *Proc. 50th Ann. Mtg. Assoc. for Comp. Ling. (Vol. 2)*, pages 208–212, 2012.

[22] C. Stab and I. Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, 2014.

[23] F. Boltužić and J. Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, 2014.

[24] H. Nguyen and D. Litman. Context-aware argumentative relation mining. In *Proc. 54th Ann. Mtg. of the Assoc. for Comp. Linguistics (Vol. 1)*, pages 1127–1137, 2016.

[25] J. Park and C. Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the first workshop on argumentation mining*, pages 29–38, 2014.

[26] R. Mochales and M. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[27] V. Feng and G. Hirst. Classifying arguments by scheme. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 987–996, 2011.

[28] A. Bansal, Z. Bu, B. Mishra, S. Wang, K. Ashley, and M. Grabmair. Document ranking with citation information and oversampling sentence classification in the luima framework, 2016.

[29] M. Falakmasir and K. Ashley. Utilizing vector space models for identifying legal factors from text. In *JURIX*, pages 183–192, 2017.

[30] O. Shulayeva, A. Siddharthan, and A. Wyner. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law*, 25(1):107–126, 2017.

[31] J. Savelka and K. Ashley. Segmenting u.s. court decisions into functional and issue specific parts. In *Proceedings, 31st Int. Conf. on Legal Knowledge and Information Systems, Jurix*, pages 111–120, 2018.

[32] Qi. Lu, J. Conrad, K. Al-Kofahi, and W. Keenan. Legal document clustering with built-in topic segmentation. In *Proc. 20th ACM int'l conf. Info. and knowledge management*, pages 383–392, 2011.

[33] C. Grover, B. Hachey, and C. Korycinski. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 33–40, 2003.

[34] A. Wyner, R. Mochales-Palau, M. Moens, and D. Milward. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*, pages 60–79. Springer, 2010.

[35] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[36] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[37] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[38] G. Batista, R. Prati, and M. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[39] G. Batista, A. Bazzan, and M. Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18, 2003.

[40] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*, 2020.

[41] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[42] A. Joulin, E. Grave, and T. Bojanowski, P.and Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200863

Interpretations of Support Among Arguments

Liuwen YU [a,b,c,1], Réka MARKOVICH [a] and Leendert VAN DER TORRE [a,d]
[a] *University of Luxembourg, Luxembourg*
[b] *University of Bologna, Italy*
[c] *University of Turin, Italy*
[d] *Zhejiang University, China*

Abstract. The theory of formal argumentation distinguishes and unifies various notions of attack, support and preference among arguments, and principles are used to classify the semantics of various kinds of argumentation frameworks. In this paper, we consider the case in which we know that an argument is supporting another one, but we do not know yet which kind of support it is. Most common in the literature is to classify support as deductive, necessary, or evidentiary. Alternatively, support is characterized using principles. We discuss the interpretation of support using a legal divorce action. Technical results and proofs can be found in an accompanying technical report.

Keywords. formal argumentation, abstract argumentation, bipolar argumentation, principle-based approach, legal interpretation

1. Introduction

The theory of formal argumentation distinguishes and unifies different notions of attack, support and preference among arguments. For example, whereas in structured argumentation attack among arguments can be based on rebut, undercutting or undermining, at the abstract level all these kinds of attack are treated in a uniform way. Likewise, deductive, necessary and evidentiary support can be unified at the abstract level [14]. The picture that emerges from the formal argumentation literature is that there is a broad agreement on how to interpret attack, even when different kinds of semantics have been proposed, whereas there is less consensus on the interpretation of support. Moreover, different kinds of support can occur in the same framework, and each variant of support can be more prominent in a particular application.

The common approach in the literature classifies a support among arguments as deductive support, necessary support or evidentiary support. Deductive support [2] captures the intuition that if a supports b, then the acceptance of a implies the acceptance of b, and as a consequence the non-acceptance of b implies the non-acceptance of a. Evidentiary support [11,10] distinguishes prima-facie from standard arguments, where prima-facie arguments do not require any support from other arguments to stand, while standard ar-

[1] This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie ITN EJD grant agreement No 814177.

guments must be supported by at least one prima-facie argument. Necessary support [9] captures the intuition that if a supports b, then the acceptance of a is necessary to get the acceptance of b, or equivalently the acceptance of b implies the acceptance of a.

In addition to this classification, we can consider the principles the support relation satisfies. In formal argumentation, principles can be used as a guideline for choosing the appropriate definitions and semantics depending on various needs. After the principles are chosen, it can be seen at a second step whether there is a semantics satisfying that set of principles. If a set of principles corresponds to one of the semantics then the support can be classified as such, but it may also be the case that no semantics corresponds to the desired set of principles.

In this paper, we consider the case in which an argument supports another one, but we do not know yet which kind of support it is. We consider a legal divorce action in which the interpretation of support is in close relation to the interpretation of law itself. In a divorce action, a judge should decide about the custody according to the child's best interest. The civil code says that, when deciding, the judge has to take the child's opinion into consideration. We also show how the interpretation of this rule influences the interpretation of the support relation, and how this latter interpretation influences the judgement.

Our paper contributes to the discussion on the formalization of legal interpretation in the following way. The role of interpretation is crucial in law, but it is also a source of criticism of using logic-based methods in modelling legal reasoning. For example, Prakken reminds to Leith's warning that the knowledge-engineer's interpretation when formalizing is necessarily premature, as the authority of interpretation of law is assigned to the judiciary [16]. Addressing this criticism, the literature on legal interpretation has discussed the possibility that legal knowledge-based systems contain alternative syntactic formalizations. Prakken observes that while on the syntactic level formalization commits us to a given interpretation, on the conceptual level, classification of factual situations as legal concepts is not an issue of logical form [16, p.14]. Alternatively, we can restrict the investigation by saying that "the only aspects of legal reasoning which can be formalized are those aspects which concern the following problem: *given* a particular interpretation of a body of legal knowledge, and *given* a particular description of some legal problem, what are then the general rational patterns of reasoning with which a solution to the problem can be obtained?" [16, p.4]. If a formal framework itself offers the different interpretations, though, then using it might be directly exploitable to the comparison of the different possibilities and routes of reasoning given each interpretation.

In a recent paper, Prakken [18] argues that for the validation of bipolar argumentation theory, the so-called theory-based validation is preferred to an empirical validation [13], which itself is preferred to an intuition-based validation. We agree with this ordering, but we believe that the principle-based analysis complements these validation methods, and that the theory of formal argumentation needs to be complemented with examples and case studies about the use of the theory. This paper contributes to the latter two areas.

The layout of this paper is as follows. In Section 2 we briefly repeat the definitions of bipolar argumentation, and we introduce a principle-based approach. In Section 3 we apply the theory to the legal divorce action. All technical details and proofs of this paper can be found in a technical report [20], and will be added to the journal extension of this paper.

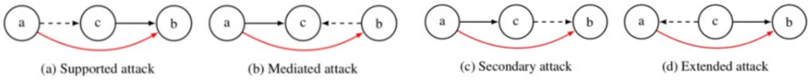
2. Formal approaches to the interpretation of support among arguments

For completeness we repeat the basic definitions of abstract argumentation theory.

Definition 1 (Dung semantics [6]) *An* argumentation framework *(AF) is a tuple* $\langle \mathcal{A}, \mathcal{R} \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary attack relation over* $\mathcal{A}$. $E \subseteq \mathcal{A}$ *is* conflict-free *iff* $\nexists a, b \in E$ *such that* $(a,b) \in \mathcal{R}$. $E \subseteq \mathcal{A}$ defends c *iff* $\forall b \in \mathcal{A}$ *with* $(b,c) \in \mathcal{R}$, $\exists a \in E$ *such that* $(a,b) \in \mathcal{R}$. $E \subseteq \mathcal{A}$ *is* admissible *iff it is conflict-free and defends all its elements. A conflict-free* $E \subseteq \mathcal{A}$ *is a* complete extension *iff* $E = \{a | E \text{ defends } a\}$. $E \subseteq \mathcal{A}$ *is the* grounded extension *iff it is the smallest (for set inclusion) complete extension.* $E \subseteq \mathcal{A}$ *is a* preferred extension *iff it is the largest (for set inclusion) complete extension.* $E \subseteq \mathcal{A}$ *is a* stable extension *iff it is admissible and attacks all arguments in* $\mathcal{A} \backslash E$.

The semantics of deductive and necessary support is based on a reduction of a bipolar framework to an argumentation framework together with a Dung semantics. Based on various interpretations, these reductions add indirect attacks obtained from sets of attack and support relations, then from the obtained indirect attacks and the support additional indirect attacks can be added and so on. We follow the style and notation of Polberg [12], and the reductions are visualized in Figure 1.

Definition 2 (Deductive and necessary support [3,12]) *A* bipolar argumentation framework *(BAF, for short) is a 3-tuple* $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$, *adding a binary support relation* $\mathcal{S} \subseteq \mathcal{A} \times \mathcal{A}$ *to AFs. In addition six reductions from BAF to AF are defined:*
SupportedReduction: $RS(BAF) = (\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{sup})$, $\mathcal{R}^{sup} = \{(a,b) | (a,c)$ *is in the transitive closure of* $\mathcal{S}, (c,b) \in \mathcal{R}\}$*;*
MediatedReduction: $RM(BAF) = \left(\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{med}\right)$, $\mathcal{R}^{med} = \{(a,b) | (b,c)$ *is in the transitive closure of* $\mathcal{S}$, $(a,c) \in \mathcal{R}\}$*;*
SecondaryReduction: $R2(BAF) = (\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{sec})$, $\mathcal{R}^{sec} = \{(a,b) | (c,b)$ *is in the transitive closure of* $\mathcal{S}, (a,c) \in \mathcal{R}\}$*;*
ExtendedReduction: $RE(BAF) = \left(\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{exd}\right)$, $\mathcal{R}^{exd} = \{(a,b) | (c,a)$ *is in the transitive closure of* $\mathcal{S}, (c,b) \in \mathcal{R}\}$*;*
DeductiveReduction: $RD(BAF) = (\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{sup} \cup \mathcal{R}^{med}_{\mathcal{R}sup})$, *where* $\mathcal{R}^{med}_{\mathcal{R}sup} = \{(a,b) | (b,c)$ *is in the transitive closure of* $\mathcal{S}, (a,c) \in \mathcal{R}$ *or* $(a,c) \in \mathcal{R}^{sup}\}$*;*
NecessaryReduction: $RN(BAF) = (\mathcal{A}, \mathcal{R} \cup \mathcal{R}^{sec} \cup \mathcal{R}^{ext})$.



(a) Supported attack (b) Mediated attack (c) Secondary attack (d) Extended attack

Figure 1. Four kinds of indirect attacks as an intermediate step towards semantics for BAFs

We write $\mathcal{E}_{SR}$ for the function from BAF to sets of extensions, $\mathcal{E}_{SR}(BAF) = \mathcal{E}_S(R(BAF))$, where S is one of the Dung semantics (grounded, complete, preferred or stable) and R is one of the reductions (RS, RM, R2, RE, RD or RN). Thus we have 6 × 4 BAF semantics.

We represent evidentiary support using self-supporting arguments, see the technical report [20] for the comparison with other kinds of formalizations of evidentiary support.

The basic idea is that every argument in the extension and every attacker is now an evidentiary chain from a self-supporting argument, a is self-supporting represented as $(a,a) \in \mathcal{S}$.

Definition 3 (Evidentiary support) *Given a $BAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$. A sequence $(a_0, \ldots, a_n)$ of elements of $\mathcal{A}$ is an evidentiary sequence for argument a_n iff $(a_0, a_0) \in \mathcal{S}$, and for $0 \leq i<n$ we have $(a_i, a_{i+1}) \in \mathcal{S}$. A set of arguments $S \subseteq \mathcal{A}$ e-defends argument $a \in \mathcal{A}$ iff for every evidentiary sequence $(a_0, \ldots, a_n)$ where a_n attacks a, there is an argument $b \in S$ attacking one of the arguments of the sequence. Moreover, a set of arguments S is e-admissible iff for every argument $a \in S$ there is an evidentiary sequence $(a_0, \ldots, a)$ such that each $a_i \in S$ (a is e-supported by S), S is conflict free, and S e-defends all its elements. A set of arguments is an e-complete extension iff it is e-admissible and it contains all arguments it e-defends; it is e-grounded extension iff it is a minimal e-complete extension; and it is e-preferred if it is maximal e-admissible extension. Moreover, it is e-stable if for every for every evidentiary sequence $(a_0, \ldots, a_n)$ where a_n not in S, we have an argument $b \in S$ attacking an element of the sequence.*

We introduce various principles for bipolar argumentation. Transitivity (P1, TRA) and closure (P2, CLO) of an extension under supported arguments are introduced by Cayrol et al. [4], inverse closure (P3) was introduced by Polberg [12]. Number of extensions (P4) says that adding support relations can only lead to a decrease of extensions. BAF directionality (P5) is a generalization of Dung's central principle of directionality. Global support (P6) and grounded support (P7) are two ways to characterize evidentiary support. Some additional principles are defined in the technical report [20].

Definition 4 (Principles) *Given a $BAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$, a set U is unattacked and unsupported if and only if there exists no $a \in \mathcal{A} \backslash U$ such that a attacks an argument of U or a supports an argument of U. The set of all sets unattacked and unsupported arguments in BAF is denoted $US(BAF)$. A semantics $\mathcal{E}_{SR}$ satisfies principle P iff for all BAF, for all E in $\mathcal{E}_{SR}(BAF)$, we have:*

P1. Transitivity *If $(a,b) \in \mathcal{S}$, $(b,c) \in \mathcal{S}$, then $\mathcal{E}_{SR}\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle = \mathcal{E}_{SR}\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \cup (a,c) \rangle$.*

P2. Closure *If $(a,b) \in \mathcal{S}$ and $a \in E$, then $b \in E$.*

P3. Inverse Closure *If $(a,b) \in \mathcal{S}$ and $b \in E$, then $a \in E$.*

P4. Number of extensions *For all $\mathcal{S}'$, $|\mathcal{E}_{SR}(\mathcal{A}, \mathcal{R}, \mathcal{S} \cup \mathcal{S}')| \leqslant |\mathcal{E}_{SR}(\mathcal{A}, \mathcal{R}, \mathcal{S})|$.*

P5. BAF Directionality *$U \in US(BAF)$, it holds that $\sigma\ (BAF_{\downarrow_U}) = \{E \cap U | E \in \sigma(BAF)\}$, $BAF_{\downarrow_U} = (U, \mathcal{R} \cap U \times U, \mathcal{S} \cap U \times U)$ is a projection, and $\sigma\ (BAF_{\downarrow_U})$ are the extensions of the projection.*

P6. Global support *If $a \in E$, then there must be an argument b such that $b \in E$, and b supports a.*

P7. Grounded support *If $a \in E$, then there must be an argument $b \in E$ and $(b,b) \in \mathcal{S}$ (or $(a,a) \in \mathcal{S}$), such that there is a support sequence $(b, a_0, \ldots, a_n, a)$, all $a_i \in E$.*

When analyzing an example with support relations, we can consider whether for this example the principles hold or not. The following table summarizes the relations between the reductions and the principles. See the technical report [20] for the proofs of these relations.

Table 1. Comparison among the reductions and the proposed principles. We refer to Dung's semantics as follows: Complete ($\mathbb{C}$), Grounded ($\mathbb{G}$), Preferred ($\mathbb{P}$), Stable ($\mathbb{S}$). When a principle is never satisfied by a certain reduction for all semantics, we use the × symbol.

Red.	P1	P2	P3	P4	P5	P6	P7
RS	$\mathbb{CGPS}$	$\mathbb{GCPS}$	×	$\mathbb{G}$	$\mathbb{CGP}$	×	×
RM	$\mathbb{CGPS}$	$\mathbb{GCPS}$	×	$\mathbb{G}$	×	×	×
R2	$\mathbb{CGPS}$	×	$\mathbb{GCPS}$	$\mathbb{G}$	$\mathbb{CGP}$	×	×
RE	$\mathbb{CGPS}$	×	$\mathbb{GCPS}$	$\mathbb{G}$	×	×	×
RD	$\mathbb{CGPS}$	$\mathbb{GCPS}$	×	$\mathbb{G}$	×	×	×
RN	$\mathbb{CGPS}$	×	$\mathbb{GCPS}$	$\mathbb{G}$	×	×	×
e-	$\mathbb{CGPS}$	×	×	$\mathbb{G}$	×	$\mathbb{GCPS}$	$\mathbb{GCPS}$

3. Divorce action

Consider the bipolar framework visualized below (the detailed description of the divorce case and its scenarios comes after). The figure should be read as follows. A normal arrow visualizes attack, a dashed arrow visualizes support, a double box visualizes a prima facie argument which is self-supporting and single box visualizes a standard argument that does not support itself.

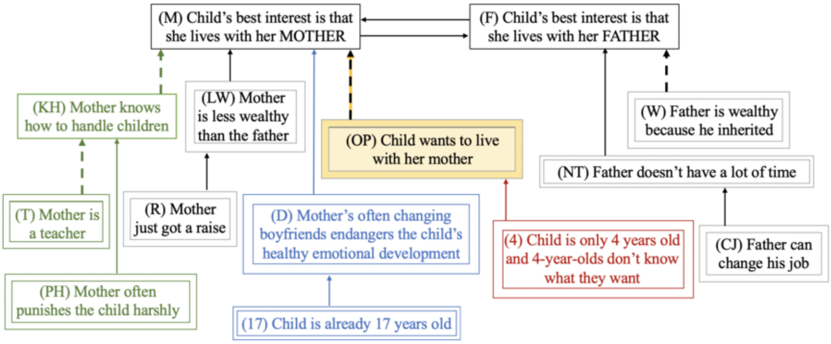

(M) Child's best interest is that she lives with her MOTHER
(F) Child's best interest is that she lives with her FATHER
(KH) Mother knows how to handle children
(LW) Mother is less wealthy than the father
(OP) Child wants to live with her mother
(W) Father is wealthy because he inherited
(NT) Father doesn't have a lot of time
(T) Mother is a teacher
(R) Mother just got a raise
(D) Mother's often changing boyfriends endangers the child's healthy emotional development
(4) Child is only 4 years old and 4-year-olds don't know what they want
(CJ) Father can change his job
(PH) Mother often punishes the child harshly
(17) Child is already 17 years old

Figure 2. Divorce action

We first consider the graph with only the arguments in black and orange and the relations among them. The basic dilemma is represented by two arguments attacking each other, stating respectively that the child's best interest is that she lives with her mother (M) or that the child's best interest is that she lives with her father (F). Obviously, there may be additional reasons for these conclusions which we do not make explicit here, and in order to illustrate the dilemma-nature of the situation the judge has to deal with, we consider a well-balanced case. There are arguments attacking (M) and (F), and arguments attacking those attackers. If we do not consider support, then the grounded extension is $\{CJ, W, OP, R\}$, two preferred extensions are $E_1 = \{R, M, W, OP, CJ\}$ and $E_2 = \{F, CJ, W, OP, R\}$. The judge cannot make any decision based on these two extensions, thus, further investigation is needed in this case.

In general, a support relation can be used to choose or select an extension, and this intuition is reflected by Principle P4 (short for P4 and the same for other principles). This so-called *number of extensions* principle says that adding support relations does not decrease the number of extensions. However, the analysis in Table 1 shows that this principle only holds trivially, for the grounded semantics. As there is only one extension under that semantics, the number of extensions can never increase. But for all other semantics, it is possible that the number of extensions grows when adding support relations. As we will see in the remainder of this section, that does not happen for this example.

The first support relation we interpret is the support of the father being wealthy (W) to the argument that it is in the child's best interest that she lives with her father (F). We leave the support of mother side in this following paragraph. There are different options for the interpretation of the support between (W) and (F). If the interpretation is deductive, then we add the supported attacks from (W) to (M) i.e., using RS, we have only one preferred extension $\{W,CJ,OP,R,F\}$. If we add the mediated attack from (M) to (W) with the same interpretation but under RM, then we have two preferred extensions: $E_1 = \{M,R,OP,CJ\}$ and $E_2 = \{R,F,W,CJ,OP\}$. If we choose RD, we still have the two preferred extensions containing (M) and (F) separately. When we consider the interpretation of this support as necessary, saying, for example, that raising a child does take a lot of money, it means adding secondary or extended attack, since there is no attack coming from or going towards (W), we have the same preferred extensions under these two reduction: $E_1 = \{R,M,W,OP,CJ\}$ and $E_2 = \{F,CJ,W,OP,R\}$. We see this as a clear case of deductive support, but as we saw, this doesn't solve the case either.

A similar analysis can be given using the P2 and P3 instead of the reductions, which as Table 1 shows are characteristic for deductive and necessary support respectively. P2 represents Closure and is characteristic for deductive support. It says that if (W) is accepted then (F) is accepted. So by contraposition, it means that if (F) is not accepted, then also (W) is not accepted. This implies that for the extension containing (M), where (F) is not accepted, by accepting closure, (W) cannot be in the extension. Based on the above, the extensions show the preferred semantics under deductive reduction satisfies P2. However, in this scenario, the extensions still do not give decisive influence to the decision of the case. It may seem counterintuitive that under the deductive interpretation, a mediated attack is added from (M) to (W), as there does not seem to be a reason to question the wealth of the father. This surprising indirect attack is partly explained by P5, which shows that mediated attack does not satisfy BAF directionality. This reflects that the direction of the indirect attack goes against the direction of the attacks and supports in the framework.

If we consider the attacks and support of the argument (M), first we need to note that, according to the judicial practice and the public opinion, for decades, (M) was taken for granted: judges automatically gave the custody for the mother, that is, (M) was a *prima facie argument* and (M) was the only argument being accepted. Thus, traditionally, (M) could have been modeled as a self-supporting argument. However, the judicial practice and the public opinion have been changing, so in the figure above, we modeled (M) as a standard argument requiring evidentiary support.

While the argument structure on the mother's side seems to be the same as the father's side, there is a difference coming from the law. The supporting argument might have a special status because of the rules of the Civil Code: *the judge has to take the child's opinion into consideration when deciding about custody*. The variants of the support interpretations and their relation to the interpretation of law can be shown with the analysis of this rule. We assume that the child wants to live with her mother (OP). What does this mean? One can say that the obligation of taking an argument into consideration means that the (*OP*) is prima facie and has to be accepted. If it is a prima facie argument, (*M*) receives the evidentiary support it needs. But this in itself doesn't decide how argument (*OP*) affects the extension. The extension depends on how we interpret the support relation between (*OP*) and (*M*): deductive or necessary. It seems to be very intuitive to interpret the support relation deductive: the obligation of taking the opinion into consideration is apparently very much in align with what deductive support means: if we accept the opinion (which is prima facie) then we have to accept (M) too. But if we interpret the relation between (*OP*) and (*M*) deductive, under RS we have the only preferred extension $\{OP, CJ, R, W\}$. Under RM we have the two preferred extensions $\{F, W, CJ, R\}$ and $\{M, R, CJ, OP\}$. These results in this scenario, on one hand, reflect the RS satisfies P8, because supported attack is directional. On the other hand, this result means that even the deductive support between the prima facie (OP) and (M), the fact that the child wants to live with her mother won't decide the case in the favor of her if there is some support on the father's side too. However, especially in such a well-balanced case, the judge's obligation to take the child's opinion into consideration might mean that it should be decisive. In order to show what that legal interpretation would mean formally, we need another approach. There is also a way to add the supported attack from (*OP*) to (*F*), and mediated attack from (*M*) to (*W*), by doing so we have the only preferred extension $\{R, OP, M, CJ\}$. That is, in order to give the opinion a decisive nature, considering both relations between (OP) and (M) and (W) and (F) still deductive support, we do so under different reduction: using RS for (OP) and (M), and RM for the other. This context-dependent solution is needed to represent the given legal interpretation.

We now consider a scenario visualized in red and black in which (*OP*) is attacked by (4): the child is only 4 years old and 4-year-olds don't know what they want. Argument (4) impairs that the child can form a reliable opinion at all, that is, (4) attacks (*OP*). If the support between (*OP*) and (*M*) is deductive, under RS the unique preferred extension is $\{R, F, 4, CJ, W\}$, while changes to $\{R, F, 4, CJ, W\}$ and $\{R, 4, CJ, M, OP\}$ under RM. If the interpretation of the support is necessary, under R2 the only extension should be $\{R, 4, F, CJ, W\}$, while under RE, the extension is the same as the framework without considering support. The P3 Inverse Closure says that if (*M*) is accepted then (*OP*) is accepted. So by contraposition, it means that if (*OP*) is not accepted, then also (*M*) is not accepted. This implies that for the extension containing (*F*), where (*M*) is not accepted, by accepting Inverse Closure, (*OP*) cannot be in the extension. Based on the above, the extensions show the preferred semantics under necessary interpretation satisfies P3.

Let's consider another scenario, as visualized in blue and black. In a Hungarian case, the court emphasized that the child's opinion is decisive concerning the custody, *unless* the child's healthy development would be endangered in the environment she would choose. This can be translated as the deductive nature of the support depends on whether there is a specific argument (of being endangered) attacking (*M*). The child wants to live with her mother, but the mother often changes her boyfriends, and according to the judge, this would endanger the child's healthy emotional development (*D*). If the interpretation of support is deductive and under RS, the only preferred extension is $\{R, 17, OP, CJ, W\}$, the results under RM and RD are not decisive, either.

Finally, we consider a scenario visualized in green and black. The mother is a teacher, which supports that she knows how to handle children, and this again clearly supports that the child's best interest is to live with her mom. However, we also have the argument that mother often punishes the child harshly attacking (*KH*). While the first support relation between (*T*) and (*KH*) is deductive, it seems reasonable to say the one between (*KH*) and (*M*) is necessary: it is difficult to defend a view as it is fine to give custody to someone who cannot handle children. (*M*) receives secondary attack from (*PH*) with the interpretation of necessary, if we still consider (*OP*) supported attacks (*F*), and the same for (*W*) to (*M*), both (*M*) and (*F*) should not be accepted. If we consider (*OP*) mediated attacks (*F*), and the same for (*W*) to (*M*), (*F*) is accepted in the only preferred semantics.

4. Concluding remarks

Polberg and Oren [14] aim to unify some of the most popular approaches to the representation of support and to construct an unified environment capable of handling the available types of support, and we agree with them that doing so not only provides important theoretical contributions, but also helps in the representation of real world domains. Towards this goal, we contribute to theory with a principle-based analysis for bipolar argumentation framework semantics, and to practice with a case study in divorce action. Our case study reveals a number of issues to be addressed in future work.

First of all, once the support relations are interpreted, it seems that we need to consider bipolar argumentation with multiple support relations $BAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle$ and adapt the definitions accordingly. Moreover, for the interpretation of the support relations, we can define new combinations. For example, it is quite natural for a support to be both deductive and evidentiary, or to be both necessary and evidentiary. This can be achieved by combining definitions 2 and 3 in the following way. We first add indirect attacks using the construction of definition 2, and then we consider evidentiary semantics instead of Dung semantics. In other words, the support relation is used twice, first to define indirect attacks, and then again to define e-admissible sets, and so on. This would lead to six more rows in Table 1, for each of the reductions now combined with evidentiary support, leading to a total of 13 rows. Such combined interpretations of support can be characterized by new principles combining on the one hand closure and inverse closure, characteristic for deductive and necessary support, and on the other hand grounded or global support. For example:

P8. Self-supported Closure For all $BAF = (\mathcal{A}, \mathcal{R}, \mathcal{S})$, for all extensions E in $\mathcal{E}_{\mathcal{S}\mathcal{R}}$, $\forall a, b \in \mathcal{A}$, if $a\mathcal{S}a$, $a\mathcal{S}b$ and $a \in E$, then $b \in E$.

P9. Self-supported Inverse Closure For all $BAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$, for all extensions E in $\mathcal{E}_{\mathcal{S}\mathcal{R}}$, $\forall a, b \in \mathcal{A}$, if $a\mathcal{S}a$, $a\mathcal{S}b$ and $b \in E$, then $a \in E$.

More generally, sometimes it seems to depend on the context whether a support is interpreted as deductive or necessary, like the interpretation of the support between (*OP*) and (*M*) in the running example. Moreover, in our experience in modeling the divorce action, the choice between deductive, necessary and evidentiary support is rather limited. This reinforces the results on the theory-based validation of existing kinds of support, where only for necessary support some results have been obtained [17,5,18]. Alternative approaches are proposed by Gabbay [7], Gottifredi *et al* [8] and Potyka [15]. The principle-based analysis suggests that there are many possible combinations which have not been explored yet within the realm of bipolar argumentation. We also found that in this example, the necessary supports didn't affect much the outcome: compared to the original framework, the result stayed the same. However, in such a symmetric, what is more, parallel case, where the two arguments constituting the dilemma by attacking each other are the same with different subjects, considering a support necessary on one side should result in an attack on the other: if knowing how to handle children is necessary on the mother side, it is on the father side too, which if he lacks, it should be added as an attack. This indirect attack nature of the necessary support is of course context-dependent, but its formal representation might be worth further investigation.

Second, we found that in modeling the divorce action, there are two ways to model support. On the one hand we can model both attacks and supports among arguments, as we did in this paper, but on the other hand we can also model all support within the arguments, and only attack among the arguments. It seems that in the former case, most authors see the need to generalize to sets of arguments attacking or supporting arguments, as in dialectical argumentation frameworks. Despite this apparently fundamental modeling choice, we found little help in the literature about the advantages and disadvantages of the two approaches. Also it is unclear to us in general whether we can translate one kind of model into the other and vice versa. Consider the scenario where the child is only 4 years old. (4) actually rather attacks the deductive support between (OP) and (M). Now consider that the attorney of the mother would like bring some arguments regarding the wealth difference between the parents. He might say: ”money is not everything”. What would this argument attack? It wouldn't attack the fact that the father is wealthy. What it would attack is the the deductive support relation between (W) and (F). Consider a scenario where the father is wealthy because he is an entrepreneur. This as an argument, or if we consider it as two arguments one deductively supporting the other, the support relation between them could be attacked by the mother's attorney's argument as ”entrepreneur is risky”. In this paper we didn't use a language for describing arguments attacking relations, but we think it might be fruitful research direction investigating comparing solutions.

Third, when defining our principles we found a lot of research on semantics and principles for complex forms with set attack and set support, and numbers representing the strengths and supports [1]. From a formal perspective, we believe that this research is very useful because it helps relate formal argumentation to other domains like multi-criteria decision making.

Fourth, what struck us in this research is the similarity between reductions for bipolar argumentation frameworks, and the for preference-based argumentation frameworks we studied in earlier work. Whereas in both frameworks, the support relation and the preference can be both added and removed. Moreover, we found that the theory of reductions for preference based argumentation and bipolar argumentation is closely related to dynamic principles for AF [19], and we expect that this can be a source of further principles. Finally, like in preference-based argumentation, we believe that bipolar frameworks with symmetric attacks can be studied as a fragment with good computational properties.

References

[1] Leila Amgoud and Jonathan Ben-Naim. Evaluation of arguments in weighted bipolar graphs. *Int. J. Approx. Reason.*, 99:39–55, 2018.

[2] Guido Boella, Dov M. Gabbay, Leendert van der Torre, and Serena Villata. Support in abstract argumentation. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 111–122. IOS Press, 2010.

[3] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning*, 54(7):876–899, 2013.

[4] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. An axiomatic approach to support in argumentation. In *International Workshop on Theory and Applications of Formal Argumentation*, pages 74–91. Springer, 2015.

