Alma Mater Studiorum - Università di Bologna
in cotutela con University of Luxembourg - Universitè du Luxembourg

DOTTORATO DI RICERCA IN

LAW, SCIENCE AND TECHNOLOGY

Ciclo 34

Settore Concorsuale: 01/B1 - INFORMATICA

Settore Scientifico Disciplinare: INF/01 - INFORMATICA

HYBRID ARTIFICIAL INTELLIGENCE TO EXTRACT PATTERNS AND RULES FROM ARGUMENTATIVE AND LEGAL TEXTS

Presentata da: Davide Liga

Coordinatore Dottorato

Monica Palmirani

Supervisore

Monica Palmirani

Supervisore

LEON VAN DER TORRE

Esame finale anno 2022

UNIVERSITÉ DU LUXEMBOURG

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

PhD-FSTM-2022-078
The Faculty of Science, Technology and Medicine

The Department of Legal Studies

DISSERTATION

Defence held on 16/06/2022 in Bologna

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

AND

DOTTORE DI RICERCA

IN LAW, SCIENCE AND TECHNOLOGY

by

Davide LIGA
Born on 5 February 1990 in Palermo (Italy)

HYBRID ARTIFICIAL INTELLIGENCE TO EXTRACT PATTERNS AND RULES FROM ARGUMENTATIVE AND LEGAL TEXTS

Dissertation defence committee

Prof. Dr. Marco Mancarella
Professor, Università del Salento

Prof. Dr. Emiliano Lorini
Professor, Centre National de la Recherche Scientifique

Prof. Dr. Rosa Ballardini
Professor, University of Lapland

Prof. Dr. Monica Palmirani, dissertation supervisor
Professor, Alma Mater Studiorum – Università di Bologna

Prof. Dr. Leon Van Der Torre, dissertation supervisor
Professor, Université du Luxembourg

Hybrid Artificial Intelligence to Extract Patterns and Rules from Argumentative and Legal Texts

Davide Liga
2022

To my parents, who taught me the meaning of love ...

Abstract

This Thesis is composed of a collection of works written in the period 2019-2022, whose aim is to find methodologies of Artificial Intelligence (AI) and Machine Learning to detect and classify patterns and rules in argumentative and legal texts. We define our approach "hybrid", since we aimed at designing hybrid combinations of symbolic and sub-symbolic AI, involving both "top-down" structured knowledge and "bottom-up" data-driven knowledge.

A first group of works is dedicated to the classification of argumentative patterns. Following the Waltonian model of argument and the related theory of Argumentation Schemes (86), these works focused on the detection of argumentative support and opposition, showing that argumentative evidences can be classified at fine-grained levels without resorting to highly engineered features. To show this, our methods involved not only traditional approaches such as TFIDF, but also some novel methods based on Tree Kernel algorithms.

After the encouraging results of this first phase, we explored the use of a some emerging methodologies promoted by actors like Google, which have deeply changed NLP since 2018-19 — i.e., Transfer Learning and language models. These new methodologies markedly improved our previous results, providing us with best-performing NLP tools. Using Transfer Learning, we also performed a Sequence Labelling task to recognize the exact span of argumentative components (i.e., claims and premises), thus connecting portions of natural language to portions of arguments (i.e., to the logical-inferential dimension).

The last part of our work was finally dedicated to the employment of Transfer Learning methods for the detection of rules and deontic modalities. In this case, we explored a hybrid approach which combines structured knowledge coming from two LegalXML formats (i.e., Akoma Ntoso and LegalRuleML) with sub-symbolic knowledge coming from pre-trained (and then fine-tuned) neural architectures.

Keywords: Artificial Intelligence, Machine Learning, Argument Mining, Natural Language Processing

Acknowledgments

This is the end point of a 3-year experience full of discoveries, meetings, growth, drawbacks, rebirths, publications, conferences, a global pandemic, loves, friendships, fights, disillusions, dreams, travels, and much more. During this period, I had the chance to spend a wonderful year at the University of Luxembourg, where I met great people and friends. I also had the opportunity to be a visiting researcher at the Zhejiang University, in China, one of my favourite countries, working in the international MIREL (MIning and REasoning with Legal texts) project; and above all I had the opportunity to do research at the University of Bologna, Alma Mater Studiorum, the mother of all the universities of the Western World, studying and living in such a unique atmosphere.

In a sense, the city of Bologna is a paradox, it is young because it is old. The fact it is such an ancient and prestigious university makes it a never-ending flow of young people, ideas, and vivacious mindsets from all around the world. This unlocks potential, encounters, revolutions and, as a matter of fact, a peculiar perception of time. In a certain sense, Bologna is sort of trapped into its own eternal youth. And from within this beautiful golden cage where time disappears sweetly, Bologna can even talk to you (provided that you are willing to listen). For example, if you listen carefully to what is written under the noisy chaos of its archways and porticos, you will discover that there are suggestions which seem directed to you. A wall might thus tell you some peculiar philosophical thought, or a goliardic joke as well. To some extent, this city has its own soul, which is complex and beautiful. Although it is a physical place that you can see with your eyes, and touch and smell, Bologna feels like an encounter. My first acknowledgment goes to her.

However, my most important "thank you" goes to my mother and my father, and all the members of my family. Whatever I do in my life, I always owe everything to them. I have been so lucky to be part of a family which has taught me the real meaning of love, showing me how to recognize what values and wishes should be the priority in life.

I also want to to thank prof.ssa Monica Palmirani, who gave me an invaluable chance to grow, provided challenging but beautiful questions to investigate and comprehend, and offered me her precious and experienced supervision. Thanks also to professor Leendert van der Torre, who guided me with kindness and balance, giving a great example of leadership. Thanks to professor Liao Baishui and Yi N. Wang who made me feel at home in Hangzhou, one of the most beautiful places on this planet (谢谢你们让我感觉到了家的温暖).

Thanks also to the people I have met and loved thus far - those still part of my life, and those who are not anymore. And thanks to my friends; many of them arrived like a surprise, others arrived by chance or coincidence, none of them arrived for nothing.

And finally, thanks to Chance, because it never disappoints me.

Contents

Chapter 1

Introduction

This Thesis is the result of a 3-year PhD project based on the analysis and elaboration of argumentative and legal texts using Natural Language Processing (NLP) methods. It is composed of a collection of publications in peer-reviewed international conferences, along with some novel unpublished studies. The main direction of this research is the automatic recognition of **argumentative patterns** and **rules**. These two aspects are crucial for Artificial Intelligence (AI), and their automation can decisively unlock long-term goals such as the ability to reason automatically from natural language, understanding people's communicative strategies and ways of thinking, as well as checking or revising the logical coherence behind argumentative stances. A domain which can benefit particularly from this kind of research is the legal one, where laws and legal sentences could be analyzed by AI systems from an argumentative and deontic point of view, providing humans with insightful solutions and useful tools of analysis and decision.

However, when considering the **long-term goal of unlocking automatic reasoning directly on natural language**, there are many obstacles to overcome. One of the main problems is that argumentative patterns and rules can be instantiated within natural language in so many different ways, which makes it difficult for NLP algorithms to automatically recognize them. Importantly, natural language has notoriously complex characteristics: it is often uncertain, ambiguous or even incomplete and misshapen. To unlock automatic reasoning, the recognition of patterns and rules should be tackled by taking into account these key complexities of natural language. It should somehow **connect natural language to more formal layers** on which reasoners can be used.

In this introductory chapter, we shortly describe the scope of this Thesis (Section 1.1). To illustrate the scope we will firstly describe the Thesis general targets, namely the detection of argumentative patterns and rules (Section 1.1.1). Then, we will introduce the concept of "Hybrid AI" which we employed to reach our targets (Section 1.1.2). And finally, we will describe the Research Questions and the related Challenges (Section 1.1.3 and 1.1.4).

After the scope, a short survey of the State of the Art will be presented in Section 1.2, and will be focused on three key aspects: argumentation schemes (Section 1.2.1), Argument Mining (Section 1.2.2), and the classification of

deontic modalities and rules (Section 1.2.3).

After this short survey, we will briefly introduce the case studies of this Thesis in Section 1.3, with a short description of each paper, its contributions, its limitations and how it is related with the targeted research questions.

As far as the other Chapters are involved, while Chapters from 2 to 7 describe the collection of case studies, Chapter 8 concludes the Thesis and gives some ideas for the future.