[5] Andrea Cohen, Simon Parsons, Elizabeth I. Sklar, and Peter McBurney. A characterization of types of support between structured arguments and their relationship with support in abstract argumentation. *Int. J. Approx. Reason.*, 94:76–104, 2018.

[6] Phan M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[7] Dov M. Gabbay. Logical foundations for bipolar and tripolar argumentation networks: preliminary results. *J. Log. Comput.*, 26(1):247–292, 2016.

[8] Sebastian Gottifredi, Andrea Cohen, Alejandro Javier García, and Guillermo Ricardo Simari. Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. *Artif. Intell.*, 262:336–368, 2018.

[9] Farid Nouioua and Vincent Risch. Bipolar argumentation frameworks with specialized supports. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 215–218. IEEE, 2010.

[10] Nir Oren, Michael Luck, and Chris Reed. Moving between argumentation frameworks. In *Proceedings of the 2010 International Conference on Computational Models of Argument*. IOS Press, 2010.

[11] Nir Oren and Timothy J. Norman. Semantics for evidence-based argumentation. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30, 2008*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 276–284. IOS Press, 2008.

[12] Sylwia Polberg. Intertranslatability of abstract argumentation frameworks. Technical report, Cardiff University, 2017.

[13] Sylwia Polberg and Anthony Hunter. Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *Int. J. Approx. Reason.*, 93:487–543, 2018.

[14] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 369–376. IOS Press, 2014.

[15] Nico Potyka. Bipolar abstract argumentation with dual attacks and supports.

[16] Henry Prakken. *Logical tools for modelling legal argument: a study of defeasible reasoning in law*, volume 32. Springer Science & Business Media, 2013.

[17] Henry Prakken. On support relations in abstract argumentation as abstractions of inferential relations. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 735–740. IOS Press, 2014.

[18] Henry Prakken. On validating theories of abstract argumentation frameworks: the case of bipolar argumentation frameworks. In *Proceedings of the 8th Workshop on Computational Models of Natural Argument (CMNA 2020), Perugia, Italy (and online)*. CEUR-WS.org, 2020.

[19] Tjitze Rienstra, Chiaki Sakama, and Leendert van der Torre. Persistence and monotony properties of argumentation semantics. In *International Workshop on Theory and Applications of Formal Argumentation*, pages 211–225. Springer, 2015.

[20] Liuwen Yu and Leendert Van der Torre. A principle-based approach to bipolar argumentation. Available at https://nmr2020.dc.uba.ar/WorkshopNotes.pdf (2020/09/20).

This page intentionally left blank

Short Papers

This page intentionally left blank

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200865

Plain Language Assessment of Statutes

Wolfgang ALSCHNER,[a1] Daniel D'ALIMONTE,[a] Giovanni C. GIUGA[a] and Sophie GADBOIS[a]
[a] *University of Ottawa*

Abstract. Legislative drafters use plain language drafting techniques to increase the readability of statutes in several Anglo-American jurisdictions. Existing readability metrics, such as Flesch-Kincaid, however, are a poor proxy for how effectively drafters incorporate these guidelines. This paper proposes a rules-based operationalization of the literature's readability measures and tests them on legislation that underwent plain language rewriting. The results suggest that our readability metrics provide a more holistic representation of a statute's readability compared to traditional techniques. Future machine-learning classifications promise to further improve the detection of complex features, such as nominalizations.

Keywords. Plain Language, Readability, Rules-based approach, Flesch-Kincaid.

1. Introduction

Statutes often use Latin "legalese" or complex sentences that make text difficult to understand for non-experts [1]. In response, several Anglo-American jurisdictions have recently passed guidelines and laws that require statutes to be written in "plain language".[2] These reforms built on decades of scholarly work that emphasizes the need for legal drafters to employ shorter, simpler sentences, use ordinary words in their normal sense and write in the active voice [2][3]. However, there is a disconnect between the principles developed in plain legal language laws, guidelines, or scholarship and the operationalization of legal readability checks in practice. While the former creates rules *specific* to the legal domain, the latter employs *generic* metrics developed outside of the legal context to assess readability, such as Flesch-Kincaid (FK) scores. North Carolina, Florida, and Oregon, for example, have enacted legislation that requires government documents to meet a minimum FK score.[3]

Using simplistic general-purpose metrics on statutes, such as FK scores that assess readability by counting syllables per words and words per sentences, is problematic for several reasons. First, peculiarities of legal texts, such frequent cross-references, lists, or unusual punctuation conventions, skew measures like FK, which require the clear sentence boundaries found in prose. Second, general-purpose metrics, at best, only indirectly capture plain legal language recommendations and, at worst, may be negatively correlated with them, *e.g.* when replacements of legalese with wordier ordinary terms reduce FK scores. Third, beyond using shorter words and sentences,

[1] Corresponding Author, Associate Professor, Common Law Section, University of Ottawa, Email: wolfgang.alschner@uottawa.ca. We are grateful to Peter Zachar for his input on earlier versions of this project.

[2] For example, the Government of Canada's Department of Justice requires that new legislative texts follow its *Legistics* guidelines, the United States' *Plain Writing Act of 2010*, the United Kingdom's *The Good Law Initiative*, and the European Union's *Better Regulation Agenda*.

[3] NC Gen Stat § 58-66-30(b) (2013); FL Stat § 627.4145 (2019); OR Rev Stat § 316.364 (2019).

generic metrics, like FK, fail to provide specific guidance to drafters on how to write more readable texts [4]. This contribution seeks to remedy this disconnect by operationalizing legal plain language guidelines for statutory readability. Our metrics describe different lexical, grammatical, stylistic, and structural properties of statutory texts that, according to the plain language literature, make legal texts more readable. To validate our measures, we test them against an original dataset of before-and-after plain language rewrites from five Anglo-American jurisdictions. The results illustrate that our metrics offer a more holistic understanding of readability compared to traditional measures, but also point to the need for future research to combine rules-based and machine-learning approaches to devise readability measures specific to the legal domain.

2. Methodology

We systematically reviewed the recommendations developed in English-speaking plain language scholarship and drafting guidelines.[4] We then ranked recommendations by their frequency to identify drafting principles that enjoy widespread support across Anglo-American jurisdictions. Finally, among the top-ranking principles, we focused on those that are difficult to evaluate manually.[5] Based on these considerations we operationalized guidelines through a rules-based approach that detects a set of lexical, grammatical, stylistic and structural properties of statutory texts summarized in Table 1.

Table 1. Readability Metrics Developed from Plain Language Guidelines and Scholarship

Metric Type	Explanation
Lexical	**(1) Shall/must:** searches for instances of "shall", "shan't", "must", and "mustn't". **(2) Legalese:** calculated based on the occurrence of a subset of terms from the Black legal dictionary that are classified as non-English by the hunspell R-package dictionary spell check function and thus likely either Latin or legalese, *e.g.* "offeror".
Grammar	**(1) Compound phrases:** counted when coordinating conjunctions identified by a POS tagger and common independent marker words are recognized. These terms typically denote when clauses have been combined, which can create complex and wordy sentences. **(2) Conditional phrase:** counts sufficient and standalone core conditional indicators. Dual use conditionals, such as "any", are not considered to limit overcounting. Sufficient and necessary conditionals are not separately counted to avoid double counting paired conditionals like "if-then". **(3) Nominalizations**: counted by identifying words with common nominalization endings that are not proper nouns, legalese or statute-specific words (*e.g.* "provision"). To limit overcounting, only problematic nominalizations that are paired with the passive voice or that are preceded by a proposition and thus contribute to wordiness are counted.
Style	**(1) Passive voice:** looks for conjugations of "to be" and verbs with the past participle tense POS. Once all indices are located, a custom made matching function is employed to determine the closest past participle match to a "to be" conjugation given the "to be" conjugation must occur first, the matches have to be within a user specified proximity of one another, and the matches have to be unique with preference given to the closer matching pair. Matching pairs are counted as instances of the passive voice. **(2) All-caps**: checks whether words are written in all caps using regular expressions.
Structure	**Counts words, sentences, and syllables**. Texts are preprocessed to improve sentence boundary detection by eliminating external references with problematic punctuation, list elements and numerical characters.

[4] To this end, we reviewed 34 plain legal language textbooks, journal articles and official drafting guidelines from several Anglo-American jurisdictions.

[5] For example, plain language scholars recommend that statutory texts contain a table of content and are written in active voice. While the former can be easily evaluated by a human reviewer, instances of passive voice are more difficult to detect efficiently.

3. Validation

To validate how well our metrics identify plain language text characteristics, we created an original dataset of statutes from five Anglo-American jurisdictions in two versions: the originally enacted legislation ("before") and a plain language rewrite ("after"). The dataset's "after" legislation includes texts (1) written by academics but not enacted (Equality Act, Takeover Codes, Timeshares Act) and (2) enacted by a government to replace legislation (Minneapolis City Charter and Contract & Commercial Law Bill).

Table 2. Before-after Legislation Used in Dataset

Piece of Legislation	**Jurisdiction**
Promotion of Equality and Prevention of Unfair Discrimination Act ("Equality Act") Section 12	South Africa
Timeshares Act	United Kingdom
Contract and Commercial Law Bill	New Zealand
Minneapolis City Charter	United States
Takeover Codes	Australia

Validation proceeded in two stages. The first stage of validation consisted of iteratively comparing human and automated feature identification results for a sample of 10% of every legislation to identify errors and to refine our identification rules. As expected, the simple shall/must, total number of words, and all caps functions performed well during this initial validation. Sentence counts were initially problematic due to incorrect sentence boundary detection, but these errors could be addressed by eliminating confounding punctuations through text pre-processing. The identification of legalese was improved by stemming words in the text and the dictionary before checking for matches, which helped detect small variations of the same term. Limitations of a rules-based identification emerged most clearly with the more complex nominalization, compound phrase, and conditional phrase metrics, where our metrics, even after refinements, approximated but did not perfectly match manual feature detection (see Section 4).

Once we were confident that our metrics captured the most common categories of compound phrases, conditionals, and nominalizations we encountered during our sampling, we proceeded to the second stage of the validation to compare the before-after texts across our metrics. The results reproduced in Table 3 validate that our metrics succeed in tracking changes between the versions. The shall/must measure show a significant decrease of shall and concomitant increase of must in the plain language rewrite. In addition, the legislations' plain language versions use fewer compound phrases, nominalizations, less passive voice and fewer total words and legalese compared to the original versions. Our metrics thus capture plain language text modifications.

4. Discussion and Limitations

Our metrics offer a more nuanced representation of a statute's readability compared to FK scores and help drafters to review or rewrite statutes based on plain language criteria.

Table 3. Original and Plain Language Legislation's Readability Measures Scores

	Minneapolis City Charter		**NZ Commercial Bill**		**SA Equality Act s.12**		**AU Take-over Codes**		**UK's Time-shares Act**	
Metric	Orig.	PL	Orig.	PL	Orig.	PL	Orig.	PL	Orig.	PL
Shall	1763	0	283	0	0	0	155	0	32	0
Must	8	121	25	124	0	0	0	106	7	14
Compound Phrases	4976	837	2095	1883	8	4	1668	792	168	135
Conditional Phrases	445	122	384	509	0	0	249	198	28	38
Nominalization	490	83	412	354	1	0	491	116	82	62
Passive Voice	1138	74	652	609	3	0	716	237	41	29
Total Words	65554	12865	35066	33523	73	40	31635	13764	3531	2600
Sentence Number	1188	676	520	780	1	3	282	267	70	86
Syllables per 100 words	161	176	157	158	211	188	162	161	163	165
Legalese	186	46	159	125	1	0	317	216	50	3
All Caps	6	0	0	0	0	0	3	2	0	0

At the same time, our approach comes with limitations. First, since plain language guidelines and the formatting of statutory texts vary across jurisdictions, our rules likely require adaptation for different jurisdictions. Second, our rules-based approach could be complemented with machine learning to correctly identify more complex text features. While some of the most prominent plain language guidelines lend themselves to a simple rules-based implementation (*e.g.* shall/must), more complex features require a more nuanced approach. Rules-based detection of nominalizations based on typical word endings, for example, leads to overcounting words that have nominalization endings but do not have verbs as root words ("business") and valid nominalizations that are not used in problematic ways ("information"). Devising rules that comprehensively capture this distinction is challenging. Human expert labelling of problematic nominalizations scaled through machine learning provides an alternative avenue for identifying relevant text features. For example, Sugisaki has shown how machine learning classifiers can identify text parameters including complex noun phrases in legal texts [5]. Future research could thus combine rules-based approaches that tackle low-hanging fruit (*e.g.* shall/must) with machine learning for more challenging feature identification tasks. In combination, they provide a scalable means to operationalize plain language assessments of statutes.

References

[1] Gail S Dykstra. Plain Language, Legal Documents and Forms: Background Information. *Can L Information Council*. 1987.

[2] Michael EJ Masson, Mary Anne Waldron. Comprehension of Legal Contracts by Non-Experts: Effectiveness of Plain Language Redrafting. 8 (1994) *Applied Cognitive Psychology* 67.

[3] Peter Butt, Richard Castle. *Modern Legal Drafting: A Guide to Using Clearer Language*, 2nd ed. Cambridge University Press. 2006.

[4] Crossley, Scott A., Stephen Skalicky, and Mihai Dascalu. Moving beyond Classic Readability Formulas: New Methods and New Models. 42 (2019) *Journal of Research in Reading* 541.

[5] Kyoko Sugisaki. Towards Data-Driven Style Checking: An Example for Law Texts. In: Florian Bex and Serena Villata (eds.). *Frontiers in Artificial Intelligence and Applications. 29th International Conference on Legal Knowledge and Information Systems (JURIX)*. 2016. p. 93-100.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200866

Permissioned Blockchains: Towards Privacy Management and Data Regulation Compliance

Paulo Henrique ALVES [a], Isabella Z. FRAJHOF [b], Fernando A. CORREIA [a], Clarisse DE SOUZA [a] and Helio LOPES [a]
[a] *Department of Informatics, PUC-Rio, Brazil*
[b] *Law Department, PUC-Rio, Brazil*

Abstract. Data privacy and protection has been a trending topic in recent years. The COVID 19 pandemic has brought about additional challenges and tensions. For example, sharing health data across several organizations is crucial for significant control and reduction of massive infection and death risks. This implies the need for broadly collecting and using personal and sensitive data, which raises the complexity of data protection and privacy challenges. Permissioned blockchain technology is one way to empower users in controlling how their data flows through the net, in a transparent and secure way, through an immutable, unified, and distributed database ruled by smart contracts. Given this background, we developed a second layer data governance model for permissioned blockchains based on the Governance Analytical Framework principles to be applied in pandemic situations. The model has been designed to organize the relationship between data subjects, data controller, and data processor. Regarding privacy concerns, our proposal complies with the Brazilian General Data Protection Law.

Keywords. privacy, governance, blockchain, regulation, public health

1. Introduction

Data privacy and data protection became one of the most critical concerns in the digital era. In order to regulate how data can be collected and used, many data protection regulations emerged to set rules to organize this environment. These kinds of regulations aims to provide rights and duties for both users and companies, whenever the processing of personal data is taking place. Thus, data protection norms also applies, and are extremely important in this scenario, when the processing of sensitive health data is taking place.

Previous pandemic outbreak experiences like influenza, MERS-CoV, Zika, Ebola[1], and now COVID-19, showed that data sharing between health institutions and other stakeholders worldwide is fundamental to fight against the broad contamination. Moreover, considering the intensive collection of personal data in these scenarios, abiding to

[1] Data Sharing in Public Health Emergencies. Available at: `https://www.glopid-r.org/wp-content/uploads/2019/07/data-sharing-in-public-health-emergencies-yellow-fever-and-ebola.pdf` Accessed at: 10/15/2020.

data protection norms is of the utmost importance [1]. It must be noted that data protection regulations do not forbid the use of personal data in a pandemic scenario, but establishes the rules and legitimate uses that must be observed. Such compliance provides that society can benefit from the uses of such data: it protects individual's privacy and data at the same time as it allows for data utility. In this sense, contact tracing apps [2] are being implemented as a manner to allow public health institutions to track the infection movement and potentially infected people.

Just recently in Brazil, a Data Protection Regulation was enacted (Law n. 13.709/2018, Lei Geral de Proteção de Dados Pessoais – LGPD). Due to Brazilian lack of tradition in this subject, it is important to provide society acculturation and awareness of the importance of protecting personal data in general. Furthermore, such regulation sets rules and obligations that regulates the use of personal data by public and private entities. Thus, in the pandemic scenario, controllers and processors must evaluate which of the legal basis foreseen in law authorizes the collection of users' data. It must be remarked that the LGPD establishes that individual consent is only one of the legal basis authorizing data processing. In any case, data controllers must abide to the law's principles, rights, safeguards and act in good faith. From a technology perspective, data privacy management is challenging. Data must be processed and kept in a safe ruled-base environment, and looks forward to a transparent and secure environment [3].

Therefore, we proposed a second layer of governance in permissioned blockchains solutions, since only the first layer, i.e., platform governance (permissioned or permissionless), is not able to address this challenge. We developed an architecture based on Hyperledger Fabric[2] to instantiate the proposed governance in the COVID-19 pandemic scenario. We base our model on the Governance Analytical Framework (GAF) [4] principles defining the Problem (such as the purpose limitation), Actors (data subject and data controller and processor), Social Norms (regulations), Process (data processor methodologies), and Nodal Points (technology used to connect stakeholders).

2. Related Work

Contact tracing apps are also useful data sources for disease contamination tracking. The DP-3T initiative [5] uses the Bluetooth signal to identify infected people or people who have been in touch with someone who was infected. Such applications are controversial solutions from both privacy and medical viewpoints: not only highlights the infected person but also who has been in touch with him/her and, from the medical perspective, at least 60% of the population should have the app installed in order to such solution be effective. Therefore, to preserve the user's privacy all the data should be anonymized and decentralized. Even though the authors in [6] proposed a blockchain-based application for electronic medical records management, they did no association with any data regulations. Panian [7] argues that companies and government organizations should define standards, policies, and data management processes. The author presents application-centric and process-centric models for data governance. However, those models do not present the concerns related to privacy and data management.

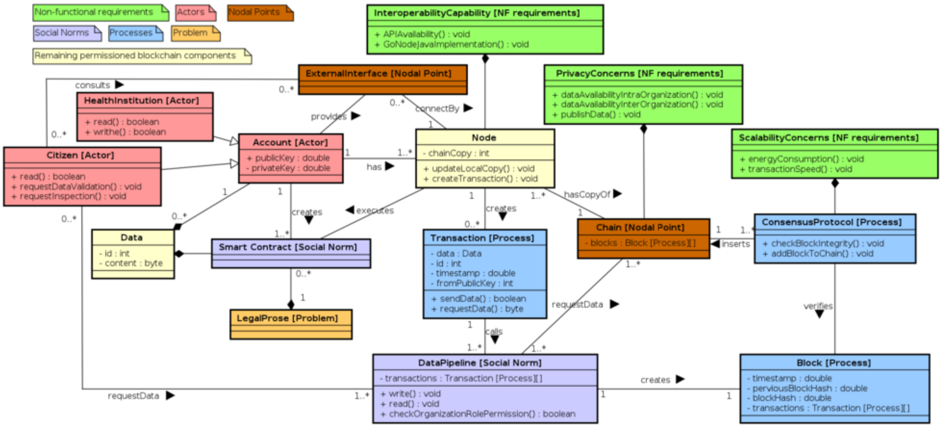
[2] Hyperledger Fabric. Available at: `https://www.hyperledger.org/use/fabric` Accessed at: 10/15/2020

As one could observe, the presented works showed essential concepts and applications regarding personal data collection and management. However, the combination of privacy management, governance model concepts and the usage of blockchain technology application to provide a safe environment for data sharing has not been explored yet. Therefore, our proposal of a second layer data governance for permissioned blockchain cames to offer a complete environment for data sharing and privacy management.

3. Blockchain Data Governance

To model the COVID-19 data governance scenario, we based our approach on the Governance Analytical Framework (GAF) [4]. The GAF is based on five principles: (i) problems, (ii) actors, (ii) social norms, (iv) processes, and (v) nodal points. This framework proposes deconstructing social problems by decomposing them on these five principles and reconstructing them by modeling the governance. This mapping helps people to identify the purpose of limitation accurately by verifying the Problem principle. The actors and norms involved can also be checked, so people are able to trigger, or even suite, the organization that broke any user's rights. Moreover, by checking the processes and nodal points, people can request how they were collected and processed. From the traceability perspective, the contact tracing apps can be modeled by the GAF principles as well.

Figure 1. GAF implementation in a permissioned blockchain architecture.


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults ▶
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
Account [Actor]
+ publicKey : double
- privateKey : double
provides ▶
connectBy ▶
has ▶
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
hasCopyOf ▶
creates
◀ executes
creates
Data
- id : int
- content : byte
Smart Contract [Social Norm]
LegalProse [Problem]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
◀ inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
requestData ▶
verifies
calls
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates ▶
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]
requestData ▶
0..*
1
1..*

This mapping should also help health institutions to not only to elaborate explanation regarding which data will be collected, in which scenarios and range, but also to guarantee data anonymization. Permissioned blockchains fit with all the presented concepts for allowing the creation of governance rules to manage entities and data. Figure 1 depicts the GAF definitions applied to this technology. Therefore, such technology can be used to store and share pandemic data, not only as a transparent link between data subjects, data controllers, and processors, but also as a data tracker and data provider to people or any other interested entity. Through permissioned blockchain, data can be audited and used as a data source for research purposes. Self-enforcement Blockchain Smart Contracts (BSC) enhances trust between the data subject and the data controller and processor. They guarantee: (i) self-execution and adherence of the purpose of the data processing, (ii) the historical information of the source of the collected data, who has accessed the data, thus, with whom such data was shared, and (iii) the timestamp.

Hence, BSC plays a vital role in this environment; it is responsible for roles assignment and can be used as a snapshot of activated norms in a specific moment. It also ensures transparency related to the dataflow. In this sense, differently from the presented literature, the proposed governance, detailed in [8], enables institutions to share data following previously agreed rules. The data provenance is available for citizens, researchers, government, and health institutions, which may improve the identification of data inconsistency worldwide by information comparison.

4. Conclusions

In this paper, we proposed a new data governance for privacy management in the permissioned blockchain platforms applying the GAF principles in the COVID-19 outbreak scenario. The LGPD rules guided our development towards compliance with data protection regulations. This technology is promising to support the data subjects by providing a transparent tool so that data subjects can confirm if their data was processed in accordance with data controllers' privacy policy. Also, permissioned blockchains, besides empowering data subjects, allow data controllers and processors to be accountable for their data processing activities. Like a fingerprint, the timestamp, combined with historical blocks, shall provide resources to reconstruct the data subject concessions over the law evolution. In this sense, this topic should be carefully evaluated to analyze the blockchain capability to be adherent to the legal basis and their advancement. Furthermore, the evaluation of different cryptography methods may contribute to data privacy and protection concerns.

References

[1] Bradford, L. R., Aboy, M., and Liddell, K., "Covid-19 contact tracing apps: A stress test for privacy, the gdpr and data protection regimes," *Journal of Law and the Biosciences*, 2020.

[2] van Kolfschooten, H. and de Ruijter, A., "Covid-19 and privacy in the european union: A legal perspective on contact tracing," *Contemporary Security Policy*, pp. 1–14, 2020.

[3] Karaçam, D. A., "Privacy and monopoly concerns in data-driven transactions.," in *JURIX*, pp. 145–150, 2019.

[4] Hufty, M., "Investigating policy processes: the governance analytical framework (gaf)," *Research for sustainable development: Foundations, experiences, and perspectives*, pp. 403–424, 2011.

[5] Fagherazzi, G., Goetzinger, C., Rashid, M. A., Aguayo, G. A., and Huiart, L., "Digital health strategies to fight covid-19 worldwide: challenges, recommendations, and a call for papers," *Journal of Medical Internet Research*, vol. 22, no. 6, p. e19284, 2020.

[6] Ekblaw, A., Azaria, A., Halamka, J. D., and Lippman, A., "A case study for blockchain in healthcare:"medrec" prototype for electronic health records and medical research data," in *Proceedings of IEEE open & big data conference*, vol. 13, p. 13, 2016.

[7] Panian, Z., "Some practical experiences in data governance," *World Academy of Science, Engineering and Technology*, vol. 62, no. 1, pp. 939–946, 2010.

[8] Alves, P. H., Frajhof, I. Z., Correia, F. A., de Souza, C., and Lopes, H., "Second layer data governance for permissioned blockchains: the privacy management challenge," 2020.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200867

Judges Are from Mars, Pro Se Litigants Are from Venus: Predicting Decisions from Lay Text

Karl BRANTING, [1] Carlos BALHANA, Craig PFEIFER, John ABERDEEN, and Bradford BROWN
The MITRE Corporation, McLean VA, USA

Abstract. Access to justice could be significantly expanded if decision support systems were able to accurately interpret statements of fact by pro se (self-represented) litigants. Prior research, which has demonstrated that case decisions can often be predicted by machine-learning models trained on judges' statements of facts, suggests the hypothesis that these same learning algorithms could be effectively applied to pro se litigants' fact statements. However, there has been a dearth of corpora on which to test this hypothesis. This paper describes an experiment testing the ability to predict the outcome of pro se litigants' complaints on a corpus of 5,842 cases initiated by citizen complaints. The results of this experiment were strikingly negative, suggesting that fact statements by unguided pro se litigants are far less amenable to simple machine-learning techniques than judges' texts and appearing to disconfirm the hypothesis above.

1. Introduction

In many nations across the world, access to justice is increasingly elusive for the majority of citizens who are not wealthy [1] [2]. In the United States, for example, "more than 80 percent of people living below the poverty line and a majority of middle-income Americans receive no meaningful assistance when facing important civil legal issues, such as child custody, debt collection, eviction, and foreclosure" [3]. A widely-acknowledged factor in this inaccessibility is the complexity of legal rules and procedures. However, an equally important factor is the gap between the ordinary parlance used by laypersons and the specialized terminology and usage of legal discourse. This linguistic gap creates challenges both for decision support on behalf of pro se (self-represented) litigants and for decision support for the adjudicators who must handle the claims of litigants unfamiliar with legal language. Even when legal rules and procedures can be formalized in computer-interpretable and executable form, it is typically a formidable challenge to elicit case facts in language compatible with those rules and procedures.

This paper describes experiments in which techniques previously used to predict decisions from statements of facts in published decisions were applied to texts written

[1] Corresponding Author: E-mail: lbranting@mitre.org. The MITRE Corporation is a not-for-profit company, chartered in the public interest. This document is approved for Public Release; Distribution Unlimited. Case Number 20-2066. ©2020 The MITRE Corporation. All rights reserved.

by citizen complainants. The results of these experiments were strikingly negative, suggesting that a different approach is needed for eliciting and interpreting fact statements produced by pro se litigants.

2. Predicting Decisions from Complainant Texts

We gained access to a Complaint Data Set consisting of 5,842 attorney misconduct complaints processed by the bar association of a US state (*Bar Association*). A key step in the Bar Association's handling of these complaints is an initial determination of whether the case should be forwarded for full investigation or whether instead the case can be closed before investigation (*CBI*) because it fails to state a *prima facie* ("colorable") claim.

Each case in the Complaint Data Set consisted of information submitted by the complainant through an online complaint form, together with metadata including the following: the history of prior complaints filed against the attorney to whom the complaint was directed (the *respondent*); the legal services to have been provided by the respondent to the complainant; and allegation codes, which correspond to provisions of the state's code of professional responsibility and statutes regarding attorney misconduct and which are manually assigned by staff based on a reading of the complaint text at intake. The complainant information included the names of the complainant and respondent attorney or attorneys, a free-text description of the events justifying the complaint, a separate free text description of the relationship between the complainant and the respondent attorney (the "connection text"), and other information not relevant to the merits of the case. We supplemented this feature set with readability features, including Flesch Reading Ease and SMOG Index,[2] and mean per-sentence sentiment [4]. Each case was labeled as to whether it was closed before investigation or was investigated further. The categories were relatively balanced, with 55.65% closed at intake.

Our initial experiment explored how accurately CBI decisions could be predicted based on information available to the intake staff at the time the complaint was submitted. The complaint texts[3] were normalized by removing newlines and replacing each person's name with the token PERSON using the Stanford Named Entity Extraction (NER) tool.[4] We tested two feature representations for the texts:

- N-gram frequency vectors, for n = 2–4
- Vectors of 250 topic models[5]

We compared the performances of six machine-learning algorithms—Naive Bayes, Bayes Net, SMO, JRip, J48, and Random Forest—in 10-fold cross validation. The results of the highest-performing algorithms are shown in Table 1. Disappointingly, performance in predicting CBI decisions was only slightly higher than chance regardless of representation or algorithm. This result contrasts with the much better results obtained in other domains from text written by attorneys or judges, e.g., [5]. We hypothesize that the highly discursive, irregular, and inconsistent character of complaint texts is responsible for the much-lower predictive accuracy.

[2] https://pypi.org/project/readability/

[3] We appended the connection text, if any, to each complaint text.

[4] https://nlp.stanford.edu/software/CRF-NER.html

[5] The topic models were constructed using gensim (https://radimrehurek.com/gensim/about.html).

Features	Mean MCC	Frequency-weighted mean F1	Algorithm
n-gram frequency vectors	0.023	0.521	SVM
250 dim topic vector (gensim)	0.010	0.425	BayesNet

Table 1. The accuracy of decision predictions based on complaint text features.

Features	Mean MCC	Frequency-weighted mean F1
Case metadata only	0.116	0.525
Case metadata plus n-gram frequency vectors	0.153	0.551

Table 2. The accuracy of decision predictions based on a BayesNet model trained on metadata features, with and without complaint text.

Features	Mean MCC	Frequency-weighted mean F1
Allegation codes	0.376	0.695
Allegation codes plus text	0.376	0.695
Allegation codes plus metadata	0.377	0.696

Table 3. Prediction results based on a BayesNet model trained on allegation codes with and without text and metadata features.

We next evaluated the predictive value of the case metadata, which consisted of (1) non-narrative information provided by the complainant in the online form, (2) information from the Bar Association attorney database, (attorney history and prior complaints), and (3) the sentiment scores and various readability metrics calculated from the complaint texts.

Table 2 shows that the case metadata is somewhat more predictive of CBI decisions than complaint texts, and the combination of metadata and complaint texts is slightly more predictive than either individually, but even in combination these features are only weakly predictive of the CBI decisions.

We next explored the degree to which CBI decisions could be predicted after the intake staff had assigned allegation codes to each case, which occurs before the decision whether to send the case forward for investigation. As shown in Table 3, allegation codes standing on their own have moderate predictive value, with the MCC of 0.376 indicating that more than $1/3$ of the uncertainty about a complaint is eliminated if the allegation codes are known. Adding the text features didn't reduce the uncertainty further, indicating that the allegation codes capture most of the relevant, predictive information in the complaint text. Combining the allegation codes with metadata increases predictive accuracy by a very small amount.

This experiment indicated that allegation codes have a moderate predictive value for CBI decisions (an MCC of 0.376), so we turned to the question whether we could predict allegation codes from complaint texts. Assigning allegation codes to each new case is time-consuming for intake staff, so automating this process could be a useful form of decision support on its own, apart from helping identify cases that are likely to be closed on intake. We evaluated performance accuracy on prediction of the 10 most frequent allegation codes based on an n-gram representation of complaint texts, which collectively cover approximately 60% of all complaints. Only one allegation code could be predicted with an MCC greater than 0.15, meaning that predictive accuracy for most allegation codes was only slightly higher than chance. We attempted to develop an annotation scheme for complaint texts so that we could apply the methodology described in [6] to the corpus, but the complaints' extreme variability and disorganization frustrated these annotation efforts.

In our view, these experimental results show that decision support systems that fail to support pro se litigants in expressing facts relevant and necessary for a claim create a high barrier to accomplishing the subsequent task of assessing whether the assertions state a prima facie case. the root problem is that pro se litigants seldom know what facts they need to establish or how to articulate and organize the facts in a manner that makes their claims amenable to evaluation.

In summary, our experiments with pro se complaint texts failed to replicate the predictive accuracy that we and others observed in previous work predicting decisions from judges' and other adjudicators' fact statements. We surmise that the characteristics of judges' and other adjudicators' language, including stylistic consistency and regularity, are critical to the ability of current machine-learning techniques to induce accurate predictive models from the statements of facts in published decisions.

3. Conclusion

This paper has presented an experiment in which the predictive accuracy previously demonstrated from judges' statements of facts could not be reproduced on fact statements written by pro se complainants. These results suggest that judges' statements of facts are a poor proxy for pro se litigants' narrative texts and that techniques suitable for prediction from judges' texts may not be appropriate for decision support for pro se litigants. We believe that a promising research direction is development of narrative elicitation techniques based on recent work on narrative schema induction [7]. Such techniques could help bridge the gap between the language of judges and the language of pro se litigants, which our experimental results suggest are as remote from one another as Mars is from Venus.

References

[1] Hadfield G. Rules for a Flat World: Why Humans Invented Law and How to Reinvent It for a Complex Global Economy. Oxford University Press; 2016.

[2] Himonas D, Hubbard T. Democratizing the Rule of Law. Stanford Journal of Civil Rights & Civil Liberties. 2020;16(2):261–282.

[3] Perlman AM. The Public's Unmet Need for Legal Services & What Law Schools Can Do about It. Daedalus. 2019;148(1):75–81.

[4] Hutto C, Gilbert E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media; 2014. p. 00–00.

[5] Chalkidis I, Androutsopoulos I, Aletras N. Neural Legal Judgment Prediction in English. CoRR. 2019;abs/1906.02059.

[6] Branting LK, Pfeifer C, Brown B, Ferro L, Aberdeen J, Weiss B, et al. Scalable and explainable legal prediction. Artificial Intelligence and Law. 2020:1–26.

[7] Belyy A, Van Durme B. Script Induction as Association Rule Mining. In: Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events. Online: Association for Computational Linguistics; 2020. p. 55–62.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200868

Evaluating the Data Privacy of Mobile Applications Through Crowdsourcing

Ioannis CHRYSAKIS [a,b], Giorgos FLOURIS [a], George IOANNIDIS [c],
Maria MAKRIDAKI [d], Theodore PATKOS [a], Yannis ROUSSAKIS [a],
Georgios SAMARITAKIS [a], Alexandru STAN [c], Nikoleta TSAMPANAKI [a],
Elias TZORTZAKAKIS [a], and Elisjana YMERALLI [a]

[a] *FORTH, Institute of Computer Science, Greece*
[b] *IDLab, Dept. of Electronics and Information Systems, UGent, imec, Belgium*
[c] *IN2 Digital Innovations GmbH, Germany*
[d] *FORTH, PRAXI Network, Greece*

Abstract. Consumers are largely unaware regarding the use being made to the data that they generate through smart devices, or their GDPR-compliance, since such information is typically hidden behind vague privacy policy documents, which are often lengthy, difficult to read (containing legal terms and definitions) and frequently changing. This paper describes the activities of the CAP-A project, whose aim is to apply crowdsourcing techniques to evaluate the privacy friendliness of apps, and to allow users to better understand the content of Privacy Policy documents and, consequently, the privacy implications of using any given mobile app. To achieve this, we developed a set of tools that aim at assisting users to express their own privacy concerns and expectations and assess the mobile apps' privacy properties through collective intelligence.

Keywords. data privacy, mobile apps, GDPR, crowdsourcing, collective intelligence

1. Introduction

We experience a massive increase in personal information utilised by smartphone applications (apps), whose invasive nature for harvesting personal data has been demonstrated in many studies. This trend is continuing, despite the recently-established legislation for personal data protection, such as CCPA (California), LGPD (Brazil) and GDPR (Europe). In fact, studies have shown that the level of compliance of organizations and businesses to GDPR is low[1]. Although tracking and data access by apps is often legitimate, users are unaware of the related privacy risks, because apps describe their privacy behavior in a vague Privacy Policy (PrP) document, which is typically written using legal language and terminology [1], in long and frequently changing documents[2], making it hard for users to read and understand the critical aspects related to their privacy. Thus,

[1] See: `https://gdpr.report/news/2019/07/22/almost-a-third-of-eu-firms-still-not-gdpr-compliant/`, `https://symantec-enterprise-blogs.security.com/blogs/expert-perspectives/gdpr-turns-1-many-companies-still-not-ready`

[2] `https://www.varonis.com/blog/gdpr-privacy-policy/`

it comes as no surprise that the typical consumer is not investing time in studying such documents before agreeing, thus unintentionally granting permission to apps to access, use, and share a wealth of personal information, in a manner unknown to the user.

In this paper, we present the CAP-A H2020 project[3], which aims to *support users in the daunting task of understanding the content of a PrP document and to be aware of the privacy implications of using any given mobile app*[4].

Our position is that technical solutions and legal regulations are necessary but not fully sufficient for accomplishing a paradigm shift; at the heart of our solution is the hypothesis that data protection can also be powered by the society itself. By mobilising consumers to become active players, we can harness our collective power, leading to a more ubiquitous adoption of the technical and regulatory frameworks. To protect privacy adequately, society needs awareness, but also consensus about privacy protecting measures and processes that generate norms, with which service providers will voluntarily comply because it is profit maximising [2]. Exploring this knowledge is also of value to social scientists to better understand the community dynamics involved, as well as to policy makers to design more accurate and timely policies.

Along these lines, CAP-A deploys ICT tools that facilitate community interaction and co-creation in various ways that improve users' privacy awareness, and support a more efficient interaction among developers and end users; the latter will lead to a new innovation model that will allow consumers to collectively express their concerns, and developers to adopt more privacy-friendly practices and to better respond to market needs. CAP-A will also help in identifying and highlighting differences in opinions (i.e., norms), in a way that will be beneficial for users, developers, social scientists and policy makers.

2. The CAP-A portal and mobile app

The CAP-A portal is a responsive web page, whereas the mobile app offers additional functionalities adapted for small screens. They both rely on the same backend (which uses data stored using semantic technologies) and are available for public use in: `https://www.cap-a.eu/tools`. Due to space restrictions, we provide a brief description of the most important functionalities of the CAP-A portal and mobile app below[5].

Expectations. Through CAP-A, users can *express expectations*, i.e., whether they consider (or not) reasonable a certain data request on behalf of the developer. Each expectation is related to a certain privacy-related process, such as “access to camera”, “minimisation of data collected” etc (called *Privacy Policy Practice* or *PPP* for short).

PrP annotator. CAP-A allows users to *annotate PrP documents* of apps, by marking a block of text in the PrP document and stating the relevance of this block to a certain PPP. Annotations are meant to highlight the important blocks of text in a PrP document and how they are related to PPPs, thereby simplifying the task of understanding its content.

Sharing evidences. Users can *share evidences* related to an app, which may be online articles, grounded claims by people who tested the app, or official documentation regarding its privacy properties. The credibility of such evidences is assessed by users.

[3] `https://cap-a.eu/`, funded by NGI_Trust, and implemented by the authors

[4] In the context of this paper, the term PrP refers to any type of Privacy Policy, Terms of Use, Consent Form etc document prescribing legally binding obligation on behalf of a developer concerning a particular app.

[5] Similar info, with screenshots, can be found in [14].

The mobile app. The *CAP-A mobile app* is a native Android app, which is not just a mobile-friendly version of the portal, but also allows users to conduct an "audit" of their installed apps, which allows targeted retrieval of information from Google Play.

Gamification and rewarding. *Gamification features* based on *rewarding mechanisms* are a well-known tool to support sustaining communities and for motivating contributors [10]. The CAP-A rewarding mechanism was developed using a general-purpose ontology [13], which captures various common features of diverse reward schemes. It encapsulates well-known gamification principles ([11]) and employs both intrinsic and extrinsic rewards ([12]).

App ratings. Each app in CAP-A is associated with two *privacy-related ratings*. These ratings are the *Satisfaction of Community's Expectations*, which measures how close the privacy expectations regarding the app (as expressed by the users) are to what the app is requesting, and the *Privacy Friendliness* rating, whose computation takes into account privacy-related best practices, such as easy-to-understand PrP documents. The calculation of an app's ratings is based on a set of weighted functions and parameters that aim to ensure an intuitive and fair behaviour.

Browsing apps. An easy-to-use *search and browsing facility for apps* is provided to allow users to access the app-related information (e.g., expectations, annotations, app ratings, evidences etc.). For legal reasons, only public information is shown. Moreover, not all apps found in Google Play have been downloaded; instead, the system automatically downloads data on the apps most relevant for its users.

The Privacy Dashboard. In the *Privacy Dashboard*, users can find visual representations of aggregated information about users and apps, as well as an aggregation of users' behavior in the form of privacy norms. For example, we can determine whether certain age ranges tend to adopt a certain privacy stance towards specific categories of apps.

The role of developers. CAP-A is not only addressed to consumers, but also developers, who can *claim the development of a certain app*, giving them special privileges.

Mini-tours. An important feature is the concept of the *mini-tours*, which allow newcomers to get a grasp of the main CAP-A functionalities, through step-by-step tutorials.

3. Related Work

Various works aim to improve privacy awareness, but using different methods than CAP-A. In [3,4,5] various techniques (textual summarization, NLP, semantic text matching etc) are used to support users in understanding the content of PrP documents. Similarly, in [6], a remodelling of PrP documents is proposed, as well as an Annotator for visualizing them using semantic metadata. Tools for improving privacy awareness using visual techniques have appeared in [7,8], whereas [9] presents an app which enables users to behaviourally analyse the privacy aspects of other installed apps.

4. Conclusion and Future Work

We presented CAP-A, a socio-technical solution aiming to improve privacy awareness and users' understanding of the privacy implications associated with the use of any given mobile app. Our solution is based on crowdsourcing and collective intelligence measures. Despite the existence of a 1000-strong user base (partly through the sister initiative CAPrice[6]), only internal evaluation has been carried out for CAP-A so far; a large-scale evaluation through several pilots is currently planned. We also consider the incorporation of a debating/chatting tool (e.g., along the lines of our previous work, APOPSIS [15]), that will allow users, experts, and developers to express opinions on privacy-related aspects, share individual experiences, or justify viewpoints (e.g., on annotations).

Acknowledgement

This work has been supported by the EU H2020 programme under the NGI_TRUST grant agreement #825618.

References

[1] Anton, A.I., Earp, J.B., Bolchini, D., He, Q., Jensen, C., Stufflebeam, W.: The lack of clarity in financial privacy policies and the need for standardization. In: IEEE Security and Privacy, vol. 2, (2004).

[2] Sloan, R.H., Warner, R.: Unauthorized Access: The Crisis in Online Privacy and Security. CRC Press, Inc., 1st edn., (2013).

[3] Tesfay, W.B., Hofmann, P., Nakamura, T., Kiyomoto, S. and Serna, J. PrivacyGuide: towards an implementation of the EU GDPR on internet privacy policy evaluation. In the 4^{th} ACM International Workshop on Security and Privacy Analytics (2018).

[4] Wilson, S., Schaub, F., Ramanath, R., Sadeh, N., Liu, F., Smith, N.A. and Liu, F. Crowdsourcing Annotations for Websites' Privacy Policies: Can It Really Work?. In WWW-16, (2016).

[5] Oltramari, A., Piraviperumal, D., Schaub, F., Wilson, S., Cherivirala, S., Norton, T.B., Russell, N.C., Story, P., Reidenberg, J. and Sadeh, N. PrivOnto: A semantic framework for the analysis of privacy policies. Semantic Web, 9(2), (2018).

[6] Pandit, H.J., O'Sullivan, D. and Lewis, D. Personalised Privacy Policies. In European Conference on Advances in Databases and Information Systems, (2018).

[7] Angulo, J., Fischer-Hübner, S., Pulls, T., Wästlund, E. Usable transparency with the data track: a tool for visualizing data disclosures. In 33^{rd} Conference on Extended Abstracts on Human Factors in Computing Systems (2015).

[8] Raschke, P., Kupper, A., Drozd, O. and Kirrane, S. Designing a GDPR-compliant and usable privacy dashboard. In IFIP International Summer School on Privacy and Identity Management, (2018).

[9] Hatamian, M., Kitkowska, A., Korunovska, J. and Kirrane, S. 'It's Shocking!': Analysing the Impact and Reactions to the A3: Android Apps Behaviour Analyser. In IFIP Conference on Data and Applications Security and Privacy, (2018).

[10] McGonigal, J.: Reality is broken: Why games make us better and how they can change the world. Penguin (2011).

[11] Morschheuser, B., Hamari, J. and Koivisto, J., 2016, January. Gamification in crowdsourcing: a review. In HICSS-16, (2016).

[12] Kavaliova, M., Virjee, F., Maehle, N., Kleppe, I.A.: Crowdsourcing innovation and product development: Gamification as a motivational driver. Cogent Business & Management 3(1), (2016).

[13] Chrysakis, I., Flouris, G., Patkos, T., Dimou, A. and Verborgh, R.: REWARD: Ontology for reward schemes. In 17^{th} Extended Semantic Web Conference: Posters and Demos (2020).

[14] Chrysakis, I., Flouris, G., Ioannidis, G., Makridaki, M., Patkos, T., Roussakis, Y., Samaritakis, G., Stan, A., Tsampanaki., N., Tzortzakakis, E., Ymeralli., E.: CAP-A: a Suite of Tools for Data Privacy Evaluation of Mobile Applications. In 32^{rd}JURIX 2020, Demo session, (to appear).

[15] Ymeralli, E., Flouris, G., Patkos, T. and Plexousakis, D.: APOPSIS: A Web-based Platform for the Analysis of Structured Dialogues. In "On the Move to Meaningful Internet Systems" (2017).

[6] https://www.caprice-community.net

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200869

Automatic Removal of Identifying Information in Official EU Languages for Public Administrations: The MAPA Project

Lucie GIANOLA [a,1], Ēriks AJAUSKS [b], Victoria ARRANZ [c],
Chomicha BENDAHMAN [c], Laurent BIÉ [d], Claudia BORG [e], Aleix CERDÀ [d],
Khalid CHOUKRI [c], Montse CUADROS [f], Ona DE GIBERT [g], Hans DEGROOTE [d],
Elena EDELMAN [c], Thierry ETCHEGOYHEN [f], Ángela FRANCO TORRES [d],
Mercedes GARCÍA HERNANDEZ [d], Aitor GARCÍA PABLOS [f], Albert GATT [e],
Cyril GROUIN [a], Manuel HERRANZ [d], Alejandro Adolfo KOHAN [d],
Thomas LAVERGNE [a], Maite MELERO [g], Patrick PAROUBEK [a],
Mickaël RIGAULT [c], Mike ROSNER [e], Roberts ROZIS [b], Lonneke VAN DER PLAS [e],
Rinalds VĪKSNA [b] and Pierre ZWEIGENBAUM [a]

[a] *Université Paris Saclay, CNRS, LIMSI, Orsay, France*
[b] *Tilde, Riga, Latvia*
[c] *ELDA/ELRA, Paris, France*
[d] *Pangeanic – PangeaMT, Valencia, Spain*
[e] *University of Malta, Msida, Malta*
[f] *Vicomtech, Donostia, Gipuzcoa, Spain*
[g] *Barcelona Supercomputing Center, Barcelona, Spain*

AbstractThe European MAPA (Multilingual Anonymisation for Public Administrations) project aims at developing an open-source solution for automatic de-identification of medical and legal documents. We introduce here the context, partners and aims of the project, and report on preliminary results.

Keywords. automatic de-identification, legal documents, open-source, multilingual

1. Introduction

Interpreting European guidelines for data sharing implies to resolve the conflicting objectives stated, on the one hand in the PSI (Public sector information) directive which encourages administrations to share as much data as possible for re-use in an open-data perspective, and on the other hand, in the General Data Protection Regulation (GDPR), which requires the protection of personal data. As the GDPR becomes an obstacle to data sharing, removing personal information allows to share data. Nevertheless, removing identifying information from documents is a challenge that public administrations (PA) face in order to fulfill their open-data commitment, in every European language.

[1] Corresponding Author: Lucie Gianola, Université Paris Saclay, CNRS, LIMSI; lucie.gianola@limsi.fr

2. Context & objectives

Multilingual Anonymisation for Public Administrations (MAPA) is a project[2] funded by the Connecting Europe Facility (CEF)[3]. Academic and industrial partners from four countries are working together to develop, test and evaluate an open-source de-identification toolkit, fully customizable by end users for legal and medical domains, in all official European Union (EU) languages. The project aims to improve data sharing opportunities for PA.

De-identification consists in hiding directly-identifying information items: name, date of birth, address, contact information, etc. [1]. Anonymization consists in making it impossible to find out who a document is about. Text anonymization is extremely difficult to achieve automatically because the facts discussed in a document may be sufficiently eloquent to reveal indirectly the identity of the involved persons, for example in high-profile criminal cases where the general public is familiar with the broad outlines of the case. With medical documents, it is sometimes the mention of rare diseases combined with other criteria that makes identification possible. Both operations conflict with the need to maintain the legal relevance [2] of the document: for example, concealing all references to legal texts invoked in a judgment may compromise its logical structure, and concealing the facts discussed may render the text incomprehensible.

The project targets de-identification because even though it is not considered as effective as anonymisation, as a minimal approach it is sufficient in many cases, technically more achievable, and it can be evaluated more formally [3]. The most straightforward way to de-identify a document is to remove all identifying data such as names, addresses, phone numbers, etc., while retaining as much as possible from the original material: otherwise what will be left will be useless, in particular for Artificial Intelligence or Natural Language Processing (NLP). It is important to replace the removed language elements by something that hints at their type (e.g., someone's name with an identifier like Person), in a consistent fashion throughout the document (e.g. if several people are mentioned, all the text spans referring to their respective names ought to be replaced by the same identifier, all occurrences of Mr Doe, John Doe, John, etc., ought to be replaced by Person_1), to preserve the internal logic of the original document. NLP has seen the emergence of the concept of Named Entity for Information Extraction applications, which has been popularized among others by the DARPA/NIST MUC series of evaluation campaigns on natural language understanding, where they appeared in the 6th venue in 1995 [4]. The entity extraction process relies on a NLP method called *Named Entity Recognition* (NER), which spots in a text all mentions of information elements of pre-defined types, such as person names, dates, etc. While the de-identification of medical documents has already been the subject of much research [5], the de-identification of legal documents has received less attention so far [2,6].

Developing a de-identification system requires to define the different types of language entities potentially subjected to removal, as well as to annotate a sufficient amount of documents by hand, in order to obtain the training material required by the neural Machine Learning approaches that have become state of the art in NLP. The annotation process requires writing precise annotation guidelines that explain to human annotators how to identify and classify the relevant text elements.

[2] https://mapa-project.eu/

[3] https://ec.europa.eu/inea/en/connecting-europe-facility

3. Preliminary tests

The project foresees the use of word-embeddings trained through manually annotated examples. As proof of concept, we trained a model by fine-tuning BERT multilingual embeddings [7]. This has been done on a dataset combining data from the CoNLL Named Entity shared tasks: 2002 (Spanish) [8] and 2003 (English) [9]. As a result, this model allows to perform named entity recognition on four basic entity types (Location, Organization, Person, and Miscellaneous). The preliminary evaluation results were 93.4% of weighted F-score for entity recognition and classification among entity types PER, LOC, ORG, and MISC, and 97.98% of weighed F-score for binary entity classification, i.e., deciding whether or not words must be de-identified. We ran an experiment on the same sentence available in 23 official languages (translation in Gaelic language is missing):

> *whereas the founder of Charter 97, Aleh Byabenin, was found hanged at his home near Minsk in September 2010; whereas Belarus-born Pavel Sheremet, a spokesperson for the organisation behind Charter 97, was killed in a car bombing in Kiev, the capital of Ukraine, in July 2016;*

This extract from a European joint motion for a resolution[4] contains 11 named entities of several types (Date, Location, Person, Organization, as well as nationality and job). Note that for person and location names, the translations take into account the writing form used in each language (French: "Pavel Cheremet", German: "Pawel Scheremet", including declination forms: "Pavelas Šeremetas" for Latvian). Although the system was only trained on English and Spanish data, it succeeded in identifying named entities in all languages.

4. Annotation scheme

The annotation guidelines define six implicit top-level entity types: Person, Time, Location, Organisation, Amount, Vehicle. These entities encapsulate other explicit or implicit Level 2 entity types: a Person entity will contain Name, Age, Profession, etc. Level 2 entities are themselves made up of components and types, which are always explicit: a Name entity may contain a Title, Given name, Family name, etc. In order to make the annotation work easier, implicit entities are inferred from either their Level-2 entities or their Level-3 components/types. As the project deals with documents from two fields of expertise (legal and medical), annotation modalities need to be adapted to be relevant for both domains. This is in line with Nadeau et Sekine [10] who point out that the performance of NER improves when the domain and textual genre are considered. Given that we deal with texts from institutions, the use of a “role” tag is intended to annotate the “side” on which the mentioned person stands: from the institution (carers, members of court) or from the public (patient, plaintiff, defendant). Vehicle has been added as a Level 1 and 2 entity type (and its components: License plate number, Colour, Model, etc.) since it may be identifying in legal texts. Manual annotation tests were carried out on a corpus that includes, for each language, 12 documents of European case law[5], totalling approximately 2,000 sentences, via the INCePTION platform [11]. A baseline

[4] https://www.europarl.europa.eu/doceo/document/RC-8-2018-0451_FR.html

[5] https://eur-lex.europa.eu/

has been established from the first annotated datasets (Spanish, French, Croatian, Latvian, Romanian), and two models, monolingual and multilingual, have been trained. The multilingual model is slightly outperformed by the monolingual model (0,82 F1 to 0,85 F1 for French data). Discrepancies were discovered in the annotations across languages, which have allowed to adjust both annotations and guidelines.

5. Conclusion & perspectives

As the need for de-identifying texts will continue to grow in the European context, so will the need for automatic and multilingual de-identification solutions. To this end, we plan to evaluate its results not only in terms of performance (precision, recall, F-score), but also in terms of adaptability across text genres. Indeed, medicine and law produce very diverse types of texts (e.g., in the legal domain, case-law is very different from police interviews; see Zweigenbaum et al [12] for the medical domain). This remark will be essential if the application of the tool is to be extended to other administrations (social, agricultural, financial, etc.). We also rely on the collaboration of PA to test the solution during its development to ensure the best possible fit with the needs of end users.

References

[1] Chevrier R, Foufi V, Gaudet-Blavignac C, Robert A, Lovis C. Use and Understanding of Anonymization and De-Identification in the Biomedical Literature: Scoping Review. J Med Internet Res. 2019 May 31;21(5):e13484.

[2] Plamondon L, Lapalme G, Pelletier F. Anonymisation de décisions de justice. In: 11e Conférence sur le Traitement Automatique des Langues Naturelles. Fès, Morocco: Bernard Bel et Isabelle Martin (eds); 2004. p. 367–376.

[3] Meystre SM, Friedlin FJ, South BR, Shen S, Samore MH. Automatic de-identification of textual documents in the electronic health record: a review of recent research. BMC Medical Research Methodology. 2010 Aug;10(1):70. Available from: https://doi.org/10.1186/1471-2288-10-70.

[4] Grishman R, Sundheim B. Design of the MUC-6 evaluation. In: Proceedings of the 6th conference on Message understanding. Stroudsburg, PA: Association for Computational Linguistics; 1995. p. 1–11.

[5] Uzuner O, Luo Y, Szolovits P. Evaluating the State-of-the-Art in Automatic De-identification. Journal of the American Medical Informatics Association. 2007;14:550–563.

[6] Tamper M, Oksanen A, Tuominen J, Hyvönen E, Hietanen A. Anonymization Service for Finnish Case Law: Opening Data without Sacrificing Data Protection and Privacy of Citizens. In: International Conference on Law via the Internet, LVI; 2018. .

[7] Devlin J, Chang M, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR. 2018;abs/1810.04805.

[8] Tjong Kim Sang EF. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: COLING-02: The 6th Conference on Natural Language Learning; 2002. .

[9] Tjong Kim Sang EF, De Meulder F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003; 2003. p. 142–147.

[10] Nadeau D, Sekine S. A survey of named entity recognition and classification. Linguisticae Investigationes. Linguisticae Investigationes. 2007;30(1):3–26.

[11] Klie JC, Bugert M, Boullosa B, Eckart de Castilho R, Gurevych I. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. Santa Fe, NM: ACL; 2018. p. 5–9.

[12] Zweigenbaum P, Jacquemart P, Grabar N, Habert B. Building a Text Corpus for Representing the Variety of Medical Language. In: Patel VL, Rogers R, Haux R, editors. Proceedings of the 10th World Congress on Medical Informatics. London, UK; 2001. p. 290–294.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200870

Identifying the Factors of Suspicion

Morgan A. Gray,[a1] Wesley M. Oliver,[b] and Arthur Crivella[b]
[a] *Crivella Technologies, Ltd*
[b] *Duquesne University School of Law*

Abstract. Probable cause determinations are problematic. Like all court decisions using totality-of-the-circumstances tests, it is difficult to use one decision – or even a few – to foresee a subsequent outcome. No human is capable of reading all the relevant Fourth Amendment opinions relevant to resolving any search and seizure issue. Machines may be capable of this task and to do so they will need to be able to identify particular types of suspicious factors from the various ways courts describe the factors. This project examines the ability of three machine learning models to examine the relevant text of opinions to identify the suspicious factors courts used to determine whether adequate suspicion existed from an intrusion protected by the Fourth Amendment.

Keywords. "information retrieval," "reasonable suspicion," "totality of the circumstances," "*k*-nearest neighbor," "decision tree," "logistic regression"

1. Introduction

Totality-of-the-circumstances legal tests, such as probable cause and reasonable suspicion, do not provide much meaningful guidance for the judges and police officers who have to apply them on a daily basis. Thousands of decisions in the United States, rendered by judges at every place in the judicial hierarchy, have interpreted these legal standards [1]. Machines are certainly capable of digesting far more information than humans, and thus, theoretically have greater capacity to apply judicial interpretations of these standards to subsequent cases. No human could read all of the Fourth Amendment cases courts have decided. And certainly, no human could determine the extent to which courts collectively conclude that a suspicious factor, or a combination of suspicious factors, demonstrate the existence of a current or past crime. The question, then, is whether machines can perform these tasks.

This paper considers the ability of computers to perform the essential first step in evaluating the capacity of computers to evaluate suspicion: examine the text of judicial opinions and identify the type of suspicious circumstances described by the facts. Applying the work of Jaromir Savelka, Huihui Xu, and Kevin Ashley [2] to a new type of dataset, three machine learning models were used to identify the language used to describe various bases of suspicion from judicial decisions which analyzed whether an officer had reasonable suspicion to detain a motorist in order to deploy a drug-sniffing dog. Drug interdiction decisions were evaluated because they typically involve very similar facts – a car is stopped for an ordinary traffic offense and the officer looks for a

[1] Morgan. A. Gray, Corresponding author, Crivella Technologies, Ltd., 3945 Forbes Ave, Pittsburgh, Pennsylvania, United States, 15260, United States of America; E-mail: morgangray99@icloud.com

basis to hold the car until a drug-sniffing dog can arrive – and provide a limited universe of potential bases of suspicion.

The models performed with varying degrees of accuracy, but with 56% accuracy, the logistic regression classifier was able to correctly identify the language of a court as fitting within one of 14 categories. This degree of accuracy suggests that machines are capable of at least the first step required to evaluate reasonable suspicion from a corpus of Fourth Amendment decisions.

2. The Data Set

The data set is comprised of 156 opinions which reflect relevant cases from almost every jurisdiction across the United States. We had success identifying and using a wide array of cases from a profoundly diverse group of jurisdictions. Search and seizure law is defined by the Fourth Amendment to the U.S. Constitution and the standard for permitting the detention of a car for a drug dog is therefore the same in all federal and state courts. If an officer, in any jurisdiction in the United States, has reasonable suspicion to believe drugs are present in a car stopped for a traffic violation, the officer may detain the car for a reasonable amount of time until a drug dog arrives [3].

The opinions were chosen because they all addressed a single legal question – whether an officer had reasonable suspicion to believe a stopped motorist possessed drugs. The cases fitting this criteria were read by one of the attorneys on our team. Each sentence describing a factor of suspicion was annotated to reflect which of twelve categories of suspicion were being described. Additionally, sentences reflecting the court's conclusions about the sufficiency of suspicion were annotated as fitting into one of two categories – a judicial finding of sufficient or insufficient suspicion. Within the 156 cases, 658 separate sentences that described the officer's suspicion were labeled as fitting into one or more of the following categories:

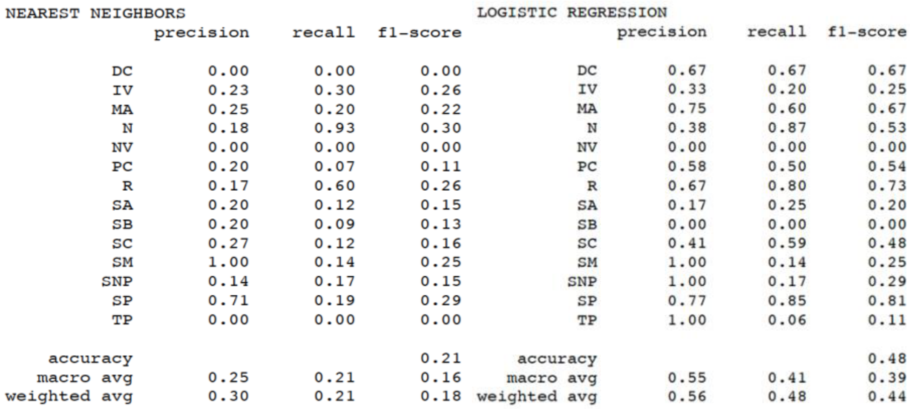
The fourteen categories of suspicion were described as such:

(1) Drug City (DC): travel to or from a city known as a source or endpoint for drugs.
(2) Items in Vehicle (IV): content such as multiple cell phones or signs of long travel.
(3) Masking Agents (MA): including deodorizers, air fresheners, cigars, etc.
(4) Nervousness (N): including nervous behavior such as shaking or trembling.
(5) No Value (NV): any descriptions of officers that legally do not support suspicion.
(6) Prior Convictions (PC): priors involving drug offenses.
(7) Rental (R): motorist driving a rental car.
(8) Suspicious Answers (SA): suspicion resulting from field interrogation.
(9) Suspicious Behavior (SB): actions of defendant based on observation.
(10) Suspicious Circumstances (SC): unusual items, covering a range of possibilities from clothing to music choice.
(11) Suspicious Movements (SM): efforts believed to conceal drugs or a weapon.
(12) Travel Plans (TP): usual routes or inconsistent travel stories.
(13) Suspicion Not Present (SNP): court concluded reasonable suspicion absent.
(14) Suspicion Present (SP): court concluded reasonable suspicion present.