1.1 Scope and Targets

1.1.1 Argumentation Schemes and Rules

We started the project by focusing on **Argumentation Schemes**, which are semi-formal stereotypical argumentative patterns written in natural language. The theory of Argumentation Schemes provides the scientific community with a rich tool of analysis that can be exploited to further improve NLP methods. Argumentation Schemes are, in fact, half a way between natural language and the sphere of argumentation. On the one side, they provide stereotypical ways in which argumentative patterns appear in natural language (so, natural language is somehow simplified into stereotypical models). On the other side, they are crucial tools for the argumentative level of analysis. It is worth pointing out that Argumentation Schemes can be considered, to a great extent, rules, especially if we analyze their structure following the Waltoninan model (86), which describes Argumentation Schemes as patterns composed by a set of premises and a conclusions. In fact, as it has been shown in the literature (52), many of these patterns are instantiations (within the sphere of natural language) of famous **rules of inference** such as *modus tollens* and *modus ponens*. However, it is important to notice that Argumentation Schemes represents just stereotypical ways of expressing inferences in natural argumentation, which means that they are an over-simplification of what people may express in reality (42).

In other words, Argumentation Schemes belongs to a layer of abstraction which is located between the logical-inferential sphere and the sphere of natural language. This position can be exploited as a useful middle layer to bridge the gap between natural language and the logical-inferential dimension by using a combination of top-down and bottom-up approaches. Figure 1.1 shows a simplified synthesis of this process.

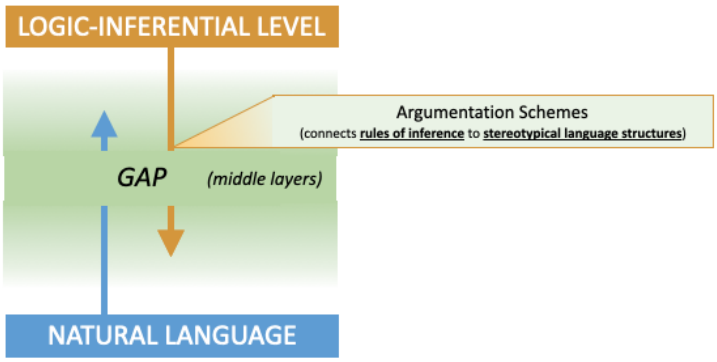


Figure 1.1: Connect natural language to the logical-inferential sphere using Argumentation Schemes, combining bottom-up and top-down approaches.

In the second part of this project, we focused on the detection of **deontic rules**, which are generally instatiated in legal natural language as obligations, premissions and prohibitions. The detection of rules in legal texts can be essential to develop automatic reasoning and Artificial Intelligence applications for the legal domain and, also in this case, there is a gap between the logical-inferential (deontic) sphere and natural language. In this case, to bridge the gap between natural language and the deontic sphere, we combined two famous LegalXML formats, namely Akoma Ntoso and LegalRuleML. Akoma Ntoso is the most important format for representing legal documents, and can connect natural language sentences to the structures of legal documents (e.g., articles, paragraphs, sub-paragraphs, and so on); meanwhile LegalRuleML can formalize legal rules while connecting them to specific structures of the legal documents. In other words, combining Akoma Ntoso with LegalRuleML provides us with a chance to connect natural language to logical rules, using the structures of legal documents as a bridge to fill the gap between natural language and the logical-inferential sphere. This process is synthesized in Figure 1.2.

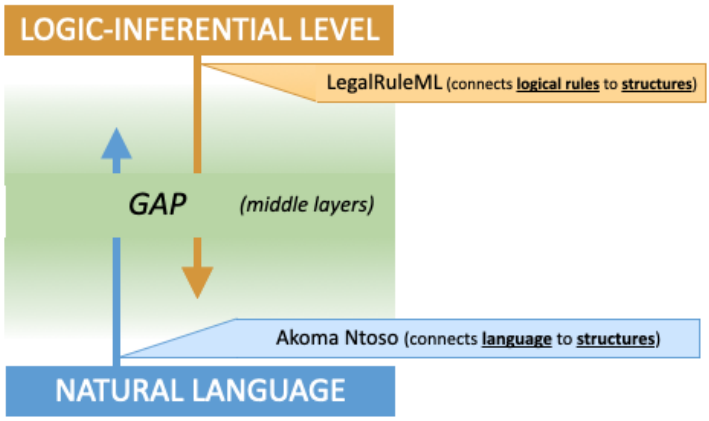


Figure 1.2: Connect legal natural language to the logic layers using Akoma Ntoso and LegalRuleML, combining bottom-up and top-down approaches.

Starting from this considerations, we can already summarize the key points on which the studies presented in this Thesis are focused:

- On the one side, they are focused on the detection/classification of argumentative patterns and rules;
- On the other side, they try to find solutions and methods to tackle the gap between the sphere of natural language and the logical-inferential sphere.

The long-term direction behind this two points is to connect logical-inferential rules to natural language, making it possible for AI systems to recognize (and perhaps reproduce) the rules which humans constantly instantiante (more or less coherently) within natural language. In this Thesis, we did not aim at reaching such a huge long-term achievement. Chasing more feasible goals, we instead aimed at tackling some of the issues and obstacles which make this long-term achievement currently unfeasible. Hardly will there be a "real" Artificial Intelligence, if we are not capable of teaching computer how to recognize and reproduce those patterns located half-a-way between the sphere of language and the sphere of reasoning, in a middle layer between logical-inferential moves and natural language schemes.

1.1.2 A Hybrid Approach to Artificial Intelligence

Before describing the research questions of this Thesis, we clarify why we use the word "hybrid" in the title. The rationale behind the use of this word comes from the above-mentioned need to find methods which can bridge the gap between the sphere of language and the sphere of logic and reasoning. We start from the assumption that these methods will need to be "hybrid", i.e. capable of combining symbolic and sub-symbolic approaches, as well as connecting data-driven Machine Learning methods (such as Deep Learning or Transfer Learning) to structured sources of knowledge (such as LegalXML knowledge bases). In fact, as we mentioned before, the detection of both argumentative patterns and rules is intrinsically connected to an upper layer of abstraction where such patterns and rules can be interpreted logically. We can thus employ methods which use structured knowledge to represent the upper layers of abstractions (i.e., rules) while using data-driven approaches to recognize patterns at the lowest levels of abstraction (i.e., natural language).

As described in (71), there might be different ways of referring to the expression "Hybrid AI". On the one side, Hybrid AI might refer to the greater and greater interaction between humans and machines. Human-Computer Interaction (HCI) is thus one of the ways in which Hybrid AI can be described. One of the challenges which derives from this first kind of Hybrid AI is related to how to correctly design the interaction of humans and machines and how

these design choices[1] can affect the ethical and normative sphere of AI. These design choices in HCI are important to avoid falling into the dychotomy *good AI* vs *bad AI*, where "good" and "bad" are often not very useful indicators of a more general oversimplification.

A second way of referring to the expression "Hybrid AI" is directly connected to the greater and greater combination of symbolic AI with sub-symbolic AI. In this perspective, the challenges are instead related to how to combine these two approaches successfully. While a growing number of studies is tackling this challenge, we would like to stress the perspective suggested in (22), where the authors envisage a combination of "top-down" and "bottom-up" methods where the former ones are data-drive methods, while the latter ones are methods based on structured knowledge (e.g., knowledge graphs). This idea of Hybrid AI is close to what we propose as Hybrid methods in this Thesis (especially at the end of the work, when we discuss the automatic detection of deontic rules from legal natural language).

As we mentioned already, this Thesis partially matches the first definition of "hybrid", since it goes towards the long-term direction of allowing AI to recognize patterns of reasoning instantiated within natural language. This can clearly have an influence in HCI, because it could allow machine to better "understand" humans, and it could allow humans to better explain machines behaviour. This explainability is desirable and can have strong consequences in the interaction of humans and machines allowing humans to design better regulations for AI systems. However, as we said, these are long-term goals.

As far as the more reachable goals of this Thesis are concerned, our definition for the word "hybrid" will be the second one. In fact, this collection of works aims at finding working methodologies of hybrid symbolic and sub-symbolic approaches capable of combining the different layers of abstraction (i.e., the layer of natural language, and the layer of logical rules and inferences) through a connection of top-down and bottom-up methods. With this purpose, we designed hybrid solutions combining symbolic knowledge (such as tree-structured data representations and LegalXML knowledge-bases) to sub-symbolic or data-driven methods coming from the training or the fine-tuning of neural architectures.