3. Results & Discussion

We used the method developed by Savelka, Ashley, and Xu to evaluate the ability of three machine learning models – decision tree, *k*-nearest neighbor, and logistic regression classifiers – to assess the ability of potential sorting of the language of Fourth Amendment opinions into categories identified by lawyers[2]. Salvelka et al. looked at the ability of these three models to identify sentences in judicial opinions that had been identified by human lawyers to be relevant to interpreting terms in statutes. Our experiment applied this methodology to a new application – the identification of the types of suspicion described by a court in a Fourth Amendment opinion. We applied each classifier to the annotated text of judicial opinions, testing their respective abilities to identify which of fourteen categories, identified in the previous section, was being described by sentences in the opinion.

Computers generally have difficulty classifying sentence-length texts[2][4], but our work with these three classifiers suggests that at least some categories of suspicion can be extracted from judicial language. The logistic regression model performed the best of the three across all categories, with an accuracy rate of 56%, demonstrating a particular aptitude for identifying those sentences describing drug cities, nervousness, prior convictions, masking agents, and rental agreements. The regression classifier was also quite adept at identifying the court's conclusion about the adequacy of suspicion. Figure 1 demonstrates the results for each of the categories for *k*-nearest neighbor and logistic regression classifiers.



```
NEAREST NEIGHBORS                                   LOGISTIC REGRESSION
                 precision    recall  f1-score                   precision    recall  f1-score

          DC          0.00      0.00      0.00            DC          0.67      0.67      0.67
          IV          0.23      0.30      0.26            IV          0.33      0.20      0.25
          MA          0.25      0.20      0.22            MA          0.75      0.60      0.67
           N          0.18      0.93      0.30             N          0.38      0.87      0.53
          NV          0.00      0.00      0.00            NV          0.00      0.00      0.00
          PC          0.20      0.07      0.11            PC          0.58      0.50      0.54
           R          0.17      0.60      0.26             R          0.67      0.80      0.73
          SA          0.20      0.12      0.15            SA          0.17      0.25      0.20
          SB          0.20      0.09      0.13            SB          0.00      0.00      0.00
          SC          0.27      0.12      0.16            SC          0.41      0.59      0.48
          SM          1.00      0.14      0.25            SM          1.00      0.14      0.25
         SNP          0.14      0.17      0.15           SNP          1.00      0.17      0.29
          SP          0.71      0.19      0.29            SP          0.77      0.85      0.81
          TP          0.00      0.00      0.00            TP          1.00      0.06      0.11

    accuracy                              0.21      accuracy                              0.48
   macro avg          0.25      0.21      0.16     macro avg          0.55      0.41      0.39
weighted avg          0.30      0.21      0.18  weighted avg          0.56      0.48      0.44
```

Figure 1. We did not include the results from derived from the decision tree classifier because most factors were poorly developed. Only five factors scored above 0.00, but scored fairly well. N scored (P – 0.69, R – 0.73, F-1 – 0.71), R scored (P – 0.50, R – 0.40, F-1 – 0.44), SC scored (P – 0.27, R – 0.88, F-1 – 0.41), SNP scored (P – 1.00, R – 0.50, F-1 – 0.67), SP scored (P – 0.76, R – 0.70, F-1 – 0.73).

The broad range of performance across each category is likely explained by the variety of language that could be used to describe suspicious factors within each category. Predictably, categories that potentially encompass a variety of circumstances, that can be described in a number of ways, were harder for the model to identify than categories that are most often described using very similar terms.

Courts use a fairly common vocabulary to describe masking agents (MA), nervousness (N), and prior convictions (PC), likely explaining the comparative advantage in identifying sentences describing these categories. Categories identifying a broader range of behavior, as one would expect, were comparatively difficult for the model to identify. Suspicious answers (SA) to questions could take a number of forms. Suspicious behaviors (SB) could include anything from talkativeness to sullenness to combativeness. One would therefore also expect the model to have difficulty identifying sentences describing suspicious circumstances (SC), a category that could include anything from wearing a tie-dye shirt to playing loud religious music. The classifier's comparative success rate (*precision (P):* 0.41; *recall (R):* 0.59) is somewhat remarkable, though likely explained by presence of this category of suspicion was quite common, accounting for 22% of the sentences analyzed.

The regression classifier had remarkable difficulty in identifying factors described by officers as suspicious but which courts disregard as having no legal value. (*P*: 0.00; *R:* 0.00). These cases were coded as having no legal value (NV). As in other catch-all categories, sentences coded in this category could describe an expansive range of circumstances. The performance of the model here was non-existent. This is almost certainly explained by the fact that only two sentences in the dataset were coded NV.

For most categories, the regression model produced similar results for recall and precision, with three notable exceptions – travel plans (*P:* 1.00; *R:* 0.06), suspicious movements (*P*: 1.00; *R*: 0.14), and judicial findings that there was inadequate suspicion (*P*: 1.00, *R*: 0.17). Descriptions of unusual or inconsistent travel plans could be described in any number of ways that are obvious to human readers but may difficult for models to detect, for instance the driver and passenger identifying travel two cities in different parts of the country. Suspicious movements accounted for a small fraction of the sentences annotated. The model's inability to recall judges' conclusions of inadequate suspicion is a mystery. There was no dearth of data for this category, and around 35% of cases analyzed concluded that suspicion was lacking. Yet, the model proved particularly good at identifying conclusions that suspicion was sufficient for a detention (*P:* 0.77; *R:* 0.85).

These results, using models that have not been tailored to this particular data set, suggest that machine learning models can be trained to meaningfully identify particular factors of suspicion in judicial opinions. The logistic regression classifier performed reasonably well across all categories, and exceptionally well in identifying language describing some categories. Moving forward we look to expand our data set and experiment on other models and to adapt them to better serve our categories and data. With larger datasets, a larger number of more precisely defined categories of suspicion can be employed. Improvement is certainly important and necessary, but our results appear to demonstrate the ability of a machine learning to meaningfully identify the types of suspicion a court relied upon on rendering a Fourth Amendment decision.

Acknowledgements

The authors acknowledge the contribution of Wayne Ackman for his efforts in helping to compile the dataset. The authors are also indebted to Dr. Kevin Ashley for his mentorship generally and with this project particularly.

References

[1] Brent E. Newton. The Real World Fourth Amendment. Hastings Const. Law Q. 2016 Apr; 43(4): 759-810.

[2] Jaromir Savelka, Huihui Xu, and Kevin D. Ashley. Improving Sentence Retrieval from Case Law for Statutory Interpretation. In: Floris Bex, editor. Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19), 2019 June 17-21, Montreal, QC, Canada. New York (NY): ACM; c2019. p. 113-112.

[3] Wayne R. LaFave. The "Routine Traffic Stop" from Start to Finish: Too Much "Routine," Not Enough Fourth Amendment, Michigan Law Review. 2004 Aug; 102(8): 1846-1902.

[4] Vanessa G Murdock. Aspects of sentence retrieval. ACM SIGIR Forum. 2001 Dec; 41(2): 127.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200871

Sleeping Beauties in Case Law

Pedro V. HERNANDEZ SERRANO [a1], Kody MOODLEY[a,b], Gijs VAN DIJCK [b]
Michel DUMONTIER[a]
[a] *Institute of Data Science at Maastricht University*
[b] *Faculty of Law, Maastricht University*
{p.hernandezserrano;kody.moodley;gijs.vandijck;michel.dumontier}@maastrichtuniversity.nl

Abstract. A challenge in computational legal research is the quantitative assessment of "relevance" in a network of court decisions. The term "sleeping beauty" (SB) was coined to denote an article that received almost no attention immediately after publication, but suddenly received multiple citations many years later. These publications can be identified by calculating their *Beauty coefficient (B-coefficient)*. In this contribution, we apply approaches used for identifying SBs to decisions arising from the Court of Justice of the European Union (CJEU). We compared B-coefficients of CJEU cases with their centrality scores from classical algorithms from network analysis, finding that these measures tend to correlate. We discuss the implications of this that are interesting for legal scholars, acknowledging that future work is required to calibrate the scale of the time variable in the B-coefficient formula for finer-grained application to case law. Our study's setup provides a foundation for new case law analytics methodologies that extends the power of traditional network analysis techniques for answering questions about the behavior of European courts.

Keywords. Citation Networks, Network Analysis, Sleeping Beauties in Science, Empirical Legal Research, Computational Legal Research

1. Introduction

A 'Sleeping Beauty' (SB) is a publication that goes unnoticed ('sleeps') for a long time, before it suddenly attracts a lot of attention and is cited frequently ('is awakened'). Van Raan [1], who first introduced SB terminology in a computational setting, provided a mathematics for describing the phenomenon. This mathematics offered a way to compare SB significances by balancing their number of citations once awakened and their age. However, van Raan's method was not generalizable outside of the particular corpus of scientific articles he studied. Other approaches, such as that of Redner [2], were focused on particular scientific domains e.g. Physics. In 2015, Ke et al. built upon van Raan and Redner's works by proposing a formula for identifying them in general scientific articles [3]. The central mathematical measure used in this study is the *beauty coefficient* or *B-coefficient* for any publication's "*beauty*": a term denoting the publication's significance as indicated by its age and degree of influence after awakening. Importantly, their mathematical foundations are a priori applicable outside the context of scientific articles, to any corpus of authored documents in which the documents reference each other over time. Since court decisions by the Court of Justice of the European Union (CJEU) have that characteristic, B-coefficients can be calculated for them and the associated citation behaviour can be studied. Previous network analyses of case law have largely overlooked the speed or delay in citations. In classical and doctrinal legal research literature there are also many references to the term "sleeping

[1] Corresponding Author: Institute of Data Science at Maastricht University, Paul-Henri Spaaklaan 1 6229 EN, Maastricht, The Netherlands; E-mail: p.hernandezserrano@maastrichtuniversity.nl

beauty" when describing legal articles, directives and court decisions with delayed recognition and significance [4, 5 and 6]. However, the term is used qualitatively to describe the significance of these texts with no acknowledgement of the potential to *computationally* define, identify and compare them. There has not been, to date, an application of SB metrics to study court decisions and this work seeks to fill this gap.

In this contribution, we apply the B-coefficient measure to study citation behavior of court decisions by the Court of Justice of the European Union (CJEU). Our objectives are to understand the influencing factors of SB cases, learn the prevalence of SBs in case law, and calculate the correlation between the B-coefficient and legal network analysis measures such as Degree centrality, Closeness centrality, PageRank and Hyperlink-Induced Topic Search (HITS). By doing so, we explore whether applying a SB methodology provides additional value to other (centrality) measures in determining the relevance of court decisions, offering insight into whether relevant cases are overlooked. From a more legal perspective, we are interested in whether SBs can be detected in time for them to be used when building and presenting a case. Our research questions are: RQ1: What is the prevalence and proportion of SBs in different areas of EU law? RQ2: Does the B-coefficient correlate with classic centrality metrics? RQ3: Do the most relevant cases have a slow awakening? RQ4: Does the duration of a case have an effect on sleep time?

Awakenings can be relevant because they may indicate importance of a certain court decision, potentially signaling a different direction by the court or 'new' arguments in old decisions, showing that certain legal questions have become salient again. To ensure similar cases are treated equally [7], it is important to recognize such developments and further analyze SBs, computationally and non-computationally.

2. Methodology

Ke et al.'s mathematical formula for the Beauty Coefficient *B* is used to quantify and capture both the incoming citations and duration of the "sleeping" period. It was implemented on Web of Science articles (384,649 papers) and American Physical Society (APS) articles (22,379,244 papers) and has power-law behavior. Our Python implementation of the B-coefficient formula is available in a public repository[2]. In previous work we gathered the CJEU data [8], consisting of metadata and citations in CSV format for 13,358 judgements (all published until December 2019)[3]. For this study we selected the subset of CJEU cases cited at least once in the citation network, resulting in 8,979 out of the 13,358 cases (67.21%), and then computed the B-coefficient and variables for all selected cases, resulting in a 13358 by 13 matrix of values, which served as the core dataset for our analyses.

Assuming a case with many incoming citations (*in-degree*) is a proxy for relevance [9], we split the dataset into two groups: *Top cited* cases (cases in the upper 10th percentile) and *Non-top cited* cases (the remainder of cases). The average citation count per case and the standard deviation (SD) for Top cited cases are 18.27 and 9.89 respectively, while for Non-top cited cases they are 3.31 and 2.41 respectively. The average sleeping time in years of all cases is 3.55 (SD 6.22). The maximum value of *sleeping time* is 55 years and the corresponding case has the highest B-coefficient of 4709.50 and an awakening time of 2 years – meaning that it slept for 55 years, was awakened, and then took 2 years to receive its maximum number of annual citations. On

[2] https://github.com/maastrichtlawtech/sleeping-beauties-case-law

[3] https://github.com/MaastrichtU-IDS/case-law-analysis

average, the expected time to gain the maximum number of annual citations in a year for all cases is 0.94 (SD 0.58) years.

A case's B-coefficient has to be above a certain threshold for the case to be a SB. This threshold is not fixed because it depends on factors specific to the citation behavior in the corpus. We adopted Ke et al.'s proposed baseline [3] of taking the cases with the top 0.1% of the B-coefficient values as the SBs. We therefore identified 90 unique SBs from the CJEU rulings dealing with a variety of legal topics.

3. Results and Findings

RQ1: What is the prevalence and proportion of SBs in the different areas of EU law? We aggregated the SBs according to legal topics as specified on the EUR-Lex website. *Approximation of Laws* has the most SBs (8 cases). *Concerned Practices* cases have, on average, the highest B-coefficients (93.93). *Copyright and related rights* have the highest proportion of SBs (1.47%). We also counted the number of cases which cite a particular piece of legislation. Then, for each legislative topic, we identified which of the cases are SBs (according to our threshold). We found that the *Regulation (EU) 2017/1001 of the European Parliament of 14 June 2017 on the European Union trademark* is the legal topic addressed by the most (six) SB cases.

RQ2: Does the B-coefficient correlate with classic centrality metrics? We selected the 90 SBs discovered and calculated classical centrality measures (Degree centrality, Closeness centrality, PageRank, and HITS algorithm[4]) with the whole citation network. Degree centrality and PageRank showed significance ($p < .05$), whereas, Closeness centrality and HITS algorithm showed only marginal significance ($p < .10$). All four metrics showed positive correlations, but no more than .30. A difference of means test compared the average sleeping time between the two case groups. Sleeping time has a right skewed distribution (Sk = 3.04); thus, a log transformation was calculated, before a T-test was performed on the *log sleeping* with a one-sided distribution and equal variance, resulting in a null hypothesis rejection: Top-cited cases sleep longer than the Non-top cited ones (0.86 years – almost 11 months).

RQ3: Do the most relevant cases have a slow awakening? The average awakening time was compared between the Non-top cited and the Top-cited groups using a T-test performed on a one-sided distribution with unequal variance. The null hypothesis was rejected, meaning that the mean awakening time of the Top-cited cases is larger than that of the Non-top cited cases (0.7 years – around 8 months).

RQ4: Does the duration of a case have an effect on sleep time? Case duration is the quantification of the days between the lodge and publication dates, potentially indicating case complexity. *Case duration* has a normal distribution with a mean and median of 1.5 years (SD 0.74). However, *sleeping time* is right skewed; therefore, a *Spearman* coefficient is used to calculate a correlation. The test's results were non-significant: there are no signs of correlation between a case's *sleep time* and the *case duration* in years.

4. Discussion and Conclusion

We applied the *B-coefficient* to identify publications with delayed citations (sleeping beauties) from CJEU case law. We found the methodology to be complementary to traditional network analysis metrics, adding novel analytical power to computational

[4] Degree centrality, Closeness centrality, PageRank and HITS algorithm were calculated using the open source implementations of NetworkX Python package. https://networkx.github.io/

studies of case law. By applying these metrics to CJEU cases, we observed the following main results: Top-cited cases usually sleep longer and tend to awaken slower than other cases. There are no signs that case duration has any effect on sleep time. While the B-coefficient generally correlates with well-known centrality measures when identifying significant cases, it does identify new ones missed by traditional metrics.

If a relevant case is dormant for a while, once it is cited the awareness of this case increases rapidly, and it is swiftly used in other decisions. On average, decisions reach the point of highest citations within the first year, but often the time between the lodge date and decision date is longer than a year, meaning parties may not know which decisions are relevant to their cases. Future research should analyze the extent to which different case types cite previous decisions that are similar. B-coefficient's positive correlation with four traditional centrality metrics, but no more than 0.30, suggests that the *B-coefficient* partly captures the information provided by traditional measures, yet produces complementary information. However, this requires further analysis as the *sleep–peak ratio* could be indicative of cases being (or becoming) landmark cases. Furthermore, Top-cited cases tend to sleep longer than the Non-top cited ones; thus, the relationship between SBs and B-coefficients with precedents and landmark cases should be explored further.

In future work we intend to address the unweighted influence of case age on the B-coefficient. The frequency and volume of citations in case law follows a different behavior from scientific literature, which is published and cited more frequently than case law [1, 2, 3]. Therefore, the hypothesis is that we need to calibrate the B-coefficient formula for case law by tuning the intensity of the timescales in the formula to match the case law context. We will also perform a deeper study of the legal topics that SBs tend to address, as well as enlist the aid of legal researchers to corroborate our findings and comment on their significance, especially comparing them across subfields of law.

References

[1] van Raan, A. F. J. (2004). Sleeping Beauties in science. *Scientometrics*, *59*(3), 467–472. https://doi.org/10.1023/B:SCIE.0000018543.82441.f1

[2] Redner, S (2005) Citation statistics from 110 years of physical review. Phys Today 58(6):49–54.

[3] Ke, Q., Ferrara, E., Radicchi, F., & Flammini, A. (2015). Defining and identifying Sleeping Beauties in science. *Proceedings of the National Academy of Sciences*, *112*(24), 7426–7431. https://doi.org/10.1073/pnas.1424329112

[4] Clancy, Pearce (2018). Arise, Sleeping Beauty: What PESCO Means for Ireland. Irish Yearbook of International Law, Vol. 13, pp. 79-98

[5] von der Dunk, Frans G. (2008). A Sleeping Beauty Awakens: The 1968 Rescue Agreement after Forty Years. Journal of Space Law, Vol. 34, Issue 2 (Winter 2008), pp. 411-434

[6] Stedman, John C. (1957-1958). Merger Statute: Sleeping Giant or Sleeping Beauty. Northwestern University Law Review , Vol. 52, Issue 5 , pp. 567-617.

[7] Maltz, E. The nature of precedent. New California Law Review 1987, vol. 66, 367.

[8] Moodley, K., Hernandez Serrano, P. V., van Dijck, G., & Dumontier, M. (n.d.). *IOS Press Ebooks—Similarity and Relevance of Court Decisions: A Computational Study on CJEU Cases*. Retrieved 7 August 2020, from http://ebooks.iospress.nl/volumearticle/53654

[9] M. van Opijnen (2012). Citation Analysis and Beyond: in Search of Indicators Measuring Case Law Importance. In JURIX, 250, 95-104, IOS Press

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200872

Digital Enforceable Contracts (DEC): Making Smart Contracts Smarter

Lu-Chi LIU [a,1], Giovanni SILENO [a] and Tom VAN ENGERS [b,a]
[a] *Informatics Institute, University of Amsterdam, Amsterdam, Netherlands*
[b] *Leibniz Institute, University of Amsterdam/TNO, Amsterdam, Netherlands*

Abstract. The combination of smart contracts with blockchain technology enables the authentication of the contract and limits the risks of non-compliance. In principle, smart contracts can be processed more efficiently compared to traditional paper-based contracts. However, current smart contracts have very limited capabilities with respect to normative representations, making them too distant from actual contracts. In order to reduce this gap, the paper presents an architectural analysis to see the role of computational artifacts in terms of various ex-ante and ex-post enforcement mechanisms. The proposed framework is assessed using scenarios concerning data-sharing operations bound by legal requirements from the General Data Protection Regulation (GDPR) and data-sharing agreements.

Keywords. Smart contracts, Norm representation, Normative reasoning, Automated enforcement, Data sharing infrastructures, GDPR

1. Introduction

Smart contracts were originally motivated by the wider purpose of facilitating the performance and enforcement of traditional paper-based contracts [1], but today there is no reference in smart contracts to normative constructs as those to be found in legally binding contracts. Several studies have shown that it is possible to perform reasoning tasks on blockchain via smart contracts, typically by querying an off-chain, trusted oracle [2] (e.g. a reasoner module [3]); this integration enables more (normative) expressiveness. Enforcement can also be achieved by means of dedicated social infrastructures. For instance, [4] proposes a model of incentives to enable enforcement for off-chain activities. On similar lines, in production settings, consider e.g. Kleros [5], a blockchain-based decentralized application was developed for multipurpose dispute resolution; smart contracts can assign Kleros as their arbitrator infrastructure in case a dispute occurs (see also Codelegit, Juris, Oath, etc.). Yet, these solutions are heterogeneous in methods and cover different aspects of compliance. The core of our contribution is to frame the problem in terms of possible types of enforcement mechanisms, to provide a modular architecture that covers the distinguishable application types. Rather than focusing on technology-dependent functions of smart contracts, this paper advances the concept of *digital enforceable contract* (DEC), with the purpose of highlighting the higher-level functions that any sound computational normative artefacts are expected to provide.

[1] Corresponding Author: l.liu@uva.nl. This research is funded by the Dutch Organisation for Scientific Research (NWO) under contracts 628.009.014 (SSPDDP project) and 628.001.001 (DL4LD project).

2. Implementing Enforcement Mechanisms

Types of enforcement Enforcement mechanisms can be distinguished in two main approaches, depending on *when* they play a role with respect to violations: *ex-ante* (before the facts) and *ex-post* (after the facts). In computational domains, most of the attention is put on the first approach; in socio-legal settings the term typically refers to the second approach, plausibly because legal activity is usually triggered by the occurrence of violations. We identified the following patterns. To avoid violations to occur (*ex-ante enforcement*) it is necessary to check: (a) whether a program/an action is permitted before executing; (b) whether any positive duties holds. As to identify that a violation has occurred (*ex-post enforcement*), we need to check: (c) whether a prohibited action has been performed; (d) whether a positive duty has not been fulfilled. The last two conditions requires active monitoring by an 'enforcement agent', i.e. an agent that has an *institutional power* to force that the duty is fulfilled, or that some other remedy is provided, and/or to 'punish' the agent that hasn't fulfilled the duty.[2] Additional design dimensions can also be taken into account. First, all enforcement mechanisms rely on some conditions that need to be evaluated (about occurrence of events, or properties of agents, etc.). This evaluation can be *lazy* (computed only at the moment of need), or *eager* (as soon as relevant conditions become true). Second, regulation can be *internal* (the agent on itself) or *external* (by a enforcer agent). Third, the monitoring task can be *internalized*, as when they are set up by claimants of duties, or *externalized*, e.g. by some infrastructural component, or by witnesses.

A modular architecture for enforcement To deal with the richness observed above, we propose a architectural model consisting of a number of modules associated to dedicated control and enforcement mechanisms that can be imported at need. This minimal set of modules has been selected as capturing and providing the functionality necessary to run all enforcement constructs.

We assume a distributed computation setting, representative of e.g. data-sharing infrastructures. We consider as minimal unity of agency a computational **actor**, characterized with a name/id. In a data-sharing infrastructure one might expect actors running applications for *users* of the infrastructure, as well as actors running applications for the *owners/maintainers* of the infrastructure, as e.g. for enforcement purposes. Actors can be then decomposed to a number of components, having unique functions. A **program** is a list of instructions which can be regarded as a plan to achieve a given design goal associated to that actor. Actors can have more than one programs (plans) to opt from depending on the situation. An **executor** provides the internal control of the actor. It follows the execution of the currently selected program or modifies the control flow if needed. A **message queue** is the communication channel for actors to interact with each other. It delivers and receives messages to/from other actors. A **monitor manager** creates and destroys monitors which hook to certain events or facts. For example a user can set up a monitor to observe whether its action has failed or not, while an enforcer can set up monitors for violations. A **regulator** is the module dedicated to normative reasoning. It is initiated with specifications of norms and should be fed up with factual data. It

[2] Considering power merely as a conditional obligation (e.g. if this action is performed, then this duty comes to hold), one cannot model the fact that powers themselves can be created and destroyed, depending on the dispositions set by the contract. The ex-ante/ex-post distinction needs thus to be inflected to the case of power.

provides regulatory information, for example whether a certain instruction or program is permitted to run, and/or whether the instruction will lead to any positive duties.

Technically the regulator can be realized as an external component to the actors as well, so that a group of actors can share the same normative reasoner. It can be regarded as a "legal" consultant who provides conclusions from the associated normative specifications when queried by the executors with some input information. The interactions of a shared regulator are not functionally dissimilar from that of any other actor, therefore this module follows the same communication channels used by other actors. With respect to content, a few query templates can be identified for the communication between executor and regulator: e.g. (a) What position do self/other actor have now (with respect to a certain action)? (b) If self/other actor performs a certain action, what kind of position self/other will take? (c) Given a source self/other actor's position, which actions can self/others perform in order to reach a certain target position?

3. Proof of Concept

To provide an example of application, we assessed the proposed architecture on a data-sharing scenario in a context relevant to the GDPR. According to the GDPR, the data-controller needs to have consent from the data-subject to process his/her data. Once given, the data-subject can at any moment modify or revoke his/her consent. We modeled this normative content into a logic-based representation[3] and set up a server interfacing with a suitable reasoner. This server acts as an externalized regulator, and is implemented as an actor itself, receiving normative requests from other actors and answering them.

We considered then three agents/actors: (1) a data-controller "Bank", (2) a data-subject "John", and (3) a data-processor "Adcom". For the three actors, we created possible actions which could be performed to interact with each other, for example "give consent", "share data" and "send advertisement". Finally, for each type of enforcement we set up a *simulation* to test whether the proposed solution functions as expected.[4] In the following paragraphs we show how our proposed architecture and the reasoning mechanism can be used for ex-ante enforcement as well as for ex-post enforcement.

Ex-ante enforcement for permission checking The Bank attempts to use John's data for analysis. By sending a query regarding the permission of such action to the regulator, the regulator will inform the executor of Bank that, for being allowed, it must obtain a consent from John. The Bank will then send this request to John, setting up an adequate monitor for the reply. At the reception of the message, the executor of John will select the program giving consent and execute it. A new message is created and sent to the requesting Bank. This message is captured by the monitor of Bank and eventually delivered to its executor, which is now aware of the consent and can start using John's data.

Ex-ante enforcement of positive duties Continuing the previous scenario, now John (the user) changes his mind. He wants to modify his consent by asking the Bank to replace the old purpose "data analysis" to "marketing". By means of its regulator, Bank is aware

[3] More concretely, the models we used were written in eFLINT [6], a language for specifying policies based on normative frames. Note however that any other choices would have been equally good in functional terms.

[4] The code used for the proof of concept can be found on `https://gitlab.com/evelynliu324/digital-enforceable-contract`.

that the modification consent request will lead to a duty and can thus set up a monitor for it. Receiving the request message from John triggers the monitor, creating a notification about taking action.

Ex-post enforcement of violations of duty Adcom acts as a service provider for Bank to place advertisements. Since John now consents to have his data used for marketing purpose, Bank is permitted to share his data to Adcom. However, after receiving too many advertisements, John revokes his consent and requsets to remove his data. According to the GDPR, Bank, as data-controller, has the duty to fulfill these requests. In the model we set up a made-up norm that if the data-controller did not respond to the request within two weeks, the duty would be regarded as being violated. On this basis, John sets up a monitor with a timeout mechanism to check for violation. When the duty is due and not fulfilled, the monitor will send a message to the executor, notifying a violation of duty.

Ex-post enforcement of violations of prohibitions Even if removal and revocation have been confirmed, John sets up monitors for Bank and Adcom for known illicit behaviors. Suppose that Bank has performed the duty to revoke John's consent but Adcom keeps processing John's data sending promotions to John. As a result, the monitor notifies the reception of advertisements from Adcom, concluding that this is a violation of prohibition. The executor of John consults the regulator for deciding how to act upon it.

4. Conclusions

The limited norm representation capability makes smart contracts unable to work with norms similarly to traditional paper-based contracts. To address this limit, this paper started investigating architectural possibilities for *digital enforceable contracts*, with reasoning capacities enabling ex-ante and ex-post enforcement through an integrated normative reasoner and possibly a dedicated social infrastructure. The usability and effectivity of the proposed framework have been assessed by a practical scenario case involving data sharing operations subjected to the GDPR, showing that the overall architecture is sound and can support automated enforcement. The example was sufficient for our purposes, but we do not claim that it covers all perspectives and complexity of real-world contracts. In typical data-sharing environments, operations are not only subjected to the GDPR, but also many other regulations. Second, a proper ex-post enforcement requires an adequate design, reflecting the essentials of social infrastructures and the capacity to automatically find solutions, remedies or repairs based on diagnostic modules.

References

[1] Nick Szabo. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, 18:2, 1996.

[2] Luc Desrosiers and Ricardo Olivieri. Extend your blockchain smart contracts with off-chain logic, 2018.

[3] Michele Ruta, Floriano Scioscia, Saverio Ieva, Giovanna Capurso, Agnese Pinto, and Eugenio Di Sciascio. A blockchain infrastructure for the semantic web of things. In *SEBD*, 2018.

[4] Huan Zhou, Xue Ouyang, Jinshu Su, Cees de Laat, and Zhiming Zhao. Enforcing trustworthy cloud sla with witnesses: A game theory–based model using smart contracts. *Concurrency and Computation: Practice and Experience*, page e5511, 2019.

[5] Clément Lesaege, Federico Ast, and William George. Kleros: Short paper v1.0.7. 2019. https://kleros.io/assets/whitepaper.pdf.

[6] L. Thomas van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. eflint: a domain-specific language for executable norm specifications. In *Proceedings of GPCE '20. ACM*, 2020.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200873

Towards Transparent Human-in-the-Loop Classification of Fraudulent Web Shops

Daphne ODEKERKEN [a,b] and Floris BEX [a,c]
[a] *Department of Information and Computing Sciences, Utrecht University*
[b] *National Police Lab AI, Netherlands Police*
[c] *Tilburg Institute for Law, Technology and Society, Tilburg University*

Abstract. We propose an agent architecture for transparent human-in-the-loop classification. By combining dynamic argumentation with legal case-based reasoning, we create an agent that is able to explain its decisions at various levels of detail and adapts to new situations. It keeps the human analyst in the loop by presenting suggestions for corrections that may change the factors on which the current decision is based and by enabling the analyst to add new factors. We are currently implementing the agent for classification of fraudulent web shops at the Dutch Police.

Keywords. law enforcement, dynamic argumentation, legal case-based reasoning

1. Introduction

Every year, the Dutch police receives thousands of complaints on online trade fraud, many of which concern reports on web shops that do not deliver goods. Nonetheless, not each of these shops has bad intentions: in many cases, the customer fell victim to malfunctioning delivery service, rather than fraud. The Dutch police has a national centre for counteracting online trade fraud, where analysts manually check suspicious web shops. This is a combination of routine work (that could be automated) and more detailed investigation (that should be done by humans). Given the high number of suspicious web shops and the necessity to act quickly, the police experiments with using artificial intelligence (AI) agents to speed up the process. In this paper, we introduce a new agent architecture for web shop classification that relies on static and dynamic algorithms for both rule-based and case-based reasoning. On first thought, this may seem an overly complex solution, given that classification problems are often solved in machine learning by training a model on a labelled data set. However, this classical machine learning approach does not suffice for classification problems in law enforcement, for three reasons.

Handling a dynamic environment. Recently, Wabeke et al. [5] presented their multiyear effort in detecting and removing counterfeit web shops from the .nl DNS zone - a problem similar to ours. They developed two detection systems. Interestingly, one of the claims in their paper is that the makers of counterfeit web shops adapted to their first system. This makes clear that a web shop classifier should be able to adapt to its environment. This issue can be handled by frequently updating the model, but that would require continuous effort from an AI expert. Instead, we aim for a more future-proof solution that directly takes input from analysts into account, as we will show in Section 3.

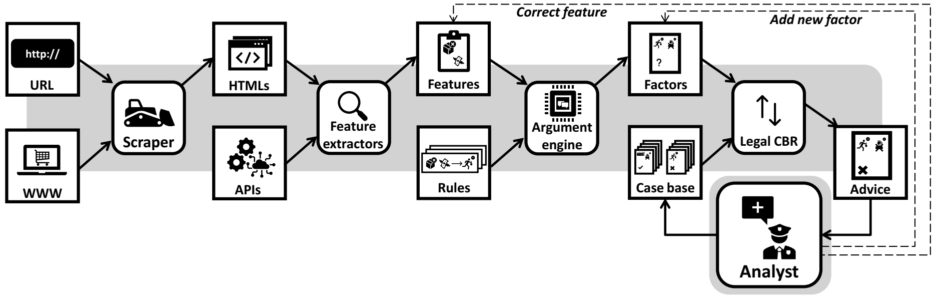


Figure 1. Web shop checker architecture.

Human in the loop. The classification outcome has serious implications: web shops classified as mala fide will be taken offline, while web shops classified as bona fide can be placed on a white list for fraud intake (see [3]). In view of this, it is required that a human analyst checks each advice given by the classifier. However, we should be alert to the control problem [6], i.e. the situation that a human analyst devolves too much responsibility to the classifier and fails to detect cases where the classifier is wrong. To prevent this, the analyst should be kept actively in the loop: he or she should for example be notified of possible mistakes by the classifier and be encouraged to check these situations. An additional motivation for a human-in-the-loop approach is that some factors influencing the decision can only be found by a manual investigation, for example since they require making a payment. In cases where these factors could be relevant, the analyst should be invited to investigate these factors and return the resulting information to the agent.

Transparent. Currently, the result of a web shop check by human analysts is a well-founded advice that includes the factors that made them decide on their conclusion. This is required for various purposes, e.g. alerting citizens and informing the registrar in a web site take-down request, so our agent should be able to produce similar explanations. In general, transparency is one of the key requirements for trustworthy AI applications, as identified by the European Commission's High-Level Expert Group on AI[1].

2. Web shop classification agent architecture: from URL to initial advice

Our proposed agent architecture is illustrated in Figure 1. The initial input is a URL that is considered suspicious. As a first step, a **web scraper** scrapes the web shop's HTML pages. Subsequently, features are extracted from the HTML by **feature extractors**. Some feature extractors require **API** calls to obtain additional information from external organisations. The resulting feature vector is the input for the argumentation engine.

The **argumentation engine** uses a set of defeasible rules to find arguments for or against factors that influence the decision if a web shop should be trusted. Factors are either bona fide (e.g. *"uses https"*) or mala fide (e.g. *"uses fake hallmark logo"*) and are identified by consulting analysts. We use an ASPIC$^+$ [4] implementation that applies rules to features, thus obtaining arguments that support and attack factors; see Figure 2 for an example. Given a set of arguments and the attack relation between them, the argumentation engine determines the set of acceptable arguments by computing the grounded

[1] https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines

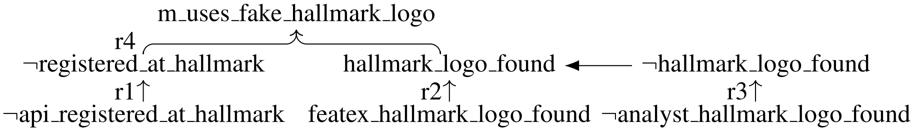

m_uses_fake_hallmark_logo
r4
¬registered_at_hallmark
hallmark_logo_found
¬hallmark_logo_found
r1
r2
r3
¬api_registered_at_hallmark
featex_hallmark_logo_found
¬analyst_hallmark_logo_found

Figure 2. Excerpt from the rule set. If a feature extractor found a hallmark logo at the web site, but an API call returns that this site is not registered at the hallmark company, there is an argument for the mala fide factor “uses fake hallmark logo”. Rule r3 is stronger than r2, so if an analyst could not find the logo, then the argument for m_uses_fake_hallmark is attacked on hallmark_logo_found - and removed from the grounded extension.

extension [1]. The grounded extension identifies arguments that can reasonably be accepted; hence all factors for which there is an argument in the grounded extension can reasonably be taken into account in the final decision. The output of the argumentation engine is the set of factors for which there is an argument in the grounded extension.

Finally, the **legal case-based reasoning (CBR)** module compares the factors of the tested web shop to a case base of earlier ⟨factor set, conclusion⟩ pairs. It identifies precedential constraints [2] based on a fortiori reasoning: a web shop is constrained to be mala fide if its factors are at least as “bad” as those of a precedent case labelled mala fide (since all mala fide factors of the precedent case apply to our web shop and all bona fide factors of our web shop apply to the precedent case). Similarly, our web shop is constrained to be bona fide if its factors are at least as “good” as those of a precedent case labelled bona fide. If no precedential constraint applies, the tested web shop is labelled undecided. This way, we obtain an initial advice (bona fide, mala fide or undecided) for our web shop.

3. Interaction between the agent and the analyst in the loop

As shown in Figure 1, the human analyst interacts with the agent in four different ways.
Explanation. The agent explains its initial advice to the analyst. This explanation consists of the factors corresponding to the web shop, together with a precedent case from the case base for which a precedential constraint applies, see Figure 3. If required, a more detailed explanation can be constructed by generating the arguments for these factors.
Correcting features. We define the rule set in such a way that the analyst can overrule feature extractors in stating that a feature is present or absent. Such a correction could lead to a change in the present factors, which may influence the advice. By using a variation on the stability algorithm from [3], we can identify which features can still be obtained by an analyst check and would result in a factor change that alters the advice. These features are presented as suggestions to the analyst, as shown in Figure 3.
Adding factors. Alternatively, the analyst may not agree with the advice since some factor is missing. In that case, he or she can add this factor to the case. This information is stored, so that the analyst implicitly constructs a data set that can eventually be used by an AI expert to develop new feature extractors and argumentation rules for this factor.
Case base update. For web shops that cannot be assigned bona fide or mala fide by some precedential constraint, the advice will be undecided. In this case, the analyst chooses between bona fide and mala fide (based on the factors) and adds this new case to the case base. Note that this cannot cause inconsistencies in the case base, since no precedential constraint applied before. Thanks to these continuous updates of the case base, the agent will be able to classify more web shops as bona fide or mala fide in the future.

Based on automatically extracted information, the web shop www.suspicious-shop.com seems to be bona fide. This advice is based on following factors:

- The Chamber of Commerce number mentioned on the web site exists;
- The VAT number on the web site is valid.

This advice is based on a comparable advice for the web shop www.bona-fide-shop.com. However, the following information would change the advice:

- Payments are transferred to a foreign bank account.
  This mala fide factor can be obtained by making a payment.

Figure 3. Example of explanation for a new advice, based on the factors of an old advice.

4. Discussion and conclusion

We proposed an agent architecture for transparent human-in-the-loop classification that combines dynamic structured argumentation with legal case-based reasoning. This way, it can explain its decisions by the contributing factors and previous cases. Thanks to continuous updates on the case base, it adapts to new situations. Finally, it keeps the human analyst actively involved in the loop by presenting suggestions for analyst checks and enabling the analyst to add new factors that can change the classification outcome.

This agent is currently being implemented for the classification of fraudulent web shops at the Dutch Police. In order to efficiently estimate which features and factors could still change the advice, we work on an extension of our stability algorithm [3].

The implementation of the proposed system requires a significant amount of knowledge engineering, since the rules for the argumentation engine are identified manually. We consider this effort to be more than worthwhile, since the rule set provides a way to generate human-readable explanations; furthermore, it is required to run algorithms for dynamic argumentation [3]. Finally note that many rules can be obtained easily since they fit in a certain scheme - for example, some feature is present if it is detected by a feature extractor or if it is observed by an analyst, see the rules for hallmark_logo_found in Figure 2. In case of conflict, the rule based on the analyst's observation is stronger.

Although the agent architecture is designed for the law enforcement domain, it could also be used for transparent human-in-the-loop classification in other domains - provided that one can identify factors that correspond to one of the two classes. Finally, we only used binary factors, but we plan to extend our approach towards dimensions.

References

[1] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[2] John F Horty. Rules and reasons in the theory of precedent. *Legal Theory*, 17:1–33, 2011.

[3] Daphne Odekerken, AnneMarie Borg, and Floris Bex. Estimating stability for efficient argument-based inquiry. In *Proceedings of the 8th International Conference on Computational Models of Argument*, 2020.

[4] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[5] Thymen Wabeke, Giovane Moura, Nanneke Franken, and Cristian Hesselman. Counterfighting counterfeit: detecting and taking down fraudulent webshops at a ccTLD. In *Proceedings of the 21st International Conference on Passive and Active Network Measurement*, pages 158–174. Springer, 2020.

[6] John Zerilli, Alistair Knott, James Maclaurin, and Colin Gavaghan. Algorithmic decision-making and the control problem. *Minds and Machines*, 29(4):555–578, 2019.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200874

From Prescription to Description: Mapping the GDPR to a Privacy Policy Corpus Annotation Scheme

Ellen POPLAVSKA [a], Thomas B. NORTON [b], Shomir WILSON [a], and Norman SADEH [c]

[a] *Pennsylvania State University, University Park, Pennsylvania, USA*
[b] *Fordham University School of Law, New York, New York, USA*
[c] *Carnegie Mellon University, Pittsburgh, Pennsylvania, USA*

Abstract. The European Union's General Data Protection Regulation (GDPR) has compelled businesses and other organizations to update their privacy policies to state specific information about their data practices. Simultaneously, researchers in natural language processing (NLP) have developed corpora and annotation schemes for extracting salient information from privacy policies, often independently of specific laws. To connect existing NLP research on privacy policies with the GDPR, we introduce a mapping from GDPR provisions to the OPP-115 annotation scheme, which serves as the basis for a growing number of projects to automatically classify privacy policy text. We show that assumptions made in the annotation scheme about the essential topics for a privacy policy reflect many of the same topics that the GDPR requires in these documents. This suggests that OPP-115 continues to be representative of the anatomy of a legally compliant privacy policy, and that the legal assumptions behind it represent the elements of data processing that ought to be disclosed within a policy for transparency. The correspondences we show between OPP-115 and the GDPR suggest the feasibility of bridging existing computational and legal research on privacy policies, benefiting both areas.

Keywords. privacy, privacy laws, privacy policies, theory, annotation, GDPR, General Data Protection Regulation

Introduction

In 2018, the GDPR entered into force, becoming one of the most influential privacy laws to date. As a result, businesses and organizations were required to change their privacy protocols to comply. For many, these changes included changes to the privacy policies provided to users. In particular, many businesses and organizations were compelled to update their privacy policies to state specific information about their data practices.

Recent efforts in natural language processing (NLP) have addressed the demand for automatic information extraction from privacy policies to ease legal analysis and build privacy-enhancing consumer technologies [15,8,4]. This work requires the creation of privacy policy corpora that contain annotations identifying salient details about privacy practices. Currently, the most extensive text annotation scheme dedicated to privacy policies is the OPP-115 annotation scheme [14], which was initially created for a corpus of 115 annotated privacy policies. This corpus now appears in several projects as part of tasks to extract information from privacy policies [2,3,9,10]. The annotation scheme was created to be agnostic to particular laws, instead concentrating on a general concept of privacy practices, or activities that an organization may perform with customers' information. Determining the relevance of OPP-115 to the GDPR clarifies how well existing work based upon this annotation scheme addresses the concerns of modern privacy law.

We perform a comparative study of the OPP-115 annotation scheme with the GDPR Article 5 principles for processing of personal data, as well as other relevant articles of the GDPR, identifying matches and mismatches between these two systematizations. We show strong connections between the two, validating OPP-115's applicability and the relevance of NLP research that continues to use the annotation scheme. We release our dataset of connections between the GDPR and OPP-115 to promote further NLP research to automatically identify connections between privacy policies and privacy law.[1]

1. Related Work

1.1. OPP-115 and its Uses

The Online Privacy Policies, Set of 115 (OPP-115) Corpus released by Wilson et al. [14] contains 115 privacy policies annotated by law students. It provides an annotation scheme of ten mutually exclusive categories into which segments of privacy text, known as *data practices*, may be sorted. The OPP-115 corpus and its annotation scheme have been utilized by other privacy researchers. Sathyendra et al. [9] used the corpus to train models to extract opt-out choices from privacy policies. Harkous et al. [2] used the corpus to classify privacy practices and answer non-factoid questions. Story et al. [10] used the corpus to automatically identify opt-out choices on websites and locate potential noncompliance. Mousavi et al. [3] used the corpus to predict categories for paragraphs of privacy text. Researchers have continued to use this annotation scheme to represent the structure of a standard privacy policy. To date, however, there has been no published work analyzing how accurately the OPP-115 categories represent privacy legislation.

1.2. Computational Uses of the GDPR

Since the GDPR came into effect, researchers have considered methods to determine compliance. Truong et al. [13] have envisioned a personal data management platform designed around GDPR compliance. Tesfay et al. [11] have created PrivacyGuide, a tool that classifies privacy policy content into eleven aspects constructed around GDPR compliance. Torre et al. [12] have created a UML representation of the GDPR as a first step towards automated compliance checking. Palmirani et al. [5] have proposed a framework for modelling legal documents for compliance checking. Palmirani et al. [6] have developed PrOnto, a privacy ontology modelling the conceptual cores of the GDPR. Bonatti et al. [1] have created the SPECIAL Usage Policy Language to describe cores of GDPR-compliant usage policies. Polleres et al. [7] have created the Data Privacy Vocabulary to describe and categorize GDPR-compliant personal data handling. In contrast with others' work, ours fills a theoretical gap between privacy policy annotations and uses of AI and NLP on privacy policies. Additionally, the OPP-115 annotation scheme's use beyond one project motivates further examination of how it connects with specific privacy laws.

[1] usableprivacy.org/data/

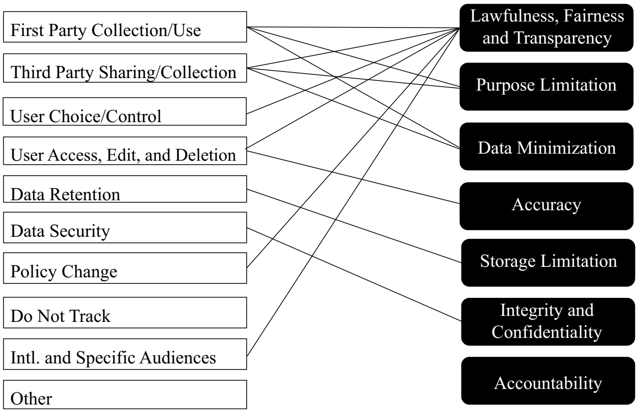

First Party Collection/Use
Third Party Sharing/Collection
User Choice/Control
User Access, Edit, and Deletion
Data Retention
Data Security
Policy Change
Do Not Track
Intl. and Specific Audiences
Other
Lawfulness, Fairness and Transparency
Purpose Limitation
Data Minimization
Accuracy
Storage Limitation
Integrity and Confidentiality
Accountability

Figure 1. OPP-115 categories, left, connected to principles from GDPR Article 5, right.

2. Approach

In Article 5, the GDPR details a set of principles for data processing, which provide an overview of the regulation's expectations for data controllers and processors. We compare these principles to the categories of OPP-115, which represent the most general level of the annotation scheme, and identify thematic connections. These connections represent instances when the principles and categories codify the same expectations (prescriptive and descriptive, respectively) for the contents of privacy policies. We also create a dataset of the connections between the 99 articles of the GDPR and the categories of OPP-115. In developing these associations, we consider the definitions of each category of OPP-115, the descriptions of the articles, the audience of each particular article, and whether the concepts described in a particular article might belong in a privacy policy.

3. Results and Discussion

Of the 99 articles, we find associations with categories of OPP-115 within 49. We find a total of 88 connections between GDPR articles and OPP-115 categories. 78 of these occur within the first five chapters of the GDPR, suggesting that some chapters contain more pertinent privacy policy details than others. Most articles are associated with multiple categories. The median number of connections for an article is two, demonstrating that the concepts within each article are usually applicable to multiple categories and that GDPR concepts overlap considerably across sections. Figure 1 displays connections between OPP-115 categories and GDPR principles. These represent thematic similarities between the concepts guiding the GDPR and the categories for data practices described by OPP-115. We release the full set of connections in CSV format for further research.

These connections and gaps between the OPP-115 annotation scheme and the GDPR reflect the similarities and differences between what privacy experts believed were the essential components of privacy policies in 2016 and the codified European privacy regulation of 2018. These give insight as to how accurately OPP-115 legal scholars' observations reflect today's legislative understanding of privacy concepts. Comparing the principles of the GDPR to the categories of data practices in OPP-115, it is apparent that legal scholars' decisions about categories of data practices are similar to legislators' descriptions of similar concepts. While OPP-115 separates First Party Collection/Use and Third Party Sharing/Collection, the GDPR presents principles that apply to all data processing by controllers and processors. This may reflect the fact that OPP-115 was created to sort data practices in privacy policies, where first-party and third-party processing are often listed in distinct sections, while the GDPR provides guidance for all data processing.

In addition to revealing how the legal insights behind OPP-115 reflect recent privacy regulation, this work demonstrates how accurately the OPP-115 corpus and annotation scheme currently used by researchers represent it. This allows researchers to contextualize their results within a set of principles similar to those represented in the regulation.

4. Acknowledgements

This work was supported in part by the US National Science Foundation under Grants #CNS-1914444, #CNS-1914446, and #CNS-1914486, as well as the PA Space Grant Consortium Research Internship Program.

References

[1] P. Bonatti, S. Kirrane, I. M. Petrova, L. Sauro, and E. Schlehahn. *The SPECIAL Usage Policy Language*, 2019. https://ai.wu.ac.at/policies/policylanguage/.

[2] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *Proc. USENIX Security*, 2018.

[3] N. Mousavi, D. Graux, and D. Collarana. Towards measuring risk factors in privacy policies. In *AIAS@ICAIL*, 2019.

[4] A. Oltramari, D. Piraviperumal, F. Schaub, S. Wilson, S. Cherivirala, T. B. Norton, N. C. Russell, P. Story, J. Reidenberg, and N. Sadeh. Privonto: A semantic framework for the analysis of privacy policies. *Semantic Web*, 2017.

[5] M. Palmirani and G. Governatori. Modelling legal knowledge for GDPR compliance checking. In *Proc. JURIX*, 2018.

[6] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo. Legal ontology for modelling GDPR concepts and norms. In *JURIX*, 2018.

[7] A. Polleres, B. Bos, B. Bruegger, E. Kiesling, E. Schlehahn, F. Ekaputra, H. Pandit, J. Fernández, M. Lizar, and R. Hamed. *Data Privacy Vocabulary (DPV)*, 2020. https://dpvcg.github.io/dpv/.

[8] A. Ravichander, A. W. Black, S. Wilson, T. B. Norton, and N. Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. In *Proc. EMNLP-IJCNLP*, 2019.

[9] K. M. Sathyendra, F. Schaub, S. Wilson, and N. Sadeh. Automatic extraction of opt-out choices from privacy policies. In *Proc. AAAI Symposium on Privacy-Enhancing Technologies*, AAAI Fall Symposium - Technical Report, 2016.

[10] P. Story, S. Zimmeck, A. Ravichander, D. Smullen, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh. Natural language processing for mobile app privacy compliance. In *Proc. PAL*. CEUR Workshop Proceedings, 2019.

[11] W. B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, and J. Serna. PrivacyGuide: Towards an implementation of the EU GDPR on internet privacy policy evaluation. In *Proc. IWSPA*, 2018.

[12] D. Torre, G. Soltana, M. Sabetzadeh, L. Briand, Y. Auffinger, and P. Goes. Using models to enable compliance checking against the GDPR: An experience report. In *Proc. MODELS*, 2019.

[13] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo. GDPR-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 2020.

[14] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. Giovanni Leon, M. Schaarup Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh. The creation and analysis of a website privacy policy corpus. In *Proc. ACL*. Association for Computational Linguistics, 2016.

[15] S. Wilson, F. Schaub, R. Ramanath, N. Sadeh, F. Liu, N. Smith, and F. Liu. Crowdsourcing annotations for websites' privacy policies: Can it really work? In *Proc. WWW*, 2016.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200875

Summarisation with Majority Opinion

Oliver RAY, Amy CONROY, and Rozano IMANSYAH
Department of Computer Science, University of Bristol
e-mail: {csxor, ac16888, lv18263}@bristol.ac.uk

Abstract. This paper introduces a method called SUmmarisation with Majority Opinion (SUMO) that integrates and extends two prior approaches for abstractively and extractively summarising UK House of Lords cases. We show how combining two previously distinct lines of work allows us to better address the challenges resulting from this court's unusual tradition of publishing the opinions of multiple judges with no formal statement of the reasoning (if any) agreed by a majority. We do this by applying natural language processing and machine learning, Conditional Random Fields (CRFs), to a data set we created by fusing together expert-annotated sentence labels from the HOLJ corpus of *rhetorical role summary relevance* with the ASMO corpus of *agreement statement* and *majority opinion*. By using CRFs and a bespoke summary generator on our enriched data set, we show a significant quantitative F1-score improvement in rhetorical role and relevance classification of 10-15% over the state-of-the-art SUM system; and we show a significant qualitative improvement in the quality of our summaries, which closely resemble gold-standard multi-judge abstracts according to a proof-of-principle user study.

Keywords. Legal Summarisation, UK House of Lords (UKHL), Machine Learning.

1. Introduction

The summarisation of legal judgments is a challenging task [1] especially in courts like the UK House of Lords (UKHL) which publish the opinions of multiple judges with no formal statement of the reasoning (if any) agreed by a majority [2,3]. The aim of this work is to automatically generate multi-judge summaries that closely resemble gold-standard abstracts published in the Incorporated Council of Law Reporting (ICLR) Daily Law Reports (DLR). We achieve this goal by integrating and extending two previously independent lines of work applying computational methods to UKHL case law [4,5].

First, we create an enriched data set of UKHL cases by fusing expert-annotated sentence labels from the HOLJ corpus of [4], which marks up the *rhetorical role* and *summary relevance* of sentences, together with the ASMO corpus of [5], which marks up explicit inter-judge *agreement statements* and *majority opinions*. Then we implement a new summary pipeline, called SUmmarisation with Majority Opinion (SUMO) that uses natural language processing and Conditional Random Fields (CRFs) to generate better quality summaries than the previous state-of-the-art system, *SUM* [4].

The main benefits of SUMO over SUM are that: (i) we increase the rhetorical role and relevance classification F1-scores by 10-15% (to about 75% and 40%, respectively); (ii) we supplement extractively generated case abstracts with abstractively generated inter-judge agreement summaries in the DLR style; and (iii) we demonstrate superior quality using both ROUGE metrics and expert feedback from a preliminary user study.

2. Background

The UKHL, or UK Supreme Court (UKSC) since 2009, differs from most other courts by publishing judgments that consist of the seriatim opinions of multiple law lords (usually 5 from a panel of 12) with no accompanying statement of consensus (if one even exists) on the *ratio decidendi*. And, while the judges always return a *majority decision* (to allow or dismiss an appeal), a binding precedent is only set by a *majority opinion* (where more than half also agree on the legal reasons) [2]. Thus, judges usually discuss drafts of their speeches with each other and often state (dis)agreements with their peers in the final judgment. But, in practice, even UKHL/UKSC judges recognise that it can be very hard to determine when a majority opinion exists [3]. As a result of this unique challenge, there is very little prior research on the automatic summarisations of UKHL cases. In fact, we found just two lines of work - that we integrate and extend in this paper.

The first strand of work is the SUM system [4] which generates extractive summaries by classifying sentences according to their rhetorical role (**Facts**, **Proceedings**, **Background**, **Framing**, **Disposal**, **Textual** and **Other**) and classifying sentences as *relevant* to the summary. They introduced the HOLJ corpus which marks up the sentences of 47 UKHL cases with expert-annotated labels indicating their main rhetorical role and to which (if any) of the DLR gold-standard summary sentences they most closely align. The sentences are also marked up with machine-generated labels denoting linguistic features like sentence length and location, named entities, quoted text, thematic words and cue phrases. These were used to train two classifiers which achieved F1-scores of 61.2% for role and 31.2% for relevance; and these predictions were then used to extract summary sentences more effectively than a variety of baseline methods.

The second strand of work is the ASMO system [5] which identifies explicit inter-judge (dis)agreement statements and uses them to infer the existence of incontestable majority opinions. They introduced the ASMO corpus which marks up the sentences in a superset of 300 UKHL cases with expert-annotated labels identifying **acknowledgements**, **outcomes**, various types of (dis)agreement (**Full**, **Partial**, **Order**, **Generic** and **Self**), along with the set of judges (if any) whose reasoning forms the **majority opinion**. The sentences are also marked up with machine-generated labels (inspired by HOLJ) denoting length and location, unigrams and POS tags, named entities and a set of hand-crafted cue phrases. These were used train a classifier which detects full agreement statements with an F1-score of 94.3% and uses them to infer incontestable majority opinions with an F1-score of 81%.

3. Summarisation by Majority Opinion (SUMO)

We began by combining the expert labels from HOLJ and ASMO to create an enriched UKHL corpus. Due to the differences in case identification and sentence splitting, this required a non-trivial alignment and merging process [6]. We used normalised variants of sentence length and location, and quotations and cue phrases as our feature-sets. We also identified generic named entities using spaCy[1] and legal entities using ICLR&D's Blackstone[2]. This resulted in 7 feature-sets which we used to train our rhetorical and

[1] https://spacy.io/ [2] https://github.com/ICLRandD/Blackstone

relevance classifiers (using predicted role as an extra feature when training the relevance classifier).

We developed our *SUMO* pipeline in Python using a combination of shallow natural language processing and supervised machine learning [7]. Our approach uses a multi-class rhetorical classifier (to predict the role of a sentence) as well as a binary relevance classifier (to predict if it aligns to a sentence in the summary). We trained the model by splitting the corpus in to self-contained speeches rather than whole judgments, as we hypothesised this would help our sequential modelling method to exploit the overall structure of each lord's speech without being confused by transitions between speeches.

We performed the classifications tasks using the novel approach of applying CRFs to summarise legal texts, previously attempted by only one piece of work [8]. CRFs avoid biases evident in other sequence models such as Maximum-entropy Markov models by using a single exponential model to determine the probability of the entire sequence of the labels. We extract the marginal probability from the relevance classifier to assign a ranking to each sentence as to *how* summary-worthy the sentence is. This give us more flexibility to create summaries of arbitrary lengths depending on the needs of the user. We combine this data with the rhetorical role to output structured summaries in the same style as the DLR gold standard summaries.

In order to replicate the manually written statements from the DLR summaries that indicate agreement between lords, we use the data from the ASMO system to identify the agreements as well as who formed the majority opinion (see [7]). This meant that our summaries include representative sentences such as: *"...* **LORD SLYNN** *and* **LORD STEYN. LORD MILLETT** *and* **LORD PHILLIPS** *delivered an opinion agreeing with* **LORD SLYNN** *and* **LORD STEYN. LORD HOPE** *did not agree with the line of reasoning..."* We combine this information with the rhetorical roles predicted by our system to select the highest ranking sentences and create a structured summary in the same style as the ICLR gold standard. This goes beyond the simple ranking only summary produced by the SUM system.

4. Results and Evaluation

Using our methodology we are able to achieve a weighted average F1-score for our rhetorical classifier of 77.8%, with RandomizedSearchCV utilised to validate our results. This is a 16.6% increase over SUM's rhetorical classifier. Our relevance classifier achieves a binary-averaged F1-score of 42.1%, validated using the same methodology as our rhetorical classifier. This is a 10.9% increase over the SUM system's relevance classifier.

Evaluation of automatically generated texts and in particular of summaries can be very difficult, largely due to the subjective nature of summaries. We use the ROUGE 2.0 toolkit[3] to quantitatively evaluate the summaries produced by our system. We compare the results of the *SUMO* system with a summary generated using the same methodology as the SUM system. The ROUGE-1 F1-score results indicate that the summary produced by SUMO (48.9%) perform better than summaries produced using the SUM methodology (37.6%) as well as the baseline summary (41.9%). Our use of the majority opinion to abstractively generate the agreement sentences that closely resembles the manually written summaries likely contributes to a higher F1-score.

[3] https://github.com/kavgan/ROUGE-2.0

As the ROUGE metrics are not necessarily indicative of a good summary, we balanced this evaluation with a user study. We recruited 8 experts (individuals with UK legal experience, either as an LLB student or graduate and/or as a legal professional), and 10 non-experts to complete our study, which was an online survey. The study evaluated our *SUMO* summary compared to the corresponding ICLR summary across three randomly selected judgments, evaluated using questions in the form of 7-point Likert scales. 81.5% of our participants agreed that our summary was a valid replacement for the ICLR gold standard, and 83.3% agreed that it contained the most important aspects of the case.

One notable comment from one of our evaluators indicated confusion regarding our use of the word agreement. While the summary states that the lord did not agree with the *line of reasoning* of his fellow lords, the first disposal sentence we extracted from him details that he agreed with his fellow lords that the outcome should be dismissed. This shows an interesting observation between the agreement as to the outcome and agreement of the line of reasoning of his fellow lords, a distinction that indicates whether the line of reasoning forms a precedent in common law systems or not.

5. Conclusion

The *SUMO* system introduced in this paper sets a new benchmark for the automatic summarisation of legal judgments in the UK. By applying CRFs to summarise legal texts, as well as introducing a new type of ASMO feature, we improve the F1-scores of the rhetorical role and summary relevance prediction tasks by 10-15% over previous research. We further exploited ASMO features in order to abstractively generate parts of the summary, which based on the ROUGE metrics and positive user feedback indicate a close resemblance to the gold-standard text.

For future work we are developing an NLP method for inferring the decisions of individual sentences from outcome statements (which an analysis of numerous problematic cases shows is not as trivial as it may first seem). This could help us address another important task, revealed by our user feedback, of automatically resolving the ambiguity often associated with different intended uses of the word ‘agreement’: such as in the DLR summaries where it is used loosely, variously referring to reasons, outcomes and orders, or just facts and issues.

References

[1] A Kanapala et al. Text summarization from legal documents: a survey. *Artif Intell Rev (2019) 51: 371–402.*

[2] G Williams. *Learning the Law*. Sweet & Maxwell, 14 edition, 2010.

[3] B Hale. Judgment Writing in the Supreme Court – UK Supreme Court Blog (October), 2010.

[4] B Hachey and C Grover. Extractive Summarisation of Legal Texts. *AI and Law*, 14(4):305–345, 2006.

[5] J Valvoda et al. Using Agreement Statements to Identify Majority Opinion in UKHL Case Law. In *Proc. 31st Int. Conf. on Legal Knowledge and Info. Sys.*, Frontiers in AI and Applications (313): 141-150, 2018.

[6] R Imansyah. Predicting the Role and Relevance of Sentences in UK House of Lord Judgements. Master’s thesis, University of Bristol, Bristol, UK, 2019.