1.1.3 Research Questions

Going towards the direction described above, this work consists of a collection of publications and studies whose research questions can be summarised as follows:

- (Q1) Can NLP methods detect and classify **argumentative support** and **opposition**?

[1] Some famous choices of design are for example the Human-in-the loop (HITL), Human-on-the-loop (HOTL), and Human-in-command principles (HIC). HITL envisage the human intervention in every decision cycle of the AI; HOCL envisage human supervision in the design of cycles and in their general supervision; HIC evisage a greater human control and ability to choose when and how to intervene in any situtation.

- (Q2) Can we fill this gap between the natural language and the logic sphere by **combining top-down and bottom-up approaches**, and **symbolic and sub-symbolic AI**?

- (Q3) Can NLP methods **detect rules** and **deontic modalities**?

As can be seen from **Q1**, this work tackled both supportive and oppositive argumentative patterns. In fact, since the concepts of support and opposition (attack) are crucial in logic and argumentation, it is important to verify the ability of classifiers to recognize both types of argumentative patterns. The tasks of detecting and classifying argumentative support and argumentative opposition can be considered under the domain of Argument Mining, since it is related to both NLP and Argumentation (47).

The research question **Q2** directly targets the need of bridging the gap between natural language (which is complex, sometimes ambiguous or uncertain, and sometimes even incomplete or misshapen) and the logical-inferential sphere (where one can reason using formal or informal logic). This second research question is probably the most challenging, but this work tried to elaborate at least some practical and preliminary directions to go towards this long-term goal of filling the above mentioned gap which keeps us separated from the ability to apply automatic reasoning directly on natural language.

The research question **Q3** focuses on the detection of rules, and we tried to answer this question by focusing on deontic rules and by designing a hybrid method of Machine Learning where structured symbolic information, typically used to represent rules, is combined with sub-symbolic learning methods.

1.1.4 Challenges and Proposed Solutions

Considering the above-mentioned research questions, one of the obstacles is that it can be **difficult to recognize some argumentative structures** (Q1), since the information channelled in natural language can be incomplete and ambiguous (which is also a common problem in NLP in general). But also because the theory about Argumentation Schemes and, particularly, the theory about their classification (which is to say how are Argumentation Schemes related among each other) is still a matter of philosophical-ontological debate (52; 26; 85). In fact, Argumentation Schemes are semi-abstract models which sometimes overlap when they are instantiated within the more complex and ambiguous context of natural language. We can define this problem as **P1**. Another important limitation is that there are **not enough datasets** specifically designed for this kind of tasks. So, even considering those schemes which are theoretically sound, it is difficult to find enough data for certain schemes and the creation of new datasets is costly. We can define this problem as **P2**. Also, another important issue is that the creation of Machine Learning algorithms can be very time-consuming because of the **need to design highly engineered features**. We can define this problem as **P3**. Finally, the most challenging issue is probably the one tackled by the research question Q2, namely how to fill the gap between language and the logic-inferential dimension. In fact,

natural language is complex and variable, it can be ambiguous or uncertain, and it can even be incomplete or misshapen. These characteristics generate a huge challenge: how can we connect natural language to the logical-inferential sphere? We call this issue of the **distance, or gap,** between natural language and the logical inferential sphere **P4**.

All these major problems are synthesized in Table 1.1.

Problem:	**Description:**
P1	Ontological complexity of argumentative patterns
P2	Scarcity of data both for the detection of Argumentation Schemes and for the detection of rules
P3	Complex features are often needed
P4	Difficulty of bridging the gap between natural language and the logical-inferential sphere

Table 1.1: Description of some major challenges (P1, P2, P3 and P4)

Even if these limitations makes this kind of research challenging, this work attempted to cope with some of these obstacles. To **tackle P1** (the ontological complexity of argumentative patterns and the difficulty of recognize them), we studied the classification made by Douglas Walton (86) and the analysis proposed by Macagno (52) about the classification of Argumentation Schemes. Starting from this theoretical background, we designed experimental settings which are focused on easily recognizable Argumentation Schemes (some of the targeted Argumentation Schemes are the Argument from Expert Opinion, the Argument from Negative Consequences, the Slippery Slope Argument(86; 85), and other related patterns) and we assessed the ability of our methods to be precise by training NLP classifiers not only on the recognition of very different argumentative patterns, but also on **the recognition of argumentative patterns which are ontologically related**[2].

To **tackle P2** (the limited availability of data), this Thesis proposes multiple solutions. As far as the detection of argumentative patterns is concerned, **some existing Argument Mining datasets have been used, and even combined**, to facilitate the detection and classification of argumentative patterns which were ontologically close. Also, a **new dataset** has been created for the detection of Argumentation Schemes (i.e., we annotated public available sentences as belonging to specific argumentative patterns). As far as the detection of rules is concerned, we focused on deontic rules tackling the problem of

[2]For example, the Argumentation Scheme from Expert Opinion is part of an umbrella category which is based on the testimony from an certain source of information (35), under which other patterns can be found (for example arguments coming from the reference to statistics or other testimonies).

the scarcity of data by leveraging the structure and the meta-data provided in LegalXML formats (i.e. AkomaNtoso, LegalRuleML). These formats provide us with important information related to the structure and content of legal texts. Starting from the assumption that, in legal documents, argumentative information and rules are closely related to the structure of the document where the information is hosted, we developed a hybrid approach **combining symbolic and sub-symbolic AI for the recognition of legal rules**, including deontic modalities (e.g. obligations, prohibitions). In this way, we **extracted** sentences and their relative classes by leveraging LegalXML documents and knowledge bases and this labelled dateset was then used to feed Machine Learning classifiers designed to detect rules and deontic modalities.

As mentioned before, even when data exist, a difficult aspect is to design what features should be taken into account for the design of the Machine Learning algorithms. To **tackle P3**, we adopted two innovative approaches: **using Tree Kernel algorithms and using Transfer Learning methodologies**. The first ones are algorithms capable of leveraging the internal grammatical and syntactical structures of textual data in order to classify natural language without the need to design complex features. The second approach, which is the protagonist of a great step forward in the recent advancements of NLP, is Transfer Learning, which allows researchers to exploit huge pre-trained neural architectures in downstream tasks; in this last case, features are internally projected within the high-dimensional space determined by the weights and parameters of the pre-trained/fine-tuned neural architectures.

The hugest problem to solve is probably **P4**, namely the difficulty of connecting natural language to the logical-inferential sphere, which is directly connected to Q2 (P4 is the obstacle tackled by Q2). This problem can be solved in two ways. The first way is to model language into representations which can handle natural language complexity, uncertainty and ambiguity while providing all the available information that may channel argumentative inferential steps or rules. This solution, which we partially discussed in (42) and (43), can be extremely difficult to achieve and requires huge efforts. For this reason, in this Thesis we focused on a second potential solution to P4. This solution is based on the use of Hybrid AI, namely the combined use of symbolic and sub-symbolic approaches, where structured knowledge is mixed with data-driven methods.

Table 1.2 synthesizes the proposed solutions for the above mentioned major challenges and problems (i.e. P1, P2, P3 and P4).

Problem:	Proposed solutions:
P1	• Classifying schemes which are **ontologically related**
P2	• **Combining** existing datasets • **Creating** a new dataset • **Extracting** labelled data from LegalXML
P3	• Using **Tree Kernels** • Using **Transfer Learning** methods
P4	• Using **Hybrid** symbolic and sub-symbolic methods, combining structured source of knowledge with data-drive sub-symbolic approaches

Table 1.2: Description of the solutions to P1, P2, P3 and P4.

A synthesised overview which shows all the case studies described in this Thesis is reported in Figure 1.3, along with the relative research questions which have been tackled by each case study.