[7] A Conroy. SUMO: A System for Automatically Summarising UK House of Lords (UKHL) Judgments using Majority Opinion. Master’s thesis, University of Bristol, Bristol, UK, 2020, submitted.

[8] M Saravanan et al. Improving Legal Document Summarization Using Graphical Models. *Frontiers in AI and Applications (152): 51-59*, 152:51, 2006.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200876

A Common Semantic Model of the GDPR Register of Processing Activities

Paul RYAN[ac,1] and Harshvardhan J. PANDIT [b] and Rob BRENNAN [a]
[a] *ADAPT, School of Computing, Dublin City University, Dublin 9, Ireland*
[b] *ADAPT, Trinity College Dublin, Dublin 2, Ireland*
[c] *Uniphar PLC, Dublin, Ireland*

Abstract. The creation and maintenance of a Register of Processing Activities (ROPA) is an essential process for the demonstration of GDPR compliance. We analyse ROPA templates from six EU Data Protection Regulators and show that template scope and granularity vary widely between jurisdictions. We then propose a flexible, consolidated data model for consistent processing of ROPAs (CSM-ROPA). We analyse the extent that the Data Privacy Vocabulary (DPV) can be used to express CSM-ROPA. We find that it does not directly address modelling ROPAs, and so needs additional concept definitions. We provide a mapping of our CSM-ROPA to an extension of the Data Privacy Vocabulary.

Keywords. GDPR, Regulatory Compliance, Semantic Web

1. Introduction

A Register of Processing Activities (ROPA) is a comprehensive record of the personal data processing activities of an organisation. It is central to meet the principle of accountability as set out in Article 30 of the GDPR. Organisations most commonly create and maintain ROPAs through informal tools and spreadsheets[2]. EU Data Protection Regulators also seem to encourage this practice by providing spreadsheet-based templates to assist organisations in preparing and maintaining ROPAs. A spreadsheet, while being a simple and commonly utilised versatile medium, requires effort to enter information and keep it updated. As a human-oriented application, spreadsheets often lack the rich data structures and semantics that are suitable for building automated toolchains, especially when modelling complex legal concepts. The creation of a common data model is required to represent ROPA information across different compliance-related processes and act as the connection between an organisation's internal compliance data and what regulators would expect. This model can be used to fuse information stored in spreadsheets and facilitate the interconnectivity of data processing systems with ROPA-maintenance/compliance systems, automatically update spreadsheets and automated querying, validation, monitoring and reporting of ROPA information. Regulator template consolidation into a semantic model will facilitate an organisation to regulator interoperability; and will provide a single data model for compliance across jurisdictions. The variation of ROPA templates, allied with the option for organisations to develop their own data structures creates significant challenges when

[1] Corresponding Author: Paul Ryan, Email: paul.ryan76@mail.dcu.ie . This work is partially supported by Uniphar PLC. and the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centre's Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

[2] IAPP. https://iapp.org/resources/article/measuring-privacy-operations/.

it comes to compliance automation and tool development. It is possible to resolve this variation with a flexible, unified data model of a ROPA. It could support multi-jurisdiction tool development for ROPA maintenance and RegTech-style automated compliance reporting to regulators, thus reducing costs [1].

Our research aims to enable the creation of technical solutions for the maintenance of ROPAs through semantic web technologies. We show in this paper that for ROPAs, there are differences within the templates provided by each regulator in terms of semantics and granularity - despite being based on common requirements of GDPR's Article 30. There is existing work regarding the use of semantic vocabularies to represent GDPR for various compliance-related tasks. We select DPV [3] [2] as the most comprehensive and representative vocabulary of the Sot A and answer the research question "to what extent can the existing Data Privacy Vocabulary (DPV) be extended to build a semantic ROPA model spanning the range of regulator ROPA templates". To address these issues, we first consolidate the different regulator issued templates into a Common Semantic Model for ROPAs (CSM-ROPA). We then evaluate and extend the DPV for representing CSM-ROPA. The contributions of this paper are (i) analysis of six ROPA templates from EU data protection regulators (ii) a consolidated semantic model of ROPA and (iii) extensions of the DPV for representing a semantic model of a ROPA. The rest of the paper is structured as follows: Section 2 presents an analysis of the ROPA templates provided by EU regulators. In Section 3, we will present our Semantic Model of a ROPA, and provide an evaluation of the DPV to represent CSM-ROPA.

2. Analysis of ROPA templates from EU Data Protection Regulators

We evaluated 6 ROPA templates provided by EU Data Protection Regulators from the jurisdictions of Belgium, Cyprus, Denmark, Finland, Luxembourg and the United Kingdom, selected for their use of the English language. Each was evaluated in terms of file format, the number of fields, relationship with GDPR Article 30 requirements fields, and controlled vocabularies provided. Our analysis also considered guidance documents or pre-populated samples provided by regulators. We found that all six templates meet the minimum GDPR Article 30 requirements by containing the 12 mandatory information fields it requires, but there was variation in the way they modelled each field. The key differences between the templates arise from the extent of data gathered through the information fields. Three ROPA templates (Finland, Denmark and Luxembourg) are direct transcriptions from Article 30 of the GDPR, containing only the 12 prescribed input fields. The other regulators' (Belgium, United Kingdom and Cyprus) ROPA templates have additional information requirements with varying complexity of the information required. The Belgian ROPA also contains a detailed controlled vocabulary of potential inputs for some fields. In the next step in our analysis, we carried out a systematic review of the concepts included in the six templates. We identified synonyms, overlapping and related concepts. We made direct relationships such as composition or qualifications such as domain and range that were implicit in the spreadsheets. We derived 43 unique concepts representing a consolidated ROPA template covering all six jurisdictions. Based on the interpretation of the GDPR and the use of concepts in ROPA, we combined these 43 concepts into the UML model represented in Figure 1.

[3] https://w3.org/ns/dpv

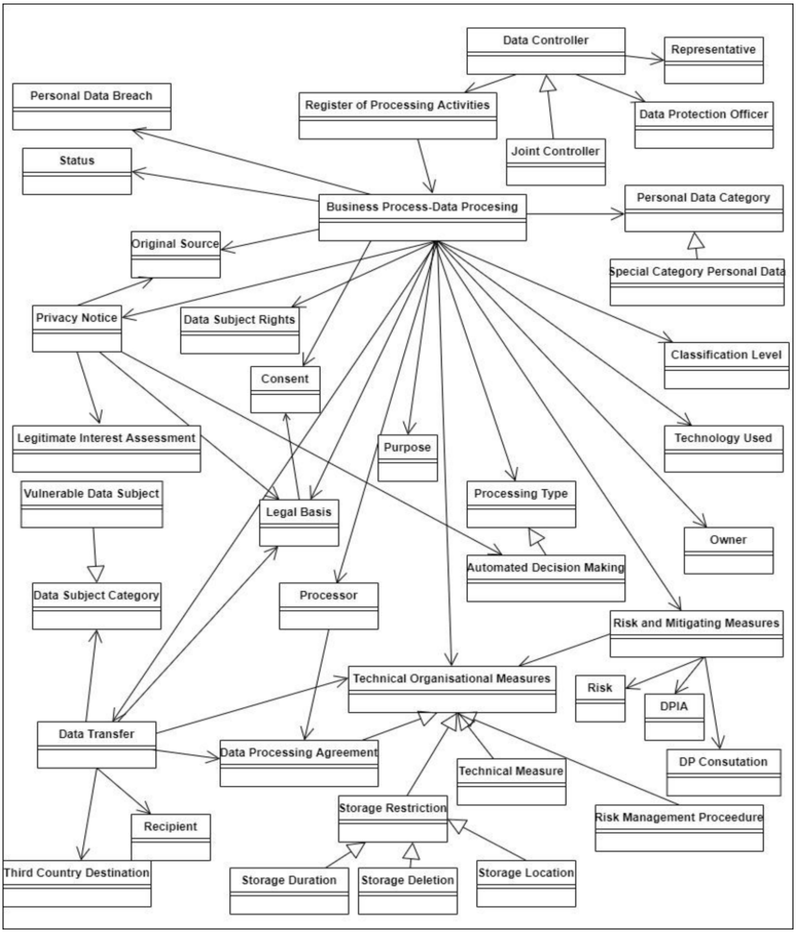


Figure 1. UML Representation of the Combined ROPA Model based on Templates Provided by EU Data Protection Regulators

3. A Common Semantic Model for the ROPA

In order to build the semantic ROPA model, we identified the Data Privacy Vocabulary (DPV)[3] as the most relevant and suitable resource to map our ROPA due to its status as a community specification through the W3C Data Privacy Vocabulary and Controls Community Group (DPVCG). The first task was to establish to what extent DPV could represent the combined ROPA model. We compared the suitability of terms in DPV for representing the 43 unique concepts identified from ROPA templates. Table 1 presents an example of the mapping process. Please refer to [4] for full table is available and analysis. We categorised the mapping as "Exact" if the field exactly corresponds to an existing DPV concept indicating no change required. If the data field has a corresponding concept in the DPV that requires an extension to DPV, we categorised it as 'Partial' match. If the required field can be specified using a combination of multiple concepts in DPV, we categorised the match as 'Complex'. If the concept is missing and needs to be added to the DPV, we categorised the match as 'None' (see Table 1). We found that 14 of our 43 identified unique fields had exact matches with DPV, 15 had partial matches, 3 had complex matches, whereas 11 unique fields had no match within DPV. Thus, the DPV requires the addition of 11 concepts and extension of 15 existing concepts in order to represent information required by the ROPA templates. The additional concepts required are International Transfers, Controller Contact Details, Original Source of Data, Data Protection Officer, Data Protection Impact Assessment, Data Subject Rights, Risk, Privacy Notice, Representative & Data Breach.

Table 1. Extract Taken from Mapping Table ROPA Unique Fields to DPV Concepts [4]

GDPR Regulation	Combined ROPA Model Field	Required GDPR Art. 30	Related DPV Concept	DPV mapping outcome	Combined No. of Specified Field Values vs DPV properties
30.1(b)	Purposes of Processing	Y	dpv: Purpose	Exact	65 /33
13/14/15	Data Subject Rights	N	No DPV Concept	None	
44-47	Transfer to Third Country	N	dpv:LegalBasis	Partial	

Most ROPA templates did not suggest any relationships for ROPA fields. Only 7 of the 43 unique fields specified any properties for ROPA fields. These properties were matched against the DPV. The results are displayed in Table 1 in the column titled "Combined No. of Specified Field Values vs DPV". The DPV will require additional expressiveness here in the form of additional properties to meet the requirements of the ROPA[4]. Alternatively, these additional properties, such as "address" can be met through other standardised vocabularies such as vCard.

Conclusion

Our research analysed six English language ROPA templates provided by EU Data Protection Regulators in terms of information required and relation with requirements of GDPR's. We identified 43 unique concepts to represent a consolidated common model that enables the representation of ROPAs that span multiple jurisdictions. We then evaluated the DPV as representative of the State of the Art and found that it can currently represent only 32 of the 43 concepts of the common semantic model ROPA (CSM-ROPA). We developed an extension to the DPV with missing concepts and a profile of simple and complex correspondences to DPV. In our future work, we will incorporate our work with the DPV standardisation process. We have provided our work to the Data Privacy Vocabulary Community Group [2] (see footnote 4).

References

[1] Ryan P, Crane M, Brennan R. Design Challenges for GDPR RegTech. ICEIS 2020 conference 2020.

[2] Harshvardhan JP, Polleres A, Bos B, Brennan R, Bruegger B, Fajar JE, Fernández, Hamed RG, Kiesling E, Lizar M, Schlehahn E, Steyskal S, Wenning R, 2019. Creating a Vocabulary for Data Privacy. In: On the Move to Meaningful Internet Systems: OTM 2019

[3] Pandit HJ, Polleres A, Bos B, Brennan R, Bruegger B, Ekaputra FJ, et al. Creating A Vocabulary for Data Privacy 2019. p. 17.

[4] Ryan P, Pandit HJ, Brennan R. Towards a Semantic Model of the GDPR Register of Processing Activities, arxiv.org, 2008.00877, 2020.

[4] https://lists.w3.org/Archives/Public/public-dpvcg/2020Sep/0006.html

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200877

Monitoring and Enforcement as a Second-Order Guidance Problem

Giovanni SILENO [a,1], Alexander BOER [b] and Tom VAN ENGERS [a,c]
[a] *Informatics Institute, University of Amsterdam, the Netherlands*
[b] *KPMG, Amsterdam, the Netherlands*
[c] *Leibniz Institute, TNO/University of Amsterdam, the Netherlands*

Abstract. This paper aims to set up a conceptual framework for studying the *second-order guidance problem*—that is, designing coordination mechanisms for autonomous actors by means of adequate monitoring and enforcement measures—in a way which is sensible for designers and users of data-sharing infrastructures such as digital market-places. The paper outlines a minimal, but reusable and extensible computational model to test the sustainability of diverse norm implementations, evaluating it against relevant higher-level models presented in the literature.

Keywords. Monitoring, Enforcement, Reward, Punishment, Non-compliance, Policy design, Policy making.

1. Introduction

Data-sharing infrastructures as *digital market-places* (DMPs) manifestly exhibit the double status of computational and socio-economic systems.[2] On the one hand there are *physical constraints* on the operations that actors can execute; on the other, actors might entertain specific *contractual agreements*, there might be *market rules* and *societal norms* (e.g. GDPR and NIS directive) in place. Research and practice in DMPs are in general dominated by *control-oriented* views (on access and usage control, containment, security, ...). A gap exists in the literature in bridging between the control-oriented and guidance-oriented perspectives in a way which is sensible for designers and users of data-sharing infrastructures. The following example illustrates a possible application:

Example 1 (Coordinating response to cyber-attacks). *Consider a consortium of internet service providers (ISPs). One of the members is under cyber-attack. Information about the attack can be used to coordinate a collective defensive response, of which everybody will be eventually beneficiary. However, releasing information about the attack could provide access to competitive information. Certain parties might decide not to participate for reasons of economic opportunity. Which infrastructural policies should be implemented to promote the correct social functioning?*

Our goal here is to introduce a minimal, but reusable and extensible computational model to test the *sustainability* of certain monitoring and enforcement regimes, and their *effectiveness* with respect to given directives in a certain context.

[1] Corresponding Author: `g.sileno@uva.nl`.

[2] This paper results from work partly conducted for the NWO-funded project DL4LD (*Data Logistics for Logistics Data*, no. 628.001.001), and partly for the NWO-funded program VWDATA.

2. Modelling framework

Norms expressions One of the function of norms is to express *relative preferences* that should guide the behaviour of members of a society, typically of an action over its omission (or vice versa), or of the presence of a certain situation over its absence (or vice versa). This relative preference motivates the introduction of a norm, which can in turn be expressed in different ways. Two prototypical forms can be identified (cf. Hohfeld's framework of normative relationships): as a *deontic directive* (attributing a duty): "*In context C, X has the duty to A, otherwise she will obtain P*", or as a *potestative directive* (assigning a power): "*In context C, X has the power to obtain R by performing A.*" P and R corresponds to two distinct enforcement regimes, based respectively on *punishments* (penalties, negative incentives, or the anecdotal "sticks"), and on *rewards* (positive incentives or "carrots"). Note that providing P and R to X requires the existence of some entity in the social system (typically some authority Y) in the role of enforcer. Both directives can be rephrased without modality, from which we observe that P and R have formally the same role. So it seems that choosing between a carrot and stick regime is an arbitrary choice. But is it?

Theoretical dimensions of norm application Traditional approaches to enforcement take an internal view over the agent, typically based upon utility theory or other decision-making models. The introduction of a reward R and/or a punishment P typically modifies the expected value for the agent X associated to action A. Without enforcement, a rational constraint for deciding towards performance would have been: $\mathbb{E}_X[A] > \mathbb{E}_X[\text{not } A]$; taking into account the enforcement we should consider: $\mathbb{E}^R_X[A] > \mathbb{E}^P_X[\text{not } A]$. (Our reference to utility theory here is just as an illustrative example of internal model). More importantly, in the following we will need to capture only the relative frequency of occurrence of conditions in which the agent X's interests supersede the normative provisions. This measure, denoted with PNC_X, captures the **potential of non-compliance** of X for that norm. PNC_X is computed at individual level, but usually it is presented in aggregated forms, e.g. at population level, here denoted as PNC. It is a crucial *policy-field* value, required, even on a simplistic heuristic basis, to start discussions on any policy design.

Rather than looking at internal models, De Geest and Dari-Mattiacci [1] focus on the external dimensions of norm application, in particular compliance monitoring. Monitoring activity requires resources, and, depending on the context, monitoring for violation or for satisfaction might have different costs and probability of success. This makes the two regimes non-equivalent. As general considerations, the authors observe that sticks usually function better, but there are two cases in which carrots have to be preferred: in presence of a *specification problem* (difficulty of identifying the specific behaviour expected from the addressees); and of a *singling out problem* (non-uniform distribution of the burden over the addressees).

Rules about punishment and reward are conditionals. Someone needs to produce *evidence* of these conditions, even before (non-)compliance can be addressed. Boer [2] suggests that following the flow of evidence provides an alternative, even more essential way to look at the problem. In case of rewards, the agent claiming the reward is the one that has to provide the evidence; in case of punishment, it is the authority. This can be read as directly connected to a *default* rule (in reward-based enforcement, actors are deemed to be generally non-compliant, the opposite with punishment-based enforcement).

Authority	Agent X (addressee)	Collectivity
Monitoring cost: $m_p \cdot P(M) \cdot N$	Certification cost: c_r	Aggregated effects of performance: $(1 - \mathsf{PNC}^{\mathsf{e}}) \cdot P(C) \cdot N \cdot e_*$
Punishment benefit: $-p \cdot N_P$	Punishment cost: p	
Reward cost: $r \cdot N_R$	Reward benefit: $-r$	
	Non-normative effects of performance: e_X	Aggregated effects of non-performance: $\mathsf{PNC}^{\mathsf{e}} \cdot P(C) \cdot N \cdot f_*$
	Non-normative effects of non-performance: f_X	

Table 1. Economic voices distributed across different parties.

Phases of normative interaction We make here explicit the general phases associated to the operationalization of a norm: *performance*, *monitoring*, *enforcement*, and *certification*. Each phase can be played in principle by a different social actor, with different interests and view on the social system. The **applicability context** C in which the norm becomes relevant might have components which are static (e.g. spatially or temporally defined) and dynamic (agent behaviour, or environmental events); a distribution aspect can be observed at population level. Performance (or non-performance) by X can be motivated by by reasons other than the norm, here captured by the condition C_X. This deliberation concerns only the **decision** to initiate performance (D), whereas the outcome of the **action** (A) might in general still be unsuccessful. We can define an *external non-compliance factor* $\mathsf{PNC}^{\mathsf{e}}_X$, including unsuccessful performances.

$$\mathsf{PNC}^{\mathsf{e}}_X = P_X(\text{not}\, A|C) = 1 - [1 - \mathsf{PNC}_X] \cdot P_X(A|D)$$

Monitoring is the starting point for any enforcement. Typically it targets the presence of some **outcome** (O) which is discriminant for the occurrence of the targeted action (A), i.e. for which $P(O|A) - P(O|\text{not}\, A) > 0$. The quality of this discrimination can be captured by the posterior probabilities, namely $P(\text{not}\, A|\text{not}\ O)$ (for punishment, in our running case) or $P(A|O)$ (for reward). Informally, these probabilities measure a relative control of monitoring on the observation points relevant to action outcomes. We can distinguish two steps in **monitoring**: a *selection* mechanism, here captured with $P(M)$, and a *classification* step, whose predictive power for **violation** (V) or **fulfillment** (F) can be measured by $P(\text{not}\, A|\text{not}\ O, M)$ or $P(A|O, M)$. In cases requiring stricter control, instead of advocating full surveillance ($P(M) \sim 1$), a trusted third party (a *certifier*) could certify the action, improving the probability that the action is a proper one. For several reasons, there might impediments to administering **reward** (R) or **punishment** (P); in the general case we should consider a probability $P(R|F)$ or $P(O|V)$.

3. Sketch of economic flow

Monitoring and enforcement have a certain cost for the authority. All these costs will be eventually sustained by the social participants, according to some distribution (e.g. violators might contribute more through penalties). Let us denote with N_P and N_R number of punishments and rewards provided at runtime, with e.g. $N_P = P(P|\text{not}\, A) \cdot \mathsf{PNC}^{\mathsf{e}} \cdot P(C) \cdot N$, N being the number of social participants; and with N_{VV} and N_{FF} respectively the actual (not the observed) numbers of violations and of fulfillments of the norm, which are proportional to the number of applicable cases N_C, following e.g. $N_{VV} = \mathsf{PNC}^{\mathsf{e}} \cdot N_C$; in turn, the number of applicable and monitored cases is proportional to the population

N, with $N_C = P(C) \cdot N$, $N_M = P(M) \cdot N$. Let us assume that the corresponding collective effects are additive and grow linearly; we denote with the factors of growth e_* and f_* the *per-capita* distribution of the aggregated effects of all actions of performance (fulfillment) and non-performance (violation). For simplicity, we will neglect the monitoring cost voice for the authority in reward-based enforcement, but in this case we assume there might be certification costs on the addressee. Table 1 summarizes the distinct economic parameters for the different parties under these assumptions.

The *sustainability* of the system can then be captured by the following constraint:

$$(1 - \mathsf{PNC}^{\mathsf{e}}) \cdot e_* - \mathsf{PNC}^{\mathsf{e}} \cdot f_* \geq m_p \cdot \frac{P(M)}{P(C)} - p \cdot P(P|\text{not}\,A) \cdot \mathsf{PNC}^{\mathsf{e}} + r \cdot P(R|A) \cdot (1 - \mathsf{PNC}^{\mathsf{e}})$$

This formula shows that attempting to bring PNC to 0 is in general not ideal, except perhaps for extremely critical contexts: besides reducing the space of autonomy for the social participants, the higher costs payed collectively might defeat the purpose.

From a theoretical point of view, it can be proven that the overall model (sustainability formula and internal model) confirms both De Geest and Dari-Mattiacci' [1] and Boer's [2] frameworks: (a) if people are generally compliant, too many "carrots" might easily make the system not sustainable; (b) reward-based enforcements become more effective if singling-out or specification problems are present; but also (c) when people are generally non-compliant. If performance is too expensive, avoidance becomes a rational choice, including contesting the authority. If *consensus* is part of the collective value structure, these effects, if quantified, would enter in the formula via f_*, eroding the surplus that was sustaining the punishment-based regime.

4. Conclusion

The example presented in the introduction reflects two norms of an ISP consortium: (i) *If you suffer of a cyber-attack, share information with the consortium*; (ii) *If you are notified of a cyber-attack, start defensive maneuvers.* With adequate values for the environmental parameters, the proposed approach can be used to compute policy parameters for monitoring and enforcement. Clearly, the model presented here is simplistic and several assumptions are unrealistic. However, its extension is easy and straightforward, particularly in integrating e.g. capacities or other non-linear and circular[3] phenomena, non-additive relationships, sounder internal models, and various dynamical aspects, as for instance agents adapting to policies. We plan to extend the model in future study and investigate its application by means of (*optimization* by) *simulation* techniques, as our research targets on aspects of social-technical systems that cannot be treated by game-theoretical approaches based on static pay-off tables.

References

[1] Gerrit De Geest and Giuseppe Dari-Mattiacci. The rise of carrots and the decline of sticks. *University of Chicago Law Review*, 80(1):341–392, 2013.

[2] Alexander Boer. Punishments, rewards, and the production of evidence. In *Legal Knowledge and Information Systems Conference: JURIX 2014*, volume 271, pages 97–102. IOS Press, 2014.

[3] For instance, the probability to get a penalty in case of non-compliance is affected by the resources put in monitoring and enforcement. This probability in turn intervenes in the enforcement costs for the number of punishments and also influences the expectations of the agents, thus modifying the potential of non-compliance.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200878

Precedent Comparison in the Precedent Model Formalism: A Technical Note

Heng ZHENG [a,1], Davide GROSSI [a,b] and Bart VERHEIJ [a]
[a] *University of Groningen, The Netherlands*
[b] *University of Amsterdam, The Netherlands*

Abstract. We outline a formalization of precedent comparison in the precedent model formalism.

Keywords. case-based reasoning, precedents, precedent comparison

1. Introduction

Case-based reasoning allows for a form of analogical reasoning [1], and a core issue is how to make decisions for a current case by comparing precedents. Building on [2, 3, 4], we formalize precedent comparison in case-based reasoning within the precedent model formalism [4]. With the definitions and properties shown below, we show that our approach has the potential to offer a novel angle on case-based reasoning.

2. Precedent comparison in the precedent model formalism

The formalism introduced in this paper uses a propositional logic language L generated from a set of propositional constants. We fix language L. We write $\neg$ for negation, $\wedge$ for conjunction, $\vee$ for disjunction, $\leftrightarrow$ for equivalence. The associated classical, deductive, monotonic consequence relation is denoted $\vDash$.

Precedents consist of factors and outcomes. Both *factors* and *outcomes* are literals. A literal is either a propositional constant or its negation. We use $F \subseteq L$ to represent a set of factors, $O \subseteq L$ to represent a set of outcomes. The sets F and O are disjoint and consist only of literals. If a propositional constant p is in F (or O), then $\neg p$ is also in F (respectively in O). A factor represents an element of a case, namely a factual circumstance. Its negation describes the opposite fact. For instance, if a factor φ is "A is a bad employee", then its negation $\neg\varphi$ is "A is not a bad employee". In our approach, factors are meant to represent generalized case facts relevant to the outcome of the case decision. However, unlike in CATO [1], our use of factors does not assume that factors favor a side of the decision, either pro-plaintiff or pro-defendant, as such an assumption is not needed for our logical definitions of precedent comparison. Unlike HYPO [1], our factors do not

[1] Corresponding Author: Heng Zheng, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands; E-mail: h.zheng@rug.nl.

come with a dimension that can express a magnitude. An outcome always favors a side in the precedent, its negation favors the opposite side. For instance, an outcome ω is "A is dismissed", its negation $\neg\omega$ is "A is not dismissed".

A *precedent* is a logically consistent conjunction of factors and outcomes. A precedent containing an outcome is a *proper precedent*. A precedent without an outcome, is a *situation* representing a current case.

Definition 1 (Precedents) *A* precedent *is a logically consistent conjunction of distinct factors and outcomes $\pi = \varphi_0 \wedge \varphi_1 \wedge \ldots \wedge \varphi_m \wedge \omega_0 \wedge \omega_1 \wedge \ldots \wedge \omega_{n-1}$, where m and n are non-negative integers. We say that $\varphi_0, \varphi_1, ..., \varphi_m$ are the* factors *of π, $\omega_0, \omega_1, ..., \omega_{n-1}$ are the* outcomes *of π. If $n = 0$, then we say that π is a* situation *with no outcomes, otherwise π is a* proper precedent.

Notice that both m and n can be equal to 0. When $m = 0$, there is one single factor. When $n = 0$, the precedent has no outcome and the empty conjunction $\omega_0 \wedge \ldots \wedge \omega_{n-1}$ is equivalent to $\top$. We do not assume that the negation of a factor holds when the factor does not occur in the precedent.

Notions of comparing precedents in case-based reasoning include analogies, distinctions and relevances expressed by general logical formulas, not only factors or outcomes. *Analogies* between two precedents are the formulas that follow logically from both precedents. *Distinctions* are the unshared formulas between two precedents, that only follow logically from one of the precedents and its negation is logically implied by the other precedent. *Relevances* are the unshared formulas between two precedents, that are relevant to the analogies and distinctions between them. These formulas only follow from one of the precedents, but both themselves and their negation are not logically implied by the other one.

Definition 2 (Analogies, distinctions and relevances) *Let $\pi, \pi' \in L$ be two precedents, we define:*

1. *a sentence $\alpha \in L$ is an* analogy between π and π' *if and only if $\pi \models \alpha$ and $\pi' \models \alpha$. A* most specific analogy between π and π' *is an analogy that logically implies all analogies between π and π'.*
2. *a sentence $\delta \in L$ is a* distinction in π with respect to π' (π-π' distinction) *if and only if $\pi \models \delta$ and $\pi' \models \neg\delta$. A* most specific π-π' distinction *is a distinction that logically implies all π-π' distinctions.*
3. *a sentence $\rho \in L$ is a* relevance in π with respect to π' (π-π' relevance) *if and only if $\pi \models \rho$, $\pi' \not\models \rho$ and $\pi' \not\models \neg\rho$. ρ is a* proper π-π' relevance *if and only if ρ is a π-π' relevance that logically implies the most specific analogy between π and π'. A most specific π-π' relevance is a relevance that logically implies all π-π' relevances.*

Both π-π' distinctions and π'-π distinctions are called the *distinctions between π and π'*. Both π-π' relevances and π'-π relevances are called the *relevances between π and π'*. When a most specific analogy/distinction/relevance exists it is by definition unique, and we can refer to it as the most specific analogy/distinction/relevance.

Figure 1 illustrates analogies, distinctions and relevances using Venn diagrams representing sets of worlds in which sentences are true. As shown in Figure 1, for any analogy α between precedents π and π', the sets of π and π' worlds are subsets of the set of α worlds; for any π-π' distinction δ, the π worlds are a subset of the δ worlds, while the π' worlds and the δ worlds are disjoint; for any π-π' relevance ρ, the π worlds are

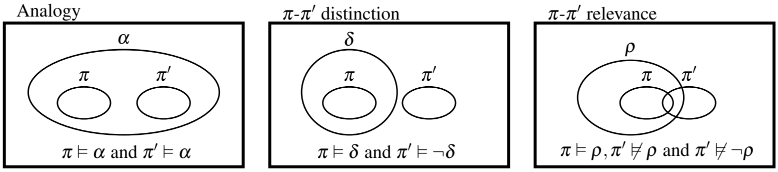

Analogy
α
π
π'
$\pi \vDash \alpha$ and $\pi' \vDash \alpha$
π-π' distinction
δ
π
π'
$\pi \vDash \delta$ and $\pi' \vDash \neg\delta$
π-π' relevance
ρ
π
π'
$\pi \vDash \rho, \pi' \nvDash \rho$ and $\pi' \nvDash \neg\rho$

Figure 1. Precedent comparison illustrated by worlds in which sentences are true

a subset of the ρ worlds, while the π' worlds and the ρ worlds are not subsets of each other and the intersection of the π' worlds and the ρ worlds are always not empty.

The following proposition shows properties of analogies, distinctions and relevances between precedents.

Proposition 1 *Let $\pi, \pi' \in L$ be precedents. Then the following hold:*

1. *The most specific analogy between π and π' always exists and is logically equivalent to $\pi \vee \pi'$;*
2. *There exists a distinction between π and π' if and only if $\pi \wedge \pi' \vDash \bot$; If a π-π' distinction exists, then the most specific π-π' distinction exists and is logically equivalent to π;*
3. *The most specific π-π' relevance does not always exist;*
4. *If the most specific π-π' distinction exists, then the most specific π-π' distinction logically implies each proper π-π' relevance. Each proper π-π' relevance logically implies the most specific analogy between π and π'.*

Proof. Let $\pi, \pi' \in L$ be precedents. For Property 1, by Definition 2, for any analogy α, $\pi \vDash \alpha$ and $\pi' \vDash \alpha$. By propositional logic it follows that any analogy α is logically implied by $\pi \vee \pi'$. By Definition 2, $\pi \vee \pi'$ is a most specific analogy. For Property 2, assume a π-π' distinction δ exists. By Definition 2, $\pi \vDash \delta$ and $\pi' \vDash \neg\delta$. It follows by propositional logic that $\pi \wedge \pi' \vDash \bot$. If $\pi \wedge \pi' \vDash \bot$, by propositional logic $\pi' \vDash \neg\pi$. By Definition 2 and propositional logic, π is therefore the most specific π-π' distinction. For Property 3, assume language L is generated from $\{f_1, f_2\}$. If $\pi = f_1$, $\pi' = \neg f_1$, the most specific π-π' relevance does not exist. π-π' relevances like $f_1 \vee f_2$, $f_1 \vee \neg f_2$ cannot be logically implied by a unique π-π' relevance. For Property 4, by Property 2 if the most specific π-π' distinction exists, it is logically equivalent to π. By Definition 2, π logically implies all π-π' relevances, including proper ones, and proper π-π' relevances always logically imply the most specific analogy between π and π'. □

As shown in Proposition 1, the most specific π-π' distinction is logically equivalent to π if it exists. Note that a precedent itself can be a distinction since precedents are formulas, hence themselves represent the most specific distinction. Property 4 in Proposition 1 shows why we have singled out proper relevances: in the formally precise sense of the proposition, they are logically 'in between' the most specific distinction (if it exists) and the most specific analogy.