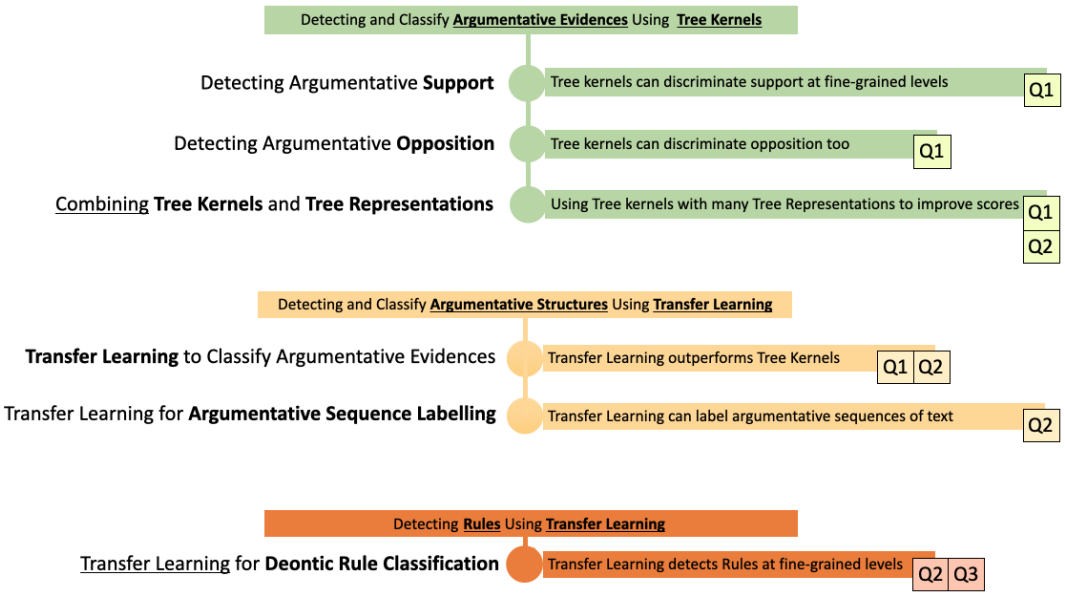


Figure 1.3: Case studies (on the left) with some of the relative achievements (on the right), with a reference to their related research questions.

In the following Section, we show a short description of the collection of case studies presented in this Thesis. These short descriptions will give a brief introduction to the content of the papers. We hope to facilitate, in this way, the general comprehension of the path that has been undertaken and, above all, of the directions which have been selected during the creation of each case study.

1.2 State of the Art Survey

1.2.1 Argumentation Schemes

An important focus of this Thesis is the classification of argumentative patterns. The definition of argumentative pattern is strictly related to the definition of a model of argument, which establishes what are the basic components which characterize any single argument, and what are the roles of these components within the structure of the argument. A famous example of model of argument is the Toulmin model, according to which any well-structured argument should be composed of six basic elements: claim, ground, warrant, backing, qualifier, rebuttal[3].

In this Thesis, we will mainly refer to the model of argument proposed by Douglas Walton, including the related theory about argumentation schemes (86). According to Walton, arguments are composed by a set of premises and a conclusion. Starting from this assumption, Walton created a compendium of stereotypical argumentative patterns, i.e., argumentation schemes which describes some common stereotypical ways in which people use natural language to argue. Moreover, for each of these schemes, a set of critical questions is provided, which is designed to assess the argumentative strength of the given scheme. It is worth remarking that the identification of argumentation schemes and their classification is an open research area which depends on the criteria used to find similarities and differences among stereotypical argumentative patterns. While Walton proposed a set of nearly 30 schemes, other scholars suggested different sets of schemes. For example, Pollock identifies less than 10 schemes (65), Grennan identifies more than 50 schemes (23), Katzav and Reed identifies more than 100 schemes (29).

A well-known example of argumentation scheme is the argument from expert opinion (reported in Table 1.3). Without considering the critical questions attached to it, this scheme is composed of just two premises and one conclusion. This argumentation scheme is very frequent in natural argumentation, and easy to find in public speeches. Intuitively, the argumentative strength of this stereotypical way of arguing comes from the appeal to the authority of an expert.

[3]The claim is the conclusion to be established by the argument; the ground (also known as "data" or "evidences") refers to the reasons or evidences supporting the claim; the warrant is the principle, provision or chain of reasoning that connects the grounds/data to the claim; the backing is an additional support, justification or chain of reasoning that back up the warrant; the qualifier is an optional element which defines the relative strength of the warrant (it can be expressed by locutions such as 'possibly', 'necessarily', etc.).

Table 1.3: The structure of the argumentation scheme known as "argument from expert opinion".

Premise 1	Source E is an expert in subject domain S containing proposition A.
Premise 2	Source E asserts that proposition A (in domain S) is true (false).
Conclusion	Proposition A may plausibly be taken to be true (false).

Another famous argumentation scheme is the argument from negative consequences, which is reported in Table 1.4.

Table 1.4: The structure of the argumentation scheme known as "argument from negative consequences".

Premise	If A is brought about, bad consequences will plausibly occur.
Conclusion	Therefore A should not be brought about.

The important aspect to remark here is that these schemes can be very useful because they provide a way to connect natural language to the logical-inferential sphere of natural argumentation. For this peculiar position, halfway between natural language and argumentation, schemes can be useful for developing computational methods of extraction of argumentative information from natural language.

So far, only few studies tried to use Natural Language Processing and Argument Mining to identify argumentation schemes within natural language. Among the first attempts, there is (17). In this Thesis, we will offer some approaches which go towards this direction, which is still to be explored.

Although one of the obstacles in the direction of automatically detect argumentation schemes is the fact that the classification of schemes is still an open research area, important steps toward the aim of creating a robust system of classification of argumentation schemes has been provided by (52). From these studies, (35) proposed an approach to classify schemes following a hierarchy of classification choices which is presented as a decision tree and which is closely related to both the work of Macagno and Walton about arguments and argumentation schemes. Probably, the improvements in our ability to develop computational models of arguments is closely connected to the improvements on this theoretical side, which is to say, the improvements in our ability to identify and classify argumentative schemes and pattern.

1.2.2 Argument Mining

Argument Mining is a relatively new domain which aims at connecting Natural Language Processing and Argumentation Theory to develop computational approaches able to automatically find and extract arguments, typically expressed as inferential structures of reasoning within natural language. Argument Mining often focuses on the following tasks:

- Detecting/classifying argumentative components and their boundaries (discarding non-argumentative components);
- Detecting relations among argumentative components;
- Reconstructing argumentative structures (following a given argument model).

Although there is no consensus about a definitive Argument Mining pipeline, a good starting point can be found in (34), where the argument analysis is presented as a 4-step process:

- Text segmentation
- Argumentative/non-argumentative
- Simple structure
- Refined structure

The first step (text segmentation) aims at extracting the fragments of text which are part of an argument, and which are usually expressed as Elementary Discourse Unit (EDUs) or Argumentative Discourse Units (ADUs) (34). This task can be tackled as a sequence labelling task (i.e., the same kind of task performed for Named Entity Recognition). In this regard there have been only few studies which tried to detect the exact span of argumentative components of text using sequence labelling. One of the first attempts to label argumentative sequences is (77), where argumentative sequences have been modeled by using highly-engineered features (including Structural, Syntactic, Lexical-Syntactic and Probabilistic elements) and the classification employed Conditional Random Field (CRF) together with an averaged perceptron. Another study improved the performances by employing a BiLTSM neural network (2). However, this Argument Mining task is still an open research area which requires further efforts, also considering the recent outbreak of language models and transfer learning techniques which can certainly allow for better and better results. In this Thesis, we offered our contribution towards this direction.

The second step (argumentative/non-argumentative) is often performed together with the first and aims at discarding the segments which are not relevant to the targeted argumentative structures. However, it can also be performed as a separate classification task, depending on the design of the Argument Mining pipeline.

The third step is about finding the relations among the extracted segments, which generally consists in identifying the relations of attack and support between segments. In this regard, an important contribution has been offered by the studies of Stab and Gurevych (78; 79).

The last step is about detecting more refined argumentative structures, identifying for example specific argumentation schemes. This step depends on what theory of schemes is taken into account by the analysts - e.g., the

Waltonian argumentation schemes, or other scheme theories (34). There have been only few works going towards this direction. One of the few ones are (17) and (33), which achieved similar results by using highly engineered features. However, this task of Argument Mining is still to be explored and further research is needed.

In our works, we focus on all these steps. For example, we performed a sequence labelling task to recognize the boundaries of argumentative components vs non-argumentative components (thus focusing on the first two points). We also classified different kinds of argumentative support and opposition (which is more related to the recognition of the argumentative structure described in the third step). Moreover, we focused on the classification of argumentative text as belonging to specific argumentative patterns and schemes (which is closer to the detection of the refined structure described in the fourth step).

1.2.3 Deontic Modality and Rule Classification

A huge and growing research domain is legal knowledge extraction. Within this research area, there are crucial tasks which need further efforts from the scientific community. For example, the automatic classification of rules and deontic modalities. So far, only few works have focused on this task.

There have been different attempts to extract rules using complex methods of extraction, for example detecting noun and verb phrases (90), or exploiting syntactic dependencies between terms (16). Among the few studies which attempted an automatic classification of deontic rules there is (30), which employed word lists, grammars and heuristics to extract obligations and other targets such as rights and constraints. Another study (21) used Machine Learning to extract different kinds of normative relationships (i.e., prohibitions, authorizations, sanctions, commitments and powers). Moreover, (84) used active learning with Multinomial Naive Bayes, Logistic Regression and Multilayer Perceptron classifiers to recognize, among the other targets, prohibitions and permissions. The studies which focuses on the automatic classification of deontic modalities using neural approaches are very few. Among them we found (59) and (10), which used Bi-LSTM architectures (the second study also employed a self-attention method). Other two studies are (28) and (74), which exploited the potential of the novel and powerful Transfer Learning approach.

The last part of this Thesis offers a similar approach by using Transfer Learning, combining its powerful sub-symbolic potential with the symbolic reliability of legal Knowledge Representation.

1.3 Case studies

1.3.1 Detecting Argumentative Support

As already mentioned, the first research question is about the detection of argumentative patterns of opposition and support. At the beginning of this project, we focused on the combined use of Tree Kernels and TFIDF methods for the classification of argumentative support. Our first work is a peer-reviewed study published in 2019, in the context of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), leading conference in the field of NLP. More specifically, it is one of the papers of the 6th ArgMining Workshop, which is currently the leading meeting related to Argument Mining. The title of this paper is "Argumentative Evidences Classification and Argument Scheme Detection Using Tree Kernels" (36).

In a second short paper called "Comparing Tree Kernels performances in argumentative evidence classification" and published in the context of CLADAG 2019, conference sponsored by the Italian Statistical Society (37), the experiment was extended considering two types of Tree Kernel algorithm: the Smoothed Partial Tree Kernel (SPTK), which was used in the first paper, but also the Partial Tree Kernel (PTK).

In this Thesis we just selected the first study, which will be now described from a general perspective. The complete version of the paper, instead, can be found in Chapter 2.

Argumentative Evidences Classification and Argument Scheme Detection Using Tree Kernels

The purpose of this paper is to deploy a novel methodology for the classification of supportive argumentative patterns, generally referred to as "supporting evidences". The proposed methodology is based on the idea that the use of Tree Kernel algorithms can be a good way to discriminate between different types of argumentative stances without the need of highly engineered features. This possibility to classify argumentative text without the need of complex features, makes Tree Kernel methods very useful in different Argument Mining sub-tasks. In this case, the focus is the classification of argumentative support (or evidence), which is a key step toward the automatic classification of Argumentation Schemes.

Interestingly, this paper was the first attempt to use Tree Kernels for the classification of different types of Argumentation Schemes' evidences. Moreover, it shows a clear comparison of the performance of a Tree Kernel classifier compared to the performance of a TFIDF classifier, along with a combination of both.

So, starting from the main research questions described before, this paper more specifically targets the first research question (that we called Q1). The resulting targeted research sub-questions are the following:

1. Is it possible to perform a fine-grain discrimination between different kinds of argumentative evidence by using Tree Kernels and TFIDF methods, thus avoiding the need of engineering sophisticated feature vectors? (**Q1**)

2. Can Tree Kernel methods overcome TFIDF methods in the classification of supportive argumentative patterns? (**Q1**)

The classifiers of this first paper exploited the ability of Tree Kernels to calculate similarities between tree-structured sentences, considering the similarity of their fragments. The experiment was performed on two famous Argument Mining datasets, which share a similar labelling system.

Some major contributions of this paper are:

- It is the first research on how Tree Kernels methods can be used to discriminate Argumentation Schemes without the need of highly engineered features (**Q1** and **P3**).

- It shows a comparison between traditional TFIDF and Tree Kernels, along with a combination of both (**Q1**).

- It shows that two famous IBM datasets can be used together, not only to increase the amount of data to be trained, but also to have a more reliable assessment on the ability of the algorithms to generalize over different data (**Q1** and **P2**).

- It shows that it is possible to use Tree Kernels to apply fine-grained discriminations among different stances of support (**Q1** and **P1**).

Clearly, being the first experiment, this paper also had some aspects that needed to be improved. For example, some major limitations of this paper are:

- It is focused on supportive evidences only, which is to say, argumentative evidences whose aim is *to support* a conclusion, not to go against it.

- The chosen algorithm of this paper only considers one type of Tree Kernel, namely the Smoothed Partial Tree Kernel (SPTK), the experiment did not take into account other types of Tree Kernels (this limitation has been tackled later in (37)).

- The paper only employs Grammatical Relation Centered Tree (GRCT) representation, without considering other types of tree-structured data representations.

- The paper does not consider the use of *n*-grams for the generation of the TFIDF representations.

1.3.2 Detecting Argumentative Opposition

To tackle some of the limitations described in the first papers, another paper has been written and published at *Rules and Reasoning, Third International Joint Conference RuleML+RR*, in 2019. The title of this paper is "Detecting 'Slippery Slope' and Other Argumentative Stances of Opposition Using Tree Kernels in Monologic Discourse" (39). In this case, we tried to see whether Tree Kernels could classify argumentative opposition.

In a further paper, this scenario was extended offering a clearer view on how Tree Kernels and TFIDF (with and without *n*-gram) perform, also considering them separately. The title of this paper is "Classifying argumentative stances of opposition using Tree Kernels" (38) and was presented at the 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2019). While the complete version of this paper can be found in Chapter 3, the following section is its introductory synthesis.

Classifying argumentative stances of opposition using Tree Kernels

In this paper, the focus was not on argumentative stances of support but, instead, on argumentative evidences of opposition. And, importantly, the dataset was created completely from scratch. This paper was an opportunity not only to show the effectiveness of the methodology of the previous two papers, but also a way to describe a brand new dataset, even if not completed yet, to assess the methodology of the first paper on a completely different kind of argumentative data.

Broadly speaking, the paper takes inspiration from the previous achievement, offering an innovative methodology to classify argumentative stances of opposition in a monologic argumentative context. In particular, the paper explores the possibility of classifying opposition stances by training multiple classifiers to reach different degrees of granularity. As already said, discriminating support and opposition stances can be particularly useful when trying to detect Argumentation Schemes, which is one of the most challenging sub-task in the Argument Mining domain. In this sense, the approach can be also considered as an attempt to classify stances of opposition that are related to specific Argumentation Schemes.

The targeted research sub-questions which have been tackled by this paper can be synthesised as follows:

1. Given the previous results on the classification of argumentative support, can the same methodology be applied on argumentative stances of opposition? (**Q1**).

2. Considering that the previous results for TFIDF just involved monograms, can *n*-grams influence TFIDF performance significantly? (**Q1**).

Apart from clarifying better the work of the previous papers, some of the contributions of this paper are the following ones:

- Importantly, the experiment was performed on a brand new dataset of more than 600 sentences (**P2**).
- The paper offers a granular approach, which shows how performances can reach fine-grained targets (**Q1** and **P1**).
- This paper together with (39) are the first works which focuses on the classification of Argumentation Schemes of opposition (while the first paper was the first work which focused on supportive evidences) (**Q1** and **P1**).
- This paper also includes *n*-grams, showing how they affect results (**Q1**).
- It compares the performance of classifiers purely based on Tree Kernels with the performance of classifiers which combine TFIDF and Tree Kernels for the detection of stances of opposition (**Q1**). This comparison shows that the combination Tree Kernel + TFIDF generally outperforms pure TFIDF/*n*-grams and pure Tree Kernel classifiers. This is somehow a confirmation of what has been assessed in the first paper, where Tree Kernels and monogrammic TFIDF were studied separately on stances of argumentative support; this time, however, the scenario is related to the stances of argumentative opposition, and takes into account also *n*-grams.

The main limitation of this paper is the following:

- Only one type of Tree Kernel has been used; trying different Tree Kernel algorithms (testing their configuration with different parameterizations values) could have been useful.