Two precedents can be compared with a third precedent using the analogy relation defined below, which is based on the shared formulas between precedents. When comparing precedents π and π' in terms of precedent π'', if the most specific analogy between π and π'' logically implies the most specific analogy between π' and π'', then we say that π is at least as analogous as π' with respect to π''.

Definition 3 (Analogy relation between precedents) *Let π, π' and $\pi'' \in L$ be precedents. We define:*

$$\pi \succeq_{\pi''} \pi' \text{ if and only if } \pi \vee \pi'' \vDash \pi' \vee \pi''.$$

Then we say π is at least as analogous as *π' with respect to π''.*

As customary, the asymmetric part of the relation is denoted as $\pi \succ_{\pi''} \pi'$, which means π is *more analogous* than π' with respect to π''. The symmetric part of the relation is denoted as $\pi \sim_{\pi''} \pi'$, which means π is *as analogous as* π' with respect to π''. If it is not the case that $\pi \succeq_{\pi''} \pi'$ and $\pi' \succeq_{\pi''} \pi$, then we say π and π' are *analogously incomparable with respect to π''.*

Proposition 2 *Let π, π' and $\pi'' \in L$ be precedents. Then the following holds:*

1. *The analogy relation is reflexive and transitive, hence a preorder;*
2. *$\pi \succeq_{\pi''} \pi'$ if and only if $\pi \vDash \pi' \vee \pi''$;*
3. *If $\pi \succeq_{\pi''} \pi'$, then $\pi \succeq_{\pi'} \pi''$ and vice versa;*
4. *For any $\alpha \in L$, if $\pi \succeq_{\pi''} \pi'$, and α is an analogy between π' and π'', then α is also an analogy between π and π''.*

Proof. For property 1, the analogy relation is reflexive, since $\pi \vee \pi'' \vDash \pi \vee \pi''$. The relation is also transitive because of the transitivity of entailment in propositional logic. Assume $\pi = f_1 \wedge f_2$, $\pi' = f_1 \wedge f_3$ and $\pi'' = f_1 \wedge f_2 \wedge f_3$, π and π' are analogously incomparable with respect to π'', hence the relation is not in general total. For Property 2, from left to right, by Definition 3 we obtain $\pi \vee \pi'' \vDash \pi' \vee \pi''$, and by propositional logic $\pi \vDash \pi' \vee \pi''$. From right to left, from $\pi \vDash \pi' \vee \pi''$ and propositional logic, we obtain $\pi \vee \pi'' \vDash \pi \vee \pi''$, and by Definition 3 $\pi \succeq_{\pi''} \pi'$. Property 3 then follows directly from Property 2. Property 4 follows directly from Definition 2 and 3. □

Notice that if $\pi \succeq_{\pi''} \pi'$, then it is still possible that $\pi \nvDash \pi'$ and $\pi \nvDash \pi''$. For instance, if $\pi = f_1$, $\pi' = f_1 \wedge f_2$, $\pi'' = f_1 \wedge \neg f_2$, then we have $\pi \succeq_{\pi''} \pi'$, but both π' and π'' are not logically implied by π. Also notice that if $\pi \succeq_{\pi''} \pi'$, it cannot be concluded that $\pi \vDash \pi'$. For instance, $\pi = f_1 \wedge f_2$, $\pi' = f_3$ and $\pi'' = f_1$. In this example, $\pi \succeq_{\pi''} \pi'$ but $f_1 \wedge f_2 \nvDash f_3$.

3. Conclusion

In this technical note, we showed how to incorporate a form of precedent comparison in the precedent model formalism of [4]. In future work we aim to develop this approach further and use it to represent and reason about actual legal cases.

References

[1] K. D. Ashley. *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, Cambridge, 2017.

[2] B. Verheij. Formalizing Arguments, Rules and Cases. In *Proceedings of the Sixteenth International Conference on Articial Intelligence and Law*, ICAIL 2017, pages 199–208. ACM, New York, 2017.

[3] H. Zheng, M. Xiong, and B. Verheij. Checking the Validity of Rule-Based Arguments Grounded in Cases: A Computational Approach. In M. Palmirani, editor, *Legal Knowledge and Information Systems. JURIX 2018: The Thirty-first Annual Conference*, volume 313, pages 220 – 224. IOS Press, Amsterdam, 2018.

[4] H. Zheng, D. Grossi, and B. Verheij. Case-Based Reasoning with Precedent Models: Preliminary Report. In H. Prakken, S. Bistarelli, F. Santini, and C. Taticchi, editors, *Computational Models of Argument. Proceedings of COMMA 2020*, volume 326, pages 443–450. IOS Press, Amsterdam, 2020.

Demo Papers

This page intentionally left blank

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200880

Arg-tuProlog: A Modular Logic Argumentation Tool for PIL

Roberta CALEGARI [a,1], Giuseppe CONTISSA [a], Giuseppe PISANO [a],
Galileo SARTOR [c] and Giovanni SARTOR [a,b]
[a] *CIRSFID—Alma AI, University of Bologna, Italy*
[b] *European University Institute, Florence, Italy*
[c] *University of Torino, Torino, Italy*

Abstract. Private international law (PIL) addresses overlaps and conflicts between legal systems by distributing cases between the authorities of such systems (jurisdiction) and establishing what rules these authorities have to apply to each case(choice of law). A modular argumentation tool, Arg-tuProlog, is here presented that enables reasoning with rules and interpretations of multiple legal systems.

Keywords. modular argumentation, private international law, Arg-tuProlog

1. Introduction

In our increasingly pervasive and interconnected world, the application and enforcement of the law makes it necessary to take into account the interplay of multiple normative systems, especially when dealing with international contracts and other commercial and social interactions involving different countries. Moreover, normative systems may also interact or conflict on different levels: this is true of both national legal systems and of various transnational or international laws and conventions. All these sources of law need to be considered to properly reason about the law. The research in this paper focuses on the field of private international law (PIL) – a growing and important domain of the law – which deals with the coexistence of multiple normative systems, having distinct and often contradictory rules, and the legal interaction of persons connected to different legal systems, trying to establish priorities between them. Conflicts about competences and rules are addressed by identifying which authority is responsible for making a decision in each given case (jurisdiction), and which set of norms should be applied (applicable law). A logical analysis of PIL has highlighted how this body of law can be suitably modelled by modular argumentation [1] so as to provide a formal model of the interaction among multiple legal systems. For this reason, we are here showing the Arg-tuProlog[2] modular extension to support modular reasoning according to the concept of modularity introduced in [1]—not yet captured and implemented in any known ready-to-use technology.

[1] Roberta Calegari and Giovanni Sartor have been supported by the H2020 ERC Project "CompuLaw" (G.A. 833647).Giuseppe Contissa, Giuseppe Pisano and Galileo Sartor have been supported by the European Union's Justice programme under Grant Agreement No. 800839 for the project "InterLex: Advisory and Training System for Internet-related private International Law".

Arg-tuProlog is a lightweight modular argumentation tool that fruitfully combines modular logic programming and legal reasoning. It makes it possible to represent, reason, and carry out an argument on conditional norms featuring obligations, prohibitions, and (strong or weak) permissions – including under any burden-of-persuasion constraints that may apply – fully supporting the modular argumentation model, i.e., allowing for theory fragmentation, thus enabling the coexistence of different modules. Arg-tuProlog allows the design of knowledge organised in distinct and separate modules that can "call" one another. In particular, a knowledge module can be used by itself, or by referring to another module. This second approach is done by directly calling and querying the relevant module.

2. The domain of private international law: an example

In this section, we will provide an example of interaction between national and transnational normative systems. We will focus on one of the EU's main PIL instruments, the Brussels Regulation: while it provides common EU rules on jurisdiction and the recognition and enforcement of judgments, there are some cases where it points to national legislation for the relevant answer. We have built an example that focuses on the conflict between national laws, namely the Italian and Bulgarian. The legal texts and their representations are extracted from the work done in the context of the Interlex project.[2]

Example 1 (General jurisdiction rule) *In this example we consider article 3.1 of the Italian Law No. 218 of 31 May 1995 (Reform of the Italian System of Private International Law) and article 4 of the Bulgarian Law DB, bp. 42 ot 17.05.2005 r. (Private International Law Code).*

> Article 3.1 (Scope of jurisdiction) Italian courts shall have jurisdiction if the defendant is domiciled or resides in Italy or has a representative in this country who is enabled to appear in court pursuant to Article 77 of the Code of Civil Procedure, as well as in the other cases provided for by law. [...]

Thus Italian courts shall have jurisdiction if the defendant is domiciled or resides in Italy.

> Article 4.1 (General Jurisdiction) The Bulgarian courts and other authorities shall have international jurisdiction where: the defendant has a habitual residence, statutory seat or principal place of business in the Republic of Bulgaria; [...]

Thus Bulgarian courts shall have jurisdiction if the defendant has an habitual residence, statutory seat, or the principal place of business in Bulgaria.

Let us consider, as a first scenario, the case of Marius, an Italian citizen with his primary residence in the city of Rome. Marius is summoned to appear in front of a judge to answer a complaint brought against him. Based on this information we can determine that the Italian court of Rome should be assigned jurisdiction in this complaint.

In a second scenario, Marius is also the owner of a business in Bulgaria. In this case, the Bulgarian PIL law – called by the Brussels Regulation – would assign jurisdiction to a Bulgarian court. Since both rules are valid, the jurisdiction in Marius's case belongs

[2] The European project Interlex is aimed at developing a consultative and training system for internet-related PIL, making it available as an online platform.

to both the Italian and Bulgarian court. If no priority was set, then a conflict of laws would arise, with two equally valid indications of jurisdiction.

The example is now reified in the Arg-tuProlog framework to show the technology's effectiveness and potential. The Brussels Regulation, the Italian national law, and the Bulgarian national law have been mapped onto the Arg-tuProlog framework exploiting three distinct modules: one for the Brussels Regulation, one for the Italian national law, and one for the Bulgarian national law. In the following, we list an extract from the Brussels Regulation codification (*BrusselsRegulation.pl* module) that makes it possible to establish jurisdiction according to the content of the articles.

```
hasJurisdiction(Article, Country, Court, ClaimId):-
 personRole(PID, ClaimId, defendant), memberState(MemberLaw),
 call_module([MemberLaw, ClaimId],
 hasJurisdiction(Article, Country, Court, ClaimId)).
hasJurisdiction(Article, Country, Court, ClaimId):-
 claimObject(ClaimId, rightsInRem), memberState(MemberLaw),
 call_module([MemberLaw,ClaimId],
 hasJurisdiction(Article,Country,Court,ClaimId)).
```

The Italian law module – *italy.pl* – is a simple theory that includes the Prolog translation of the articles from the Italian PIL law as described above. The articles may be represented in the Arg-tuProlog system as follows:

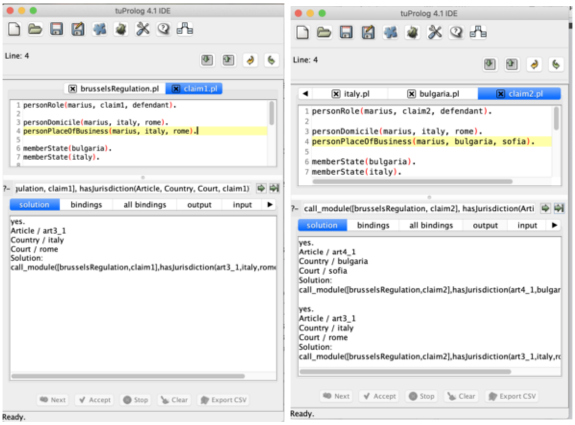
```
hasJurisdiction(art3_1, italy, Court, ClaimId) :-
 personRole(PID,ClaimId,defendant),personDomicile(PID, italy, Court).
hasJurisdiction(art3_1, italy, Court, ClaimId):
 personRole(PID, ClaimId, defendant),
 personAgent(AgentId,PID), personDomicile(AgentId,italy,Court).
hasJurisdiction(art51, italy, Court, ClaimId):-
 claimObject(ClaimId,rightsInRem),immovProperty(ClaimId,italy,Court).
```

The Bulgarian national law is represented by the *bulgaria.pl* module which contains the Prolog translation from the Bulgarian PIL law as described in 2:

```
hasJurisdiction(art4_1, bulgaria, Court, ClaimId):-
 personRole(PID, ClaimId, defendant),
 personDomicile(PID, bulgaria, Court).
hasJurisdiction(art4_1, bulgaria, Court, ClaimId):-
 personRole(PID,ClaimId,defendant),
 personPlaceOfBusiness(PID,bulgaria,Court).
hasJurisdiction(art12,bulgaria,Court,ClaimId):-claimObject(ClaimId,
 rightsInRem),immovProperty(ClaimId,bulgaria,Court).
```

Let us consider the case discussed in Example 1. The facts and details of the case are stored in a separate module (*claim1.pl*), listed in the following.

```
personRole(marius, claim1, defendant).
personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, italy, rome).
memberState(bulgaria).
memberState(italy).
```



```
personRole(marius, claim1, defendant).

personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, italy, rome).

memberState(bulgaria).
memberState(italy).
```

```
?- julation, claim1], hasJurisdiction(Article, Country, Court, claim1)
```

```
yes.
Article / art3_1
Country / italy
Court / rome
Solution:
call_module([brusselsRegulation,claim1],hasJurisdiction(art3_1,italy,rome
```

```
personRole(marius, claim2, defendant).

personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, bulgaria, sofia).

memberState(bulgaria).
memberState(italy).
```

```
?- call_module([brusselsRegulation, claim2], hasJurisdiction(Arti
```

```
yes.
Article / art4_1
Country / bulgaria
Court / sofia
Solution:
call_module([brusselsRegulation,claim2],hasJurisdiction(art4_1,bulgar

yes.
Article / art3_1
Country / italy
Court / rome
Solution:
call_module([brusselsRegulation,claim2],hasJurisdiction(art3_1,italy,rc
```

Figure 1. Arg2p interface: result of claim1 (left) and claim2 (right).

To evaluate the case, we can select the jurisdiction simply by calling the following goal over the top module *brusselsRegulation.pl*:

```
call_module([brusselsRegulation, claim1],
    hasJurisdiction(Article, Country, Court, claim1)).
```

Figure 1 (left) shows the result, which is that under article 3.1 of the Italian law, the court to which the case is assigned is in Rome, Italy. The result is perfectly consistent, since the defendant is domiciled in Italy (and also his place of business). Let us now consider the same case, with the only difference that the place of the defendant's business is in Sofia, Bulgaria (*claim2.pl*).

```
personRole(marius, claim2, defendant).
personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, bulgaria, sofia).
memberState(bulgaria).
memberState(italy).
```

As shown in Figure 1 (right), the answer in this case is twofold. Article 4.1 of the Bulgarian law and Article 3.1 of the Italian law should apply at the same time, assigning jurisdiction to the Sofia (Bulgarian) court in one case and the Rome (Italian) court in the other. The system makes it possible to detect and point out this inconsistency, indicating that two different articles, with different answers in the matter of jurisdiction, should apply simultaneously.

References

[1] P.M. Dung and G. Sartor, The modular logic of private international law, *Artificial Intelligence and Law* **19**(2–3) (2011), 233–261. doi:10.1007/s10506-011-9112-5.

[2] G. Pisano, R. Calegari, A. Omicini and G. Sartor, Arg-tuProlog: a tuProlog-based argumentation framework, in: *CILC 2020 – Italian Conference on Computational Logic. Proceedings of the 35th Italian Conference on Computational Logic*, CEUR Workshop Proceedings, CEUR-WS, Rende, CS, Italy, 2020.

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200881

CAP-A: A Suite of Tools for Data Privacy Evaluation of Mobile Applications

Ioannis CHRYSAKIS [a,b], Giorgos FLOURIS [a], George IOANNIDIS [c],
Maria MAKRIDAKI [d], Theodore PATKOS [a], Yannis ROUSSAKIS [a],
Georgios SAMARITAKIS [a], Alexandru STAN [c], Nikoleta TSAMPANAKI [a],
Elias TZORTZAKAKIS [a], and Elisjana YMERALLI [a]

[a] *FORTH, Institute of Computer Science, Greece*
[b] *IDLab, Dept. of Electronics and Information Systems, UGent, imec, Belgium*
[c] *IN2 Digital Innovations GmbH, Germany*
[d] *FORTH, PRAXI Network, Greece*

Abstract. The utilisation of personal data by mobile apps is often hidden behind vague Privacy Policy documents, which are typically lengthy, difficult to read (containing legal terms and definitions) and frequently changing. This paper discusses a suite of tools developed in the context of the CAP-A project, aiming to harness the collective power of users to improve their privacy awareness and to promote privacy-friendly behaviour by mobile apps. Through crowdsourcing techniques, users can evaluate the privacy friendliness of apps, annotate and understand Privacy Policy documents, and help other users become aware of privacy-related aspects of mobile apps and their implications, whereas developers and policy makers can identify trends and the general stance of the public in privacy-related matters. The tools are available for public use in: `https://cap-a.eu/tools/`.

Keywords. data privacy, privacy evaluation, mobile applications, crowdsourcing

1. Introduction

We experience a massive increase in personal information utilised by smartphone applications, whose invasive nature for harvesting personal data (despite the recently-imposed GDPR legislation) has been demonstrated in many studies. Apps typically analyze their privacy behavior in Privacy Policy (PrP) documents, which describe, in legal terms, the critical privacy-related aspects of the app, such as the types of personal data being accessed, or the way they are being used. However, PrP documents are typically lengthy, difficult to read (containing legal terms and definitions [1]) and frequently changing[1]; a recent study by the Norwegian Consumer Council showed that just reading these documents for apps on a typical smartphone would take several hours[2].

Considering the scope, length and complexity of PrP documents, it comes as no surprise that the average consumer is not investing sufficient time to study such a doc-

[1] https://www.varonis.com/blog/gdpr-privacy-policy/

[2] http://www.forbrukerradet.no/side/the-consumer-council-and-friends-read-app-terms-for-32-hours/

ument before agreeing to it, thus unintentionally granting permission to apps to access and process a wealth of personal information in an unknown manner.

The CAP-A project[3] aims to *support the average user in the daunting task of understanding the content of a PrP document, and to be aware of the privacy implications of using any given mobile app*. This is done through a set of tools employing crowdsourcing techniques to support users in expressing their privacy concerns and expectations, annotating PrP documents, and better understanding privacy-related information regarding the used apps. Developers are also able to contribute to the platform, e.g., by providing justification of the apps' behaviour. The whole approach results in the assessment of mobile apps along two different metrics, which quantify their privacy-related behaviour, as judged by the users' contributions. To enhance participation and provide motivation for active contribution to the platform, we apply a unified rewarding strategy that includes gamification features for active users and developers. Note that CAP-A is not a technical solution, and does not scan or monitor users' devices or apps to assess their behaviour; instead, the project leverages crowdsourcing methods to improve user awareness [3].

2. The CAP-A tools

Due to space restrictions, we only describe the most important functionalities of the CAP-A tools, which are the following:

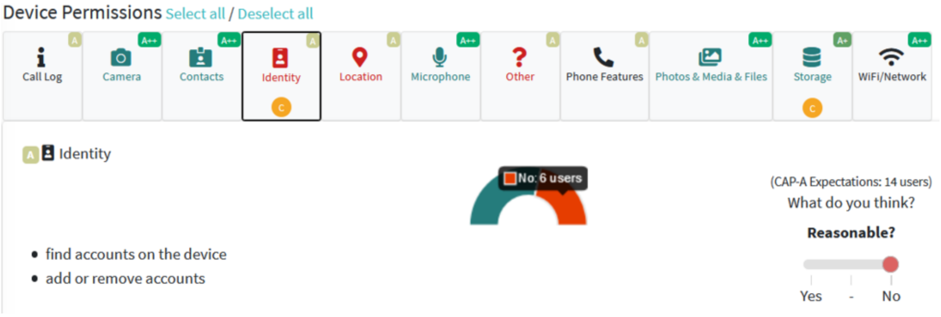
- *Expressing expectations* regarding the expected (or desired) privacy behaviour of each app (Subsection 2.1).
- *Annotating parts of a PrP document* in order to support other users in understanding its content (Subsection 2.2).
- *Accessing app privacy information*, including its privacy evaluation ratings, and viewing interesting statistics through the *Privacy Dashboard* (Subsection 2.3).
- The above functionalities are supported by a rewarding mechanism (Subsection 2.4), which aims at motivating the community to generate the necessary input.

We should note that the CAP-A tools include a mobile app, available through Google Play, which provides a mobile-friendly version of these functionalities. Moreover, developers are also part of CAP-A, and can claim the development of a certain app, giving them special privileges over that app, e.g., being able to justify the access requests of their apps. Details on these functionalities are omitted due to space limitations. The CAP-A tools can be found at: `https://www.cap-a.eu/tools`. Note that the CAP-A tools fully support both the English and the Greek language.

2.1. Expressing expectations

Mobile apps often request access to specific parts of a mobile phone, such as the contacts, the camera etc. The CAP-A tools allow users to express their *expectations* with regards to such requests, i.e., whether they consider reasonable (or not) for a given app to make a given request, showing also the expectations of other users (see Figure 1 as an example).

[3] `https://cap-a.eu/`

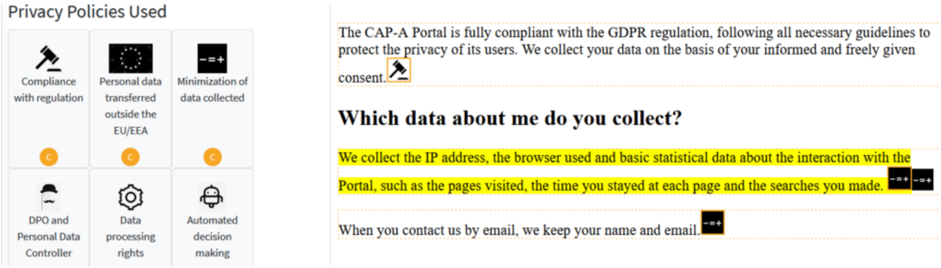


- find accounts on the device
- add or remove accounts

Figure 1. Expressing privacy expectations in CAP-A

2.2. *Annotating PrP documents*

The *PrP Annotator* allows users to mark a block of text in a PrP document and state its relevance to a certain request for access. Annotations are meant to highlight the important blocks in a PrP document, and how they are related to access requests, thereby simplifying the task of understanding its content (see Figure 2). The credibility of this information is assessed based on the (dis)agreement of users' annotations.


Privacy Policies Used
Compliance with regulation
Personal data transferred outside the EU/EEA
Minimization of data collected
C
DPO and Personal Data Controller
Data processing rights
Automated decision making

The CAP-A Portal is fully compliant with the GDPR regulation, following all necessary guidelines to protect the privacy of its users. We collect your data on the basis of your informed and freely given consent.

Which data about me do you collect?

We collect the IP address, the browser used and basic statistical data about the interaction with the Portal, such as the pages visited, the time you stayed at each page and the searches you made.

When you contact us by email, we keep your name and email.

Figure 2. Annotating the PrP document of an app

2.3. *Privacy-related information on apps: Ratings, and the Privacy Dashboard*

Through CAP-A, users can access app-related information; apart from the standard information found also in Google Play, the user may be able to see privacy-related ratings for apps, namely the *"Satisfaction of Community's Expectations"* and the *"Privacy Friendliness"* ratings. The former is calculated based on how close the expectations expressed by the users are to what the application is requesting, whereas the latter takes into account privacy-related best practices, such as frequency of change and understandability of PrP documents, as assessed by users. The related calculations are based on a series of parameters that ensure an intuitive, as well as fair behaviour.

The *Privacy Dashboard* provides interesting visual representations of aggregated statistics regarding users and apps. More importantly, it provides an aggregation of users' input to allow the identification of patterns, such as specific preferences or stances of specific user groups towards certain app categories (e.g., see Figure 3). This can help developers understand how close their services are to what their clients would wish, or help policy makers and simple users identify trends. We constantly consider alternative diagrams to enrich the information given through the Privacy Dashboard.

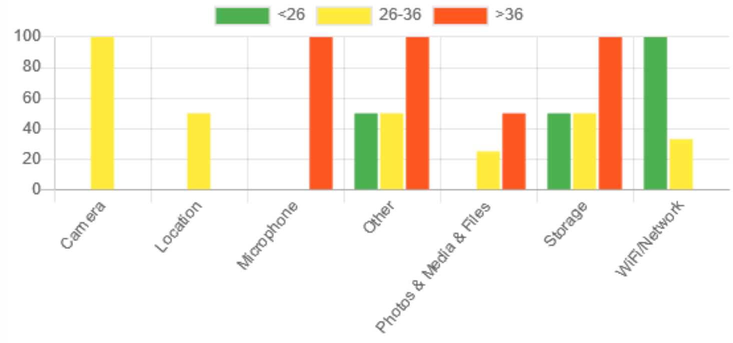


Figure 3. Privacy Dashboard: Game apps and reasonable access to permissions, based on user age

2.4. Rewarding mechanism

Rewarding and gamification mechanisms are indispensable components of most crowd-based solutions. The rewarding mechanism of CAP-A is based on *tiers*, obtained through *points*, provided by *tasks* [2]. Tiers represent the experience level of a user in CAP-A (i.e., amount of interaction with the system). Points are earned through the accomplishment of tasks, which represent useful activities in the system and are organised in levels of sophistication; more complex ones are available to higher-tier users only, to avoid the probable ad-hoc behaviour of first-time users.

3. Conclusion

This paper presented the tools of CAP-A, which aim to improve privacy awareness and users' understanding of the privacy implications associated with the use of any given mobile app, based on crowdsourcing and collective intelligence measures. In our immediate future plans is the evaluation of our platform with real users, in the context of several planned pilots to take place throughout Europe, including a pilot involving legal experts for supporting the project from the legal perspective.

Acknowledgement

This work has been supported by the EU H2020 programme under the NGI_TRUST grant agreement #825618.

References

[1] Anton, A.I., Earp, J.B., Bolchini, D., He, Q., Jensen, C., Stufflebeam, W.: The lack of clarity in financial privacy policies and the need for standardization. In: IEEE Security and Privacy, vol. 2, (2004)

[2] Chrysakis, I., Flouris, G., Patkos, T., Dimou, A. and Verborgh, R. REWARD: Ontology for reward schemes. In 17^{th} Extended Semantic Web Conference: Posters and Demos, pp. 1-5 (2020).

[3] Chrysakis, I., Flouris, G., Ioannidis, G., Makridaki, M., Patkos, T., Roussakis, Y., Samaritakis, G., Stan, A., Tsampanaki., N., Tzortzakakis, E., Ymeralli., E.: Evaluating the data privacy of mobile applications through crowdsourcing. In 32^{rd}JURIX 2020 (to appear).

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200882

Ontology-Based Liability Decision Support in the International Maritime Law

Mirna EL GHOSH [a,1], Habib ABDULRAB [a]
[a] *Normandie Université, INSA Rouen, 76000, Rouen, France*

Abstract. In this paper, we present an ontology-based liability decision support task in the international maritime law, specifically the domain of carriage of goods by sea. We analyze the liabilities of the involved legal agents (carriers and shippers) in case of loss or damage of goods. Thus, a well-founded legal domain ontology, named CargO-S, is used. CargO-S has been developed using an ontology-driven conceptual modeling process, supported by reusing foundational and legal core ontologies. In this work, we demonstrate the usability of CargO-S to design and implement a set of chained rules describing the procedural aspect of the liabilities legal rules. Finally, we employ these rules in a liability rule-based decision support task using a real case study.

Keywords. well-founded legal domain ontologies, legal decision support, ontology-based decision support, maritime law

1. Introduction

Legal decision support aims to help solving problems in the juridical domain. The most commonly known approaches focus on employing rules to describe judges' strategies and procedures to analyze legal issues [1]. However, the particular characteristics of the legal domain cause specific difficulties in establishing such tasks [1]. In this study, we propose a decision support task that helps making decisions in the legal domain using ontological models. The implementation of such tasks implies creating an appropriate legal domain ontology that reflects as much as possible the real application domain [2]. Ontology models that are faithful to realities are called *well-founded* domain ontologies [3]. These ontologies are *grounded* in validated foundational ontologies where concepts and relations are previously analyzed in the light of a foundational ontology. The domain application of this work is international maritime law. Specifically, the domain of carriage of goods by sea represented by the Hague-Visby Rules[2] is designated. The main goal is to define the liabilities of the involved legal agents (carriers and shippers) in case of loss or damage of goods. Thus, a well-founded legal domain ontology, named CargO-S, is used. In this work, we do not expose the building process of CargO-S. However, we demonstrate the ontology's usability to design a set of chained rules describing the procedural aspect of liabilities legal rules. Furthermore, we employ these rules in a rule-

[1] Corresponding Author:

[2] Hague–Visby Rules is a set of international rules for the international carriage of goods by sea, *Wikipedia, last visited September, 04 2020*

based decision support task. The remainder of this paper is organized as follows: Section 2 introduces CargO-S. Section 3 discusses the modeling and formalizing of legal rules. In section 4, we employ the formalized rules in a decision support task. Finally, section 5 concludes the paper.

2. CargO-S: A Well-Founded Legal Domain Ontology for the Domain of Carriage of Goods by Sea

In this section, we briefly introduce CargO-S, a pattern-based well-founded legal domain ontology for the domain of carriage of goods by sea. CargO-S has been developed using an ontology-driven conceptual modeling process [4] and grounded in the unified foundational ontology UFO by applying the ontology-driven conceptual modeling language OntoUML [5]. Besides, the development of CargO-S has been supported by reusing conceptual ontology patterns from UFO[3] and the legal core ontology UFO-L [6]. The structure of CargO-S is composed of three different layers located at different granularity levels: *upper*, *core* and *domain*. The upper and core layers are composed of different types of ontology patterns, which are applied either by extension or analogy for building the domain layer. In this study, we focus on the domain layer where different legal categories are defined: (1) legal roles such as, Carrier and Shipper; (2) legal relators such as, Contract_of_Carriage_of_Goods_by_Sea, Right_Duty_to_Indemnity, and Disability_Immunity_to_Liabilities; (3) legal moments such as, Right_to_Indemnity, and Disability_to_Indemnity; (4) legal events, such as Carriage_of_Goods_by_Sea, Loss_or_Damage_of_Goods; (5) situations, such as Inaccuracies and Unseaworthiness.

3. Modeling and Formalizing the Procedural Aspect of Liabilities Legal Rules using CargO-S

In this section, we describe the modeling and formalizing of legal liability rules using CargO-S. *Rule 1* is an example of a simple legal rule where a situation of inaccuracies triggers the loss or damage of goods.