1.3.3 Combining Tree Kernels and Tree Representations

The papers described so far tackled the classification of argumentative support and opposition using Tree Kernels. We will now describe the last paper in which we tried to give a complete exploration of all the potentialities of Tree Kernels by combining different types of Tree Kernel function with different types of tree representations. This paper, which concludes the exploration of Tree Kernel algorithms for the detection of argumentative patterns, has been published under the name "Combining tree kernels and tree representations to classify argumentative stances", at *Advances in Semantics and Linked Data: Joint Workshop Proceedings from ISWC 2020*. The complete version of the paper is in Chapter 4.

Combining tree kernels and tree representations to classify argumentative stances

This paper gathered together all the efforts of the previous papers about Tree Kernel methods. It investigates how the combination of different tree

representations with different Tree Kernel functions influences the results of the classification of both supportive and oppositive evidences.

In this case, the research sub-questions are the following:

1. Are there differences in the performance when trying out different combinations of Tree Kernels and tree representations? (**Q1** and **Q2**)

2. If so, which is the most performative combination in the classification of the stances of support? (**Q1**)

3. And which is the most performative combination in the classification of the stances of opposition? (**Q1**)

The main contribution are:

- The paper shows how the performance of classifiers is influenced not only by the kind of Tree Kernel algorithm, but also by the kind of tree representation (**Q1**).

- The paper shows a comprehensive comparison of the main combinations between Tree Kernels and Tree Representations (**Q1**).

- Also 'smoothed' and 'compositional' trees are considered, providing a hybrid method which combines tree kernel calculations with calculations based on semantic vector representations (**Q2**). In other words, a semantic layer is inserted, which influences the calculation of the similarity performed by the tree kernel function.

1.3.4 Transfer Learning to Classify Argumentative Evidences

After having successfully explored Tree Kernels methods, Google brought about a big success in the domain of NLP by presenting BERT (15), which paved the way for the proliferation of language models and Transfer Learning approaches. These method showed to be capable of improving the State of the Art scores of a great number of NLP benchmarks. For this reason, we started watching at these methodologies to see if they could outperform our previous experiments, and to see if they could provide us with more tools to tackle our research questions.

The first paper which employed Transfer Learning, was "Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification" and it was published in the context of the famous COMMA conference (leading conference in the domain of argumentation), more precisely at the 20th Workshop on Computational Models of Natural Argument (CMNA), which is the most important Argument Mining event together with the previously mentioned ArgMining. The complete version of this paper is in Chapter 5.

Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification

The paper describes a Transfer Learning methodology aiming at discriminating evidences related to Argumentation Schemes using three different pre-trained neural architectures. Although Transfer Learning techniques are increasingly gaining momentum, the number of Transfer Learning works in the field of Argument Mining is relatively small and, to the best of our knowledge, no attempt has been performed towards the specific direction of discriminating evidences related to Argumentation Schemes. The research question of this paper is whether Transfer Learning can discriminate Argumentation Schemes' components, a crucial yet rarely explored task in Argument Mining. Results show that, even with small amount of data, classifiers trained on sentence embeddings extracted from pre-trained transformer-based language models can achieve encouraging scores, outperforming our previous results on evidence classification.

The main research sub-questions for this paper are:

1. Can Transfer Learning methods discriminate different Argumentation Scheme's evidences? (**Q1**)
2. Can Transfer Learning methods outperform Tree Kernel methods and TFIDF methods? (**Q1**)

Some of the contributions of this paper are:

- It shows results for 3 pre-trained neural architectures and 3 classification algorithms (**Q1**).
- It shows that the contextual embeddings produced by pre-trained language models can provide powerful 'learned' semantic representations (**Q2-P4**).
- It shows that Transfer Learning methods can reach the best performances despite being employed on small datasets (which tackles **P2**).

The limitations of this work can be synthesised as follows:

- Although this method seems to have encouraging results, the range of Argumentation Schemes which have been targeted is still quite restricted.
- Moreover, sentence embeddings are just one possible approach of Transfer Learning. It did not explore the so-called 'fine-tuning' approach (which will be employed in the next case study).

So far, the introduced papers have been all related to classification tasks, the next case study is instead related to the task of sequence labelling, which is crucial to connect portions of text to specific logical-inferential categories.

1.3.5 Transfer Learning for Argumentative Sequence Labelling

After we proved that Transfer Learning methods outperform our previous approach (i.e., the Tree Kernel method), we used it in the context of a more difficult task, namely the task of sequence labelling. Sequence labelling is the process of labelling span of text. Named Entity Recognition is an example of sequence labelling task. More specifically, the next paper that we are going to introduce, and which is fully available in Chapter 6, describes an *Argumentative* Sequence Labelling task, performed by fine-tuning a pre-trained language model (while the previous paper employed another Transfer Learning approach, based on the extraction of sentence embeddings).

Argumentative Sequence Labelling using Transfer Learning

The paper "Argumentative Sequence Labelling using Transfer Learning" is still to be published. This paper describes a Transfer Learning method to detect the exact span of argumentative components (i.e. premises, conclusions). This task is defined as "Argumentative Sequence Labelling" and can be helpful for the second research question, related to the ability to connect language to the logical-inferential sphere (Q2-P4). In fact, it may allow for the connection of specific portions of text to specific classes related to single argumentative components. This granularity is arguably necessary to reach the long-term goal of filling the gap between language and logic.

For this paper, the research sub-question can be expressed as:

1. Can we use Transfer Learning to detect the exact span of argumentative components? (**Q2-P4**)

The contributions of this paper are the following:

- It shows that Transfer Learning can operate Argumentative Sequence Labelling by using two famous Argument Mining datasets (**Q2-P4**);
- It shows the process of sequence labelling at different levels (i.e., token level, span level) and using two labelling strategies (i.e., BILUO and BIO) (**Q2-P4**) .

Even if its results are encouraging, the paper shows that at the token level there is still large room for improvement.

1.3.6 Transfer Learning for Deontic Rule Classification

The last paper of this Thesis, which is fully available in Chapter 7, is currently unpublished and is related to the last research question of this project (**Q3**). More specifically, it assesses whether it is possible to detect rules and deontic modalities using Transfer Learning.

Transfer Learning for Deontic Rule Classification

In this last study, we developed classifiers which automatically detect rules and deontic modalities using a combination of symbolic and sub-symbolic methods. More precisely, we employed two famous LegalXML formats (namely, LegalRuleML and Akoma Ntoso) to feed different neural architectures for the generation of 'fine-tuned' language models capable of classifying rules and deontic modalities. This process has been applied in the context of the European General Data Protection Regulation (GDPR), using its Akoma Ntoso representation along with its LegalRuleML modeling.

One of the research questions of this work is Q3 ("Can NLP methods detect rules?"). More specifically, a sub-question can be synthesized as follows:

1. Can we leverage structures and meta-data of LegalXML documents to facilitate the detection of deontic rules and modalities? (**Q3**)

The contributions of this paper are related not only to Q3, related to the detection and classification of rules, but also to Q2, which is the most challenging among our research questions. In this regard, the contributions can be synthesized as follows:

- The paper shows that it is possible to detect and classify rules and deontic modalities (**Q3**);
- For this task, three fine-tuned neural architectures are compared (**Q3**);
- The paper offers a Hybrid AI approach which combines symbolic "top-down" knowledge (using Akoma Ntoso together with the biggest LegalRuleML knowledge base) with a sub-symbolic ("bottom-up", data-drive) neural approach (**Q2-P4**).
- The paper shows how to retrieve working labelled data by combining and leveraging the information contained in Akoma Ntoso and LegalRuleML (**P2**);

A limitation of this paper is that its deontic classification only considers obligations and permissions. In future works, we will also target prohibitions.

This paper clearly does not solve the problem of filling the gap between language and the sphere of logical-inferential rules (P4). However, it offers a working solution to partially cover this gap. More efforts are still required to connect *portions of natural language* to *portions of logical rules*. In this regard the use of LegalRuleML can be crucial in future research projects, because it provides information about the single components of logical formulæ within legal sources, as well as the reference to the portion of natural language where these formulæ are located.

In future works, we will focus on how to match legal rules' internal components (at the level of LegalRuleML) to their respective portions of text (at the level of natural language). This can unlock extremely exciting perspective for the domain of Legal Artificial Intelligence.

Chapter 2

Detecting Argumentative Support

Original title: **Argumentative Evidences Classification and Argument Scheme Detection Using Tree Kernels**

Abstract

The purpose of this study is to deploy a novel methodology for classifying different argumentative support (supporting *evidences*) in arguments, without considering the context. The proposed methodology is based on the idea that the use of Tree Kernel algorithms can be a good way to discriminate between different types of argumentative stances without the need of highly engineered features. This can be useful in different Argumentation Mining sub-tasks. This work provides an example of classifier built using a Tree Kernel method, which can discriminate between different kinds of argumentative support with a high accuracy. The ability to distinguish different kinds of support is, in fact, a key step toward Argument Scheme classification.

2.1 Introduction to the Argument Mining Pipeline

Argument Mining (AM) is a field of growing interest in the scientific community and a growing number of works have been written about this topic in the last few years (9; 47). Since it is a relatively young research domain, its specific target area is huge and its taxonomy is relatively flexible, for example *Argument Mining* and *Argumentation Mining* are used interchangeably. In spite of this flexibility, it is possible to define a unique and broad target, which is the extraction of argumentative units and their relations from data.

Another characteristic of AM is its close connection with other domains such as Knowledge Representation and Reasoning, Computational Argumentation, Information Extraction, Opinion Mining, Human-Computer Interaction. Also, there is a strong relation between AM and Natural Language Processing (NLP), since language is the means by which humans express arguments.

Habernal et al. (25) noticed a relation between Opinion Mining (also known as Sentiment Analysis) and Argument Mining. The former aims to detect *what* people say, the latter wants to understand *why*. For this reason, Lippi and Torroni (47) consider AM as an evolution of Opinion Mining in terms of targets.

Being AM a multifaceted problem, it can be useful to imagine it as a pipeline (with much research focused on one or more of the involved steps). For example, Lippi and Torroni (47) described it as a three-steps process, from a Machine Learning perspective. The first step is to discriminate between argumentative and non-argumentative data; the second step is to detect argument boundaries; the third step is to predict the relations between arguments or between argumentative components. The second and third step are strictly dependent on the underlying argumentative model (the most frequently used is the claim/premise model described in (86), while another frequent choice is the model proposed by (81)). Cabrio and Villata (9) proposed a simpler two-step pipeline, where the first phase is the identification of arguments and the second step is the prediction of argument relations. In this case, the first step involves not only the classification argumentative vs non-argumentative, but also the sub-tasks of identifying arguments components (claims, premises, etc.) and their boundaries. While, the second step comprises predicting the heterogeneous nature of argument relations (e.g., *supports*, *attacks*) and the links between evidences (premises) and claims (conclusions). For the purposes of this paper, this two-step pipeline will be considered.

In an ideal AM pipeline, after having detected the argumentative units, their relations (e.g., premises, conclusions) and the nature of their relations (e.g., support, attack), the further step is to fit this argumentative map into a suitable Argument Scheme (e.g., argument from Expert Opinion, argument from Example).

To do so it is necessary to develop classifiers able to discriminate between different kinds of argumentative evidences. This work is an attempt to give a contribution to the achievement of this sub-task of the pipeline, finding a working methodology to discriminate between different types of support

prepositions (or *evidence*), since being able to classify different kind of support is a crucial aspect when dealing with the classification of Argument Schemes.

In particular, the proposed methodology is based on the use of Tree Kernels (TKs).

2.2 Related Works

This work presents an approach for classifying evidence typology within arguments using Tree Kernels (TKs, described in (57)) with the aim to facilitate the detection of Argument Schemes. TKs have already been used successfully in several NLP-related works, for example in semantic role labelling (58), metaphor identification (27) and question answering (18). However, the application of TK in the domain of AM has been relatively limited compared to other methodologies mostly that are dependent on highly engineered feature sets. One of the first use in Argumentation Mining was proposed by Rooney et al. (72), who simply employed sequences of Part-of-Speech tags. At that moment, however, the Argumentation Mining community was still too young. Some years later, Lippi and Torroni (47) suggested to exploit the potentialities of TKs for detecting arguments (the first step in the Argument Mining pipeline) and presented a promising tool for automatically extract arguments from text (48). Interestingly, TKs have been used to specific domains: Mayer et al. (56) exploited them for an AM approach related to Clinical Trials, while promising results have been achieved also in the legal domain (49; 50). TKs have also been used in (83) for analyzing the similarities between argumentative structures, thus focusing not on the level of the sentences (step one), but on the level of the argumentative relations (step two of the Argument Mining pipeline).

To the best of our knowledge, this is the first attempt to use TKs in the very last part of the Argument Mining pipeline. In fact, the approach presented here aims to differentiate different kinds of evidences (or *premises*), which is an important sub-task when trying to detect the most suitable Argumentative Scheme.

Other studies tried to classify arguments by scheme using different approaches. For example, Feng and Hirst (17) created a complex pipeline of classifiers that achieved and accuracy ranging from 63 to 91% in one-against-others classification and 80-94% in pairwise classification. In another study Lawrence and Reed (33) achieved a similar result, with F-scores ranging from 0.78 to 0.91. However, these two works employed a set of highly engineered features, which is exactly what this study wants to avoid.

2.3 Tree Kernels Methods

From a very general perspective, a classification problem can be considered as an attempt to learn a function f able to map in the best way an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, where the former is the initial vector space and the latter

is the set of target labels. While in many cases the input space is composed of simple features such as Bag-of-Words or n-grams occurrences, sometimes highly engineered (and costly) features are needed, especially when dealing with complex classification problems like those typically encountered in the AM pipeline. TK methods can solve the problem of costly engineered features, embedding in the input space $\mathcal{X}$ more complex structural information (e.g., graphs, trees) without creating *ad-hoc* features. In other words, sentences can be converted into tree representations and their similarity can be calculated by considering the number of common substructures (*fragments*).

Kernel machines classifiers, such as support-vector machine (SVM), have been widely used in classification problems. A kernel can be considered as a *similarity measure* that is able to map the inputs of an original vector space $\mathcal{X}$ into a high-dimensional feature space $\mathcal{V}$ *implicitly*, which is to say without the need to calculate the coordinates of data in the new space. More specifically, a kernel $k(x, x')$ (where x and x' belong to the input space $\mathcal{X}$ and represent the labelled and unlabelled input respectively) can be represented as an inner product in a high-dimensional space $\mathcal{V}$. In this regard, the kernel can be considered as a mapping $\varphi : \mathcal{X} \rightarrow \mathcal{V}$ where φ is an implicit mapping. The kernel function can be thus represented as:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{V}} \tag{2.1}$$

Where $\langle ., . \rangle_{\mathcal{V}}$ must necessarily be an inner product.

Given a training dataset of n examples $\{(x_i, y_i)\}_{i=1}^{n}$, where $i \in \{c_1, c_2\}$ with c_1 and c_2 being the specific classes of a binary classification, the final classifier $\hat{y} \in \{c_1, c_2\}$ can be calculated using the above-mentioned kernel function in the following way:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i k(x_i, x') \tag{2.2}$$

Or:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i \varphi(x).\varphi(x') \tag{2.3}$$

Where w_i are the weights learned by the trained algorithm.

A TK can be considered a *similarity measure* able to evaluate the differences between two trees. Before selecting the appropriate TK function, two important steps should be considered: choosing the type of tree representation and the type of fragments. In this work, sentences have been converted into Grammatical Relation Centered Tree (GRCT) representations, which involves PoS-Tag units and lexical terms. While their structures have been divided into Partial Trees (PTs) fragments (57), where each node is composed of any possible sub-tree, partial or not, providing a higher generalization. A description of various kind of tree representations can be found in Croce et al. (13), while a brief description of tree fragments can be found in Nguyen et al. (60) and Moschitti (57).

DS1	n.
Expert/testimony	372
Study/statistics	281
Total	653

DS2	n.
Expert/testimony	311
Study/statistics	258
Total	569

Table 2.1: Number of sentences in the two datasets, grouped by category group.

In this case, the PTK can be expressed using the following equation (57):

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \tag{2.4}$$

Where T_1 and T_2 are the two trees whose similarity should be evaluated, N_{T_1} and N_{T_2} are their respective sets of nodes, and $\Delta(n_1, n_2)$ represents the number of common fragments in n_1 and n_2 (57).

2.4 The Use Case

Two important Argument Mining datasets have been considered, and they will be referred to as DS1 and DS2. The first one is taken from Al Khatibet al. (4), while DS2 is from Aharoni et al. (1). This work is "downstream" from these two previous works which interestingly contains arguments taken from several topics, facilitating the creation of a context-independent classifier.