Rule 1. (Article 3. Section 5) The shipper shall indemnify the carrier against all loss, damages and expenses arising or resulting from inaccuracies.

For the modeling process, a core ontology pattern (Right_duty_to_an_Action) is applied by analogy with the legal rule for representing its procedural aspect. Following the isomorphism principle stated by Bench-Capon [7] the conceptual model is transformed into formal rule. In this paper, we use SWRL[3] as a formal rule language. By representing legal rules using SWRL, we assume that they are conflict-free and will be complied with. In Figure 1, we depict the ontological model of *Rule 1* represented in OntoUML [5].

[3] https://www.w3.org/Submission/SWRL/

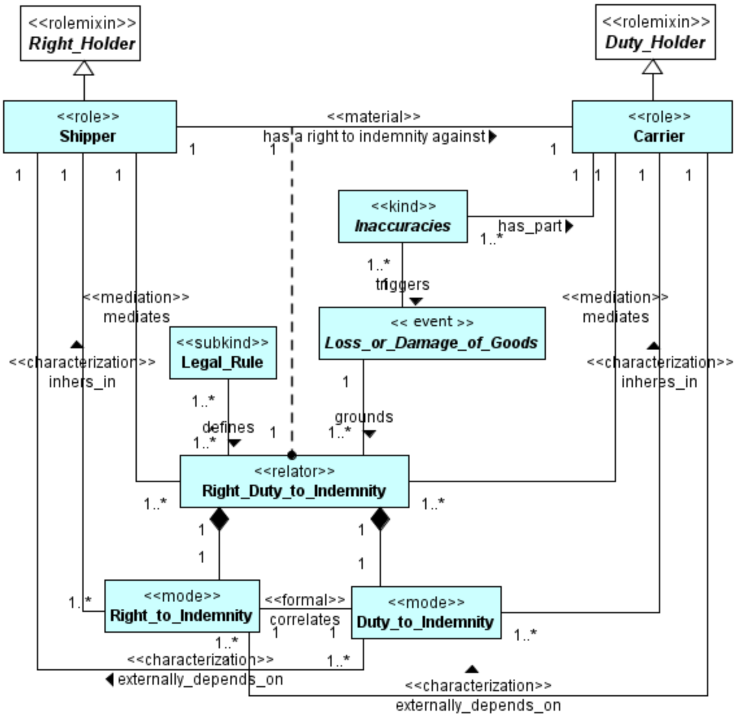


Figure 1. Modeling the procedural aspect of Rule 1 represented in OntoUML.

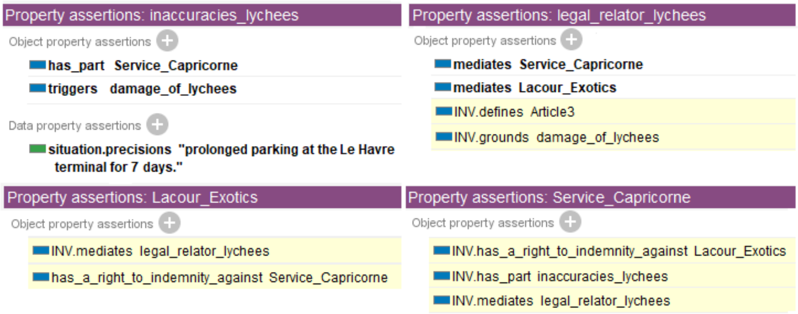
Based on the ontological model, two chained formal rules are generated where the latter's condition is the former's consequence. These rules are represented by an obligation rule in the following form: *IF condition (operative facts) THEN conclusion (legal effect).*

```
Carrier(?c) ∧ Shipper(?s) ∧ Legal_Relator(?r) ∧ mediates(?r, ?c) ∧ mediates(?r, ?s) ∧
Loss_or_Damage_of_Goods(?d) ∧ grounds(?d,?r) ∧ Inaccuracies(?i) ∧ has_part(?i, ?c) ∧
triggers(?i, ?d) ⟹ Right_Duty_to_Indemnity(?r)∧ mediates(?r, ?s) ∧
mediates(?r, ?c) ∧ defines(Article3, ?r)
```

```
Right_Duty_to_Indemnity(?r)∧ Carrier(?c)∧ Shipper(?s)∧ mediates(?r, ?c)∧
mediates(?r, ?s) ⟹has_a_right_to_indemnity_against(?s, ?c)
```

4. Rule-Based Liability Decision Support

This section outlines the formal rules' employment in a lightweight decision support task. Given an event of carriage of lychees operated by the carrier Service Capricorne and the shipper Lacour Exotics. At the destination port, a prolonged parking for seven days has occurred. Such a situation of inaccuracies have triggered the damage of fruits. Thereby, based on the given facts, different legal effects are computed and generated following the execution of rules (Figure 2). As a legal decision, we obtained that the shipper has a right to indemnity against the carrier.



Property assertions: inaccuracies_lychees

Object property assertions

- has_part Service_Capricorne
- triggers damage_of_lychees

Data property assertions

- situation.precisions "prolonged parking at the Le Havre terminal for 7 days."

Property assertions: legal_relator_lychees

Object property assertions

- mediates Service_Capricorne
- mediates Lacour_Exotics
- INV.defines Article3
- INV.grounds damage_of_lychees

Property assertions: Lacour_Exotics

Object property assertions

- INV.mediates legal_relator_lychees
- has_a_right_to_indemnity_against Service_Capricorne

Property assertions: Service_Capricorne

Object property assertions

- INV.has_a_right_to_indemnity_against Lacour_Exotics
- INV.has_part inaccuracies_lychees
- INV.mediates legal_relator_lychees

Figure 2. The legal effects as inferences following the execution of SWRL rules in Protégé[4]

5. Conclusion

In this paper, we presented an ontology-based liability decision support task in international maritime law. We demonstrated that decision support could solve basic legal problems using well-founded legal domain ontologies. A well-founded legal domain ontology, named CargO-S, has been used for this purpose. In CargO-S, various legal categories have been defined by reusing conceptual ontology patterns from foundational and core ontologies. Core patterns have been applied by analogy to describe the procedural aspect of legal rules. Furthermore, the legal rules' ontological model has been transformed into chained formal rules embedded in a lightweight decision support task. This work was a preliminary study. In further works, we will examine more complex rules taking into consideration different formal rule languages. *Acknowledgment*: This work has been supported by the European Regional Development Fund (ERDF) under Grant Agreement n HN0002134 in the project CLASSE2.

References

[1] Zeleznikow J (2004) Building Intelligent Legal Decision Support Systems: Past Practice and Future Challenges. In: Fulcher J, Jain L C (eds) Applied Intelligent Systems. Studies in Fuzziness and Soft Computing, vol 153. Springer, Berlin, Heidelberg

[2] Daduna J, Hunke K and Prause G (2012) Logistics Corridors and Short Sea Shipping in the Baltic Sea Area. In Proceedings of the International Research Conference on Short Sea Shipping, Estoril, Portugal

[3] Guizzardi G (2005) PhD Thesis. Ontological Foundations for Structural Conceptual Models. Telematica-Institut / CTIT

[4] El Ghosh M, Abdulrab H (2019) The application of ODCM for Building Well-founded Legal Domain Ontologies: A Case Study in the Domain of Carriage of Goods by Sea Conventions. In: International Conference on Artificial Intelligence and Law, ICAIL, Montreal

[5] Guerson J, Sales T P, Guizzardi G and Almeida J (2015) OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies. In: 19th IEEE Enterprise Computing Conference.

[6] Griffo C (2018) UFO-L, A Core Ontology of Legal Aspects Building Under the Perspective of Legal Relations, PhD Thesis, Federal University of Espirito Santo

[7] Bench-Capon T and Coenen F (1992) Isomorphism and Legal Knowledge Based Systems. In: Artificial Intelligence and Law 1.1, pp. 65–86

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200883

JURI SAYS: An Automatic Judgement Prediction System for the European Court of Human Rights

Masha MEDVEDEVA [a,b,1], Xiao XU [a], Martijn WIELING [a] and Michel VOLS [b]
[a] *Center for Language and Cognition, University of Groningen, The Netherlands*
[b] *Department of Legal Methods, University of Groningen, The Netherlands*

Abstract.
In this paper we present the web platform JURI SAYS that automatically predicts decisions of the European Court of Human Rights based on communicated cases, which are published by the court early in the proceedings and are often available many years before the final decision is made. Our system therefore predicts *future* judgements of the court. The platform is available at jurisays.com and shows the predictions compared to the actual decisions of the court. It is automatically updated every month by including the prediction for the new cases. Additionally, the system highlights the sentences and paragraphs that are most important for the prediction (i.e. violation vs. no violation of human rights).

Keywords. European Court of Human Rights, machine learning, web platform

1. Introduction

In recent years, the use of machine learning for predicting judicial decisions has become more popular [1, 2, 3, 4, 5, 6, 7] as these methods are able to detect patterns in increasingly large legal datasets. In this paper we introduce an online platform, JURI SAYS, which automatically retrieves legal documents from the European Court of Human Rights (ECtHR) database, and subsequently predicts the judgements of the cases on the basis of information which was available *before* the judgement was made. In addition to predicting decisions, JURI SAYS identifies and highlights sentences that were most important for its prediction.

The JURI SAYS system can roughly be divided into three parts: 1) a database, 2) a machine learning system, and 3) a web platform. Each part is independent from the others and offers a set of Application Programming Interfaces (APIs) to add flexibility for the future, allowing (for example) more documents to be added, new machine learning models to be included, or adjusting the interface. Before discussing the architecture of the system, however, some background on the legal data underlying our system is necessary.

[1] Corresponding Author: Masha Medvedeva, Center for Language and Cognition Groningen, Oude Kijk in 't Jatstraat 26, 9712 EK Groningen, the Netherlands; Email: m.medvedeva@rug.nl

2. Data

The European Court of Human Rights is an international court established in 1959 that deals with individual and State applications claiming a violation of various rights laid out in the European Convention on Human Rights (ECHR) [8, 9, 10]. Applications are always brought by an individual against a State or multiple States that have ratified the ECHR, with rare exception of State against State cases.

To our knowledge, all previous research on predicting decisions of the ECtHR used various parts of the final decision published by the court. In this work we refrain from using these documents for prediction, because when they are compiled the final decision is already known and the text (even parts which do not contain the judgement itself) may reflect that decision [7]. Fortunately, however, the court publishes multiple documents at various stages of the proceedings.

Once the application is made by the alleged victim, and fits the formal admissibility criteria, the court often *communicates* the facts of the case to the State against which there is a complaint. It also poses some questions to the State, so that the State may corroborate or deny the allegations. These documents are labeled as *Communicated cases* on the HUDOC website.[2] JURI SAYS predicts the decisions on the bases of these documents. Once the case is communicated, it goes through an admissibility stage, where it is evaluated based on merit. The clear-cut cases with no violation are then found inadmissible, the rest move to the next stage, where the judgement is made, and the final document with facts, arguments and the decision is produced.

3. JURI SAYS

3.1. Database

Our database only includes documents in English. Every month new documents are downloaded and a new machine learning model to predict the ECtHR decisions of that month is trained (see below). At present our database contains 4929 communicated cases with their associated decision. While the predictions are only based on the communicated cases, we also include information from cases from the last ten years that were not communicated to increase the amount of data available to train our model. For those cases, we only extract the “Facts” part from the final document with the judgement [7].

Our system automatically extracts the raw text of the communicated cases from the database of the ECtHR, in addition to some metadata, such as the decisions (for admissibility cases and judgements), data, parties, articles involved, et cetera. The decisions are then associated to the communicated cases on the basis of the application number.

3.2. Machine learning system

Every month, after downloading the new documents, the system behind our web platform JURI SAYS carries out three tasks. It first trains a new machine learning model (introduced in Medvedeva et al. [7]) on the basis of all data *excluding* the data from the most recent month. Then it predicts the judicial decision for the cases of the most recent month on

[2] https://hudoc.echr.coe.int

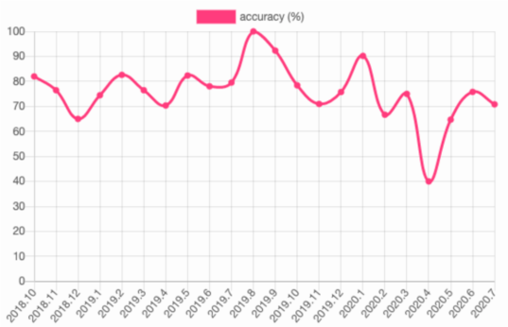
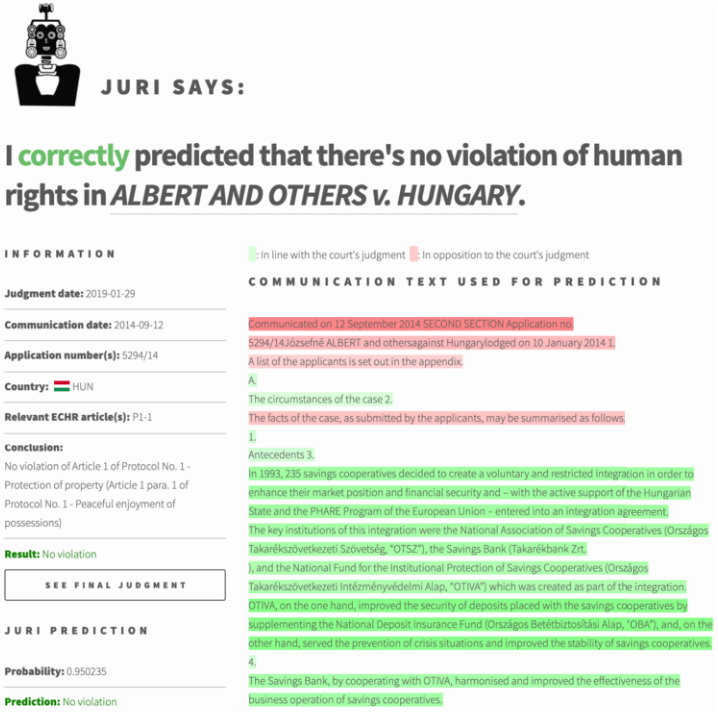


Figure 1. Accuracy of JURI SAYS over the past two years predicting future judgements.


JURI SAYS:

I correctly predicted that there's no violation of human rights in *ALBERT AND OTHERS v. HUNGARY*.

INFORMATION

Judgment date: 2019-01-29

Communication date: 2014-09-12

Application number(s): 5294/14

Country: HUN

Relevant ECHR article(s): P1-1

Conclusion:
No violation of Article 1 of Protocol No. 1 - Protection of property (Article 1 para. 1 of Protocol No. 1 - Peaceful enjoyment of possessions)

Result: No violation

SEE FINAL JUDGMENT

JURI PREDICTION

Probability: 0.950235

Prediction: No violation

🟩: In line with the court's judgment 🟥: In opposition to the court's judgment

COMMUNICATION TEXT USED FOR PREDICTION

Communicated on 12 September 2014 SECOND SECTION Application no.
5294/14Józsefné ALBERT and othersagainst Hungarylodged on 10 January 2014 1.
A list of the applicants is set out in the appendix.
A.
The circumstances of the case 2.
The facts of the case, as submitted by the applicants, may be summarised as follows.
1.
Antecedents 3.
In 1993, 235 savings cooperatives decided to create a voluntary and restricted integration in order to enhance their market position and financial security and – with the active support of the Hungarian State and the PHARE Program of the European Union – entered into an integration agreement.
The key institutions of this integration were the National Association of Savings Cooperatives (Országos Takarékszövetkezeti Szövetség, "OTSZ"), the Savings Bank (Takarékbank Zrt.
), and the National Fund for the Institutional Protection of Savings Cooperatives (Országos Takarékszövetkezeti Intézményvédelmi Alap, "OTIVA") which was created as part of the integration.
OTIVA, on the one hand, improved the security of deposits placed with the savings cooperatives by supplementing the National Deposit Insurance Fund (Országos Betétbiztosítási Alap, "OBA"), and, on the other hand, served the prevention of crisis situations and improved the stability of savings cooperatives.
4.
The Savings Bank, by cooperating with OTIVA, harmonised and improved the effectiveness of the business operation of savings cooperatives.

Figure 2. An example of a correctly predicted case by JURI SAYS with highlighted sentences.

the basis of the newly-created model. The performance (accuracy) of JURI SAYS for each month during the last two years can be found in Figure 1. Finally, for each sentence in the text of the communicated case, it identifies how strongly it is related to the actual judgement of the court (by estimating the probability of the sentence belonging to a case with a violation versus to a case without a violation of human rights; see also Figure 2).

3.3. Web platform

JURI SAYS is the web platform of our system presenting the results of applying our machine learning system to the extracted data of the ECtHR. JURI SAYS is updated every month by publishing the predictions for the most recent ECtHR cases. It also offers a list of all historical cases that may be ordered or filtered by date or article involved. For every single case there is a separate page that offers more detailed information, including the predicted outcome of the case, together with an associated probability of that predicted outcome, and the actual judgement of the court. For each sentence in the text of the communicated case, the predicted label and associated probability is shown when the mouse pointer is hovered over a sentence. Sentences which are highlighted in green are consistent with the court's actual decision, those in red are more likely to be associated with the opposite decision. See Figure 2 for an example. The intensity of the colour reflects how strongly associated the sentences are with the respective decisions.

4. Conclusion

In this paper, we have presented JURI SAYS, an automatic judgement prediction system for the ECtHR. Our system uses automatically extracted textual information from documents available long before the court decision was made. In addition, our model predicts cases for the following month (i.e. the future), which is a hard task [7]. Therefore, it is nice to see the relatively high performance of our system with an accuracy of 75%. By automatically highlighting critical sentences, and automatically updating every month, our system aims to offer a user-friendly web platform for legal professionals.

References

[1] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.

[2] Sulea OM, Zampieri M, Vela M, Van Genabith J. Predicting the law area and decisions of French Supreme Court cases. arXiv preprint arXiv:170801681. 2017;.

[3] Katz DM, Bommarito MJ, Blackman J. A general approach for predicting the behavior of the Supreme Court of the United States. PloS one. 2017;12(4):e0174698.

[4] Chen DL, Eagel J. Can machine learning help predict the outcome of asylum adjudications? In: Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law; 2017. p. 237–240.

[5] Lage-Freitas A, Allende-Cid H, Santana O, de Oliveira-Lage L. Predicting Brazilian court decisions. arXiv preprint arXiv:190510348. 2019;.

[6] O'Sullivan C, Beel J. Predicting the outcome of judicial decisions made by the European Court of Human Rights. arXiv preprint arXiv:191210819. 2019;.

[7] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. 2020;28(2):237–266.

[8] Greer S, Gerards J, Slowe R. Human rights in the Council of Europe and the European Union: Achievements, trends and challenges. Cambridge Studies in European Law and Policy. Cambridge University Press; 2018.

[9] Harris DJ, O'Boyle M, Bates E, Buckley C. Harris, O'Boyle & Warbrick: Law of the European Convention on Human Rights. Oxford University Press, USA; 2014.

[10] Gerards J. General principles of the European Convention on Human Rights. Cambridge University Press; 2019.

S. Villata et al. (Eds.)

doi:10.3233/FAIA200884

Reasoning About Applicable Law in Private International Law in Logic Programming[1]

Ken SATOH [a,2] and Matteo BALDONI [b] Laura GIORDANO [c]
[a] *National Institute of Informatics, Japan*
[b] *Università di Torino, Italy*
[c] *Università del Piemonte Orientale, Italy*

Abstract. We formalized renvoi in private international law in JURIX 2019 in terms of modal logic fragment. In this demonstration paper, we show an implementation of the formalism by translating modal formula into a logic program.

1. Introduction

Private international law (PIL)"enables the coexistence of multiple normative systems, having distinct and often contradictory rules [3]". In international legal relations, we have to decide "applicable law" for an international legal issue since the issue is related with multiple countries which might have contradictory rules each other for the issue. We[2] formalize PIL to decide "applicable law" for a legal issue with "renvoi": to determe an applicable law for an international legal matter in one country may require to refer to another law in another country which may result in a sequence of references of laws to different countries. We formalize this reasoning by a rule-based fragment of the modal language in [1], extended with context variables, and allows the interactions among contexts to be captured, context variables to occur within modalities and context names to be used as predicate arguments, thus supporting a simple combination of meta-predicates and modal constructs. In this paper, we translate the formalism into a logic program which reifies predicate to express legal matters with a variable to express the country of applicable law for the international legal matters. As a related research, the specificity of the rules in Conflict of Laws have been analyzed by Markovich [5] in the formalism of the input/output framework [4], where such rules assign a set of sets of norms (a legal system) to a given domain (a set of statements).

2. Reasoning about Applicable Law

We review how to reason about applicable law defined in [2]. Given a legal matter P in one country, C, we would like to decide whether the matter is valid in the country in the following way.

[1] This work was supported by JSPS Topic-Setting Program to Advance Cutting-Edge Humanities and Social Sciences Research Grant Number JPJS00119217216.

[2] Corresponding Author: Ken Satoh, National Institute of Informatics, 2-1-2, Chiyoda-ku, Tokyo, Japan; E-mail:ksatoh@nii.ac.jp

1. We decide the country X whose law is applied to decide the matter P as follows.
   (a) There should be a rule in the private international law in C which indicates an applicable law in (possible another) country C' for the matter P in the country, C.
   (b) If $C' = C, X = C$.
   (c) Else ($C' \neq C$), we need to again decide the country X of the applicable law for P according to the private international law in C' (called "renvoi")
   (d) If we detect a loop in the "renvoi", we set the applicable law to the starting country of the loop. For example, if the private international laws makes this reference of applicable law, "$A \to B \to C \to D \to B$", then we decide an applicable law for the matter as country B.
2. We decompose the matter P into submatters according to a rule defined in the applicable law in X.
3. If a submatter is determined by a global fact and the global fact is in the fact base, the submatter is valid.
4. Otherwise, we iterate the process above (we decide an applicable law of the submatter and then check the submatter is valid in the applicable law).

3. Translation of Modal Program into a Logic Program

Top rule is translated into:

```
holds(P#CountryA) :- applicable_law(P,CountryB,[])#CountryA, P#CountryB.
```

which means that

1. we reason about renvoi, that is, decide a country for an applicable law, `CountryB` whose law determines a matter, `P` in CountryA, then
2. we determine `P` in the law of `CountryB`.

Here is a rule to compute applicable law to handle a loop case in renvoi:

```
applicable_law(P,CountryA,ReferredHistory)#CountryA :-
    member(CountryA,ReferredHistory).
```

means that if the current country is in the sequence, `ReferredHistory` of referred country so far, then the country of applicable law is `CountryA`.
Here is a rule to computing applicable law for specific predicate:

```
applicable_law(P,CountryC,ReferredHistory)#CountryA :-
    \+member(CountryA,ReferredHistory),
    some_coditions_to_refer_to_other_country(CountryB),
    applicable_law(P,CountryC,[CountryA|ReferredHistory])#CountryB.
```

For example, to decide a country for an appliable law for `heir(Child,Parent)`, a condition to refer to other country is `home_country(Parent,CountryB)` so a rule of computing an applicable law is:

```
applicable_law(heir(Child,Parent),CountryC,ReferredHistory)#CountryA :-
    \+member(CountryA,ReferredHistory),
    home_country(Parent,CountryB),
    applicable_law(
        heir(Child,Parent),CountryC,[CountryA|ReferredHistory])#CountryB.
```

```
% Top rule
holds(P#CountryA) :-
     applicable_law(P,CountryB,[])#CountryA,
     P#CountryB.
% Renvoi
applicable_law(P,CountryA,ReferredHistory)#CountryA :-
     member(CountryA,ReferredHistory).
applicable_law(heir(Child,Parent),CountryC,ReferredHistory)#CountryA :-
     \+member(CountryA,ReferredHistory),
     home_country(Parent,CountryB),
     applicable_law(heir(Child,Parent),CountryC,[CountryA|ReferredHistory])#CountryB.
applicable_law(legitimate_child_parent_rel(Child,Parent),CountryC,ReferredHistory)#CountryA :-
     \+member(CountryA,ReferredHistory),
     home_country(Parent,CountryB),
     applicable_law(legitimate_child_parent_rel(Child,Parent),CountryC,[CountryA|ReferredHistory])#CountryB.
applicable_law(marriage(Spouse1,_),CountryC,ReferredHistory)#CountryA :-
     \+member(CountryA,ReferredHistory),
     home_country(Spouse1,CountryB),
     applicable_law(marriage(Spouse1,_),CountryC,[CountryA|ReferredHistory])#CountryB.
% Global Rules/ Facts
home_country(Person,Country) :-
     nationality(Person,Country), habitual_residence(Person,Country).
nationality(john,country1). habitual_residence(john,country1).
bilogical_child_parent_rel(taro,john). agreement(marriage,john,yoko).
registering(marriage,john,yoko,country1).
% Domestic Laws
heir(Child,Parent)#country1 :-
      holds(legitimate_child_parent_rel(Child,Parent)#country1).
legitimate_child_parent_rel(Child,Parent)#country1 :-
      holds(marriage(Parent,Spouse)#country1),
      bilogical_child_parent_rel(Child,Parent).
marriage(Spouse1,Spouse2)#Country :-
      agreement(marriage,Spouse1,Spouse2),
      registering(marriage,Spouse1,Spouse2,Country).
```

Figure 1. PROLOG program for Private International Law

Global rules and facts represented in the modal settings, "□(H :- B1,...,Bn)." or "□H." is translated into ordinary horn clauses: "H :- B1,...,Bn." or "H."
Domestic rules specific to a country, Country, "□[Country] {H :- B1,...,Bn.}" is translated into H#Country :- B1',...,Bn'. where Bi' is Bi#Country if Bi is the head of a domestic rule, otherwise (Bi is the head of a global rule), Bi.

Fig. 1 shows the entire program to compute heir(taro,john)#japan. In the program, holds(P#Country) and P#Country are similar predicates but P#Country means that a predicate, P, is true according to the law in a country, Country whereas holds(P#Country) means that a predicate P is true according to a law in a country, Country' after we find that the an applicable law of the Country' for P starting from Country.

4. Demonstration

We will demonstrate a process to reason about applicable law for the example in [2]. Fig. 2 is an output of how to reason about heir(taro,john) in Japan. In this example, firstly we decide a coutry whose low is applied (we call *applied country* here) to this issue. We assume that a rule for this decision is universal in that the applied country is the home country of a parent derived from the facts of parent's nationality and habitual residence. In this example, it is country1. Then, we check heir(taro,john) is true in country1. Then according to the law of country1, we should check legitimate_child_parent_rel(Child,Parent) in country1.
Then, again we decide an applied country for legitimate child-parent relation and we find that country1 should be the applied country. Then, legitimate child-parent relation is true if the parent is married and biological relation between the parent and the child exits.

```
  Starting to prove holds(heir(taro,john)#japan)
  holds(heir(taro,john)#japan):-
      applicable_law(heir(taro,john),_9392,[])#japan, heir(taro,john)#_9392.
    is matched
    applicable_law(heir(taro,john),_9392,[])#japan:-
        \+member(japan,[]), home_country(john,_9460),
         applicable_law(heir(taro,john),_9392,[japan])#_9460.
      is matched
      Succeeded in Proving \+member(japan,[])
      Starting to prove home_country(john,_9460)
      .................... (omitted)
      Succeeded in Proving home_country(john,country1)
      Starting to prove applicable_law(heir(taro,john),_9392,[japan])#country1
      applicable_law(heir(taro,john),_9392,[japan])#country1:-
          \+member(country1,[japan]), home_country(john,_9654),
          applicable_law(heir(taro,john),_9392,[country1,japan])#_9654.
        is matched
        Succeeded in Proving \+member(country1,[japan])
        Starting to prove home_country(john,country1)
        .................... (omitted)
        Succeeded in Proving home_country(john,country1)
        Starting to prove applicable_law(heir(taro,john),_9392,[country1,japan])#country1
        applicable_law(heir(taro,john),country1,[country1,japan])#country1:-
            member(country1,[country1,japan]).
          is matched
          Succeeded in Proving member(country1,[country1,japan])
        Succeeded in Proving applicable_law(heir(taro,john),country1,[country1,japan])#country1
      Succeeded in Proving applicable_law(heir(taro,john),country1,[japan])#country1
    Succeeded in Proving applicable_law(heir(taro,john),country1,[])#japan
/* We have shown that an applicable law to prove heir(taro,john) in Japan is country1's law */
    Starting to prove heir(taro,john)#country1
    heir(taro,john)#country1:- holds(legitimate_child_parent_rel(taro,john)#country1).
      is matched
      Starting to prove holds(legitimate_child_parent_rel(taro,john)#country1)
      holds(legitimate_child_parent_rel(taro,john)#country1):-
          applicable_law(legitimate_child_parent_rel(taro,john),_9900,[])#country1,
          legitimate_child_parent_rel(taro,john)#_9900.
        is matched
        Starting to prove applicable_law(legitimate_child_parent_rel(taro,john),country1,[])#country1
        .................... (omitted)
        Succeeded in Proving applicable_law(legitimate_child_parent_rel(taro,john),country1,[])#country1
/* We have shown that an applicable law to prove legitimate_child_parent_rel(taro,john)
   in country1 is country1's law */
        Starting to prove legitimate_child_parent_rel(taro,john)#country1
        legitimate_child_parent_rel(taro,john)#country1:-
          holds(marriage(john,_10182)#country1), bilogical_child_parent_rel(taro,john).
          is matched
          Starting to prove holds(marriage(john,yoko)#_10182)
            Starting to prove applicable_law(marriage(john,_10182),country1,[])#country1
            .................... (omitted)
            Succeeded in Proving applicable_law(marriage(john,_10182),country1,[])#country1
/* We have shown that an applicable law to prove marriage(john,_10182) in country1 is country1's law */
          .................... (omitted)
          Succeeded in Proving holds(marriage(john,yoko)#country1)
          Starting to prove bilogical_child_parent_rel(taro,john)
          Succeeded in Proving bilogical_child_parent_rel(taro,john)
        Succeeded in Proving legitimate_child_parent_rel(taro,john)#country1
      Succeeded in Proving holds(legitimate_child_parent_rel(taro,john)#country1)
    Succeeded in Proving heir(taro,john)#country1
  Succeeded in Proving holds(heir(taro,john)#japan)
```

Figure 2. PROLOG Execution to check `heir(taro,john)` in `japan`

To show the marriage status, we need to find an applied country for the marriage status and we find that the applied country is `country1`. According to a universal rule, we can show the marriage status by showing agreement between a pair and registration.

5. Conclusion

We show a translation method of modal formalization of reasoning about applicable law into logic programming.

References

[1] Baldoni M, Giordano L, Martelli A. A modal extension of logic programming: modularity, beliefs and hypothetical reasoning. J. Log. Comput., Vol. 8, No. 5. 1998. p 597-635.

[2] Baldoni M, Giordano L, Satoh, K. Renvoi in private international law: a formalization with modal context. Proc. of JURIX 2019. 2019. p157-162.

[3] Dung PM, Sartor G. The modular logic of private international law. Artif. Intell. Law, Vol. 19, No. 2-3. 2011. p233-261.

[4] Makinson D, van der Torre L. What is input/output logic? Foundations of the Formal Sciences II. 2003. p163-174.

[5] Markovich R. On the formal structure of rules in conflict of laws. Proc. of JURIX 2019. 2019. p199-204.

This page intentionally left blank

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

Subject Index

Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

Author Index