Although these two datasets have been built for different tasks, they share a very similar labelling system. The two datasets, in fact, classify argumentative text depending on three common labels (i.e. Study/Statistics, Expert/Testimony, Anecdote). In this study, only the first two groups have been considered suitable for the final purpose of detecting evidence typology. The idea is to train a classifier to automatically recognize when a text is an evidence coming from *studies/statistics* and when it comes from an expert *opinion/testimony*.

Since the two datasets have been created for other purposes, there is a further layer of complexity. For example, DS1 was composed of very segmented data, and it was necessary to recompose segmented sentences. Moreover, even though the two datasets share a similar labelling system when referring to some evidence typology (especially anecdote, study/statistics and expert/testimony), they could assume a slightly different idea of what these labels actually describe. In spite of these problems, their combination can be a powerful set of data for our aims, and the results of this experiment seem to confirm this assumption.

As can be seen from Table 2.1, a total of 653 sentences have been extracted from DS1 (372 belonging to the group "expert/testimony" and 281 belonging to the group "study/statistics"). While 569 sentences have been extracted from DS2 (311 for the "expert/testimony" group, 258 for the "study/statistics" group).

After having extracted the sentences from DS1 and DS2, a Grammatical Relation Centered Tree (GRCT) representation was created for each sentence of the two datasets. Furthermore, a TFIDF vectorialization has been applied to each dataset.

In other words, the sentences of the two datasets were converted into two kinds of "representation", with each labelled example having both a Grammatical Relation Centered Tree and a vector of TFIDF BoW, representing the features of the sentence.

For example, the sentence: *"Lucretius believed the world was composed of matter and void"* taken from DS2, can be represented as the GCRT in the Figure 3.1 and can have the following TFIDF vectorial representation:

```
the:0.0924 and:0.1237 of:0.1193
was:0.1095 believed:0.2526
world:0.1537 matter:0.2092
void:0.3157 composed:0.3020
```

The final classification algorithm was trained on these two kinds of representations by using KeLP (19). Since KeLP allows to combine multiple kernel functions, the classification algorithm was built as a combination of a Linear Kernel and a Smoothed Partial Tree Kernel (SPTK) (14), with the first kernel related to the TFIDF vectors and the second kernel related to the GRCT representations. More details on kernel combinations can be found in Shawe-Taylor and Cristianini (75). However, to evaluate the contribution of TKs, the experiment was also performed by using just one of the two representations (SPTK or TFIDF).

More precisely, two groups of classifiers were trained following two different strategies. The classifiers of the first group were trained on the 653 instances of DS1, dividing it into two subsets of 458 and 195 instances, for training and test. The second group of classifiers was trained on the 569 instances of DS2, dividing it into two subsets of 399 and 170 sentences, for training and test. After having been trained and tested on its given dataset, each classifier has also been tested on the other dataset (DS2 for the first group, DS1 for the second group). In this way, the ability of classifiers to generalize can be evaluated.

Since each group has three classifiers (TFIDF, SPTK, and the combination SPTK+TFIDF), a total of six classifiers has been evaluated.

2.5 Results

The results can be seen in Table 2.2. To evaluate the performance of the two groups of classifiers, a simple "Majority" baseline was created. Interestingly, all classifiers outperformed the baseline in all metrics.

Overall, TKs (SPTKs, in this case) outperformed simple TFIDF in three cases out of four (the TFIDF of the first classifier is the only exception). It

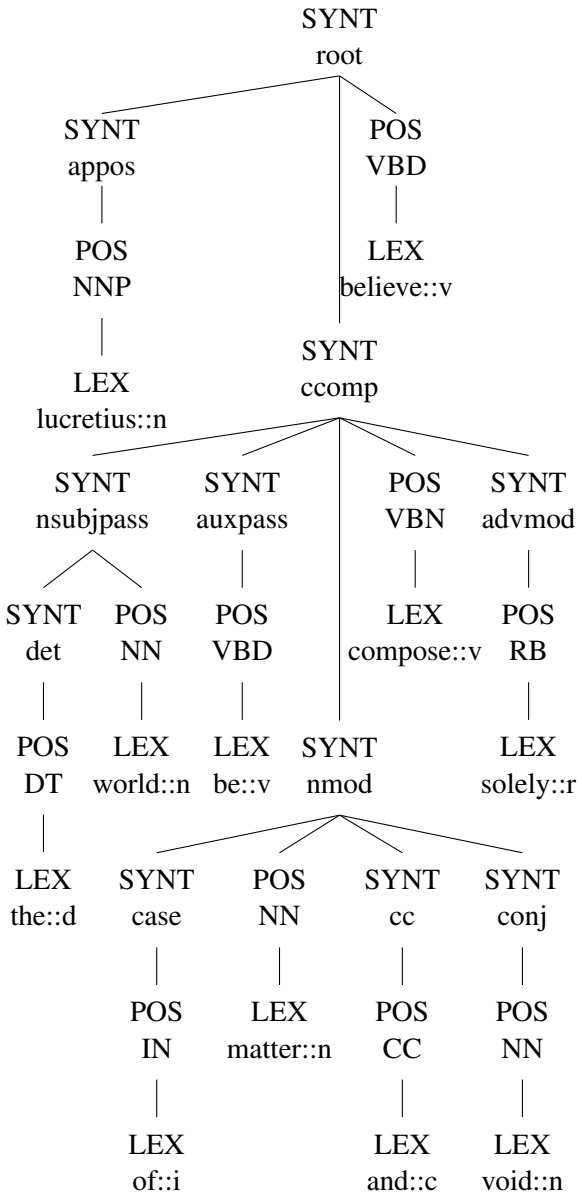


Figure 2.1: The GCRT representation for the sentence *"Lucretius believed the world was composed solely of matter and void"*

BASELINE	DS1			DS2		
	P	R	F1	P	R	F1
Averages (macro)	0.28	0.50	**0.36**	0.27	0.50	**0.35**

GROUP 1	Performance on DS1								
	TFIDF			**SPTK**			**SPTK+TFIDF**		
	P	R	F1	P	R	F1	P	R	F1
Study	0.93	0.87	0.90	0.88	0.83	0.85	0.90	0.92	0.91
Expert	0.89	0.94	0.92	0.85	0.90	0.88	0.93	0.91	0.92
Average F1 (macro)		**0.91**			**0.87**			**0.92**	
	Performance on DS2								
Study	0.80	0.55	0.65	0.77	0.67	0.71	0.78	0.66	0.72
Expert	0.70	0.88	0.78	0.75	0.83	0.79	0.75	0.85	0.80
Average F1 (macro)		**0.72**			**0.75**			**0.76**	

GROUP 2	Performance on DS1								
	TFIDF			**SPTK**			**SPTK+TFIDF**		
	P	R	F1	P	R	F1	P	R	F1
Study	0.84	0.54	0.66	0.81	0.78	0.80	0.82	0.80	0.81
Expert	0.73	0.92	0.81	0.84	0.86	0.85	0.85	0.87	0.86
Average F1 (macro)		**0.74**			**0.82**			**0.84**	
	Performance on DS2								
Study	0.70	0.67	0.68	0.76	0.64	0.69	0.69	0.69	0.69
Expert	0.73	0.76	0.74	0.73	0.83	0.78	0.74	0.74	0.74
Average F1 (macro)		**0.71**			**0.73**			**0.72**	

Table 2.2: Results of the majority baseline and two groups of classifiers, reporting precision (P), recall (R) and F1.

means that TKs can not only reach the performances of traditional features such as TFIDF, but also outperform them. Noticeably, the combination of TK and TFIDF has always performed better than simple TFIDF, which means that combining TKs and traditional features is a valid strategy to improve performances.

The classifiers of the first group had a good performance not only on the dataset they were trained on (DS1), but also on DS2. Noticeably, also the classifiers of the second group performed better on DS1.

2.6 Conclusion

The aim of this work is to show that it is possible to perform a fine-grain discrimination between different kinds of argumentative evidence by using TKs, without the need of using sophisticated feature vectors. The achieved classifier exploited the ability of Tree Kernels to calculate similarities between tree-structured sentences, considering the similarity of their fragments.

The experiment was performed on two famous Argument Mining datasets, which share a similar labelling system (they were referred to as DS1 and

DS2). More specifically, two groups of classifiers were trained combining a SPTK related to the GCRT representations and a linear kernel related to the TFIDF-BoW vector representations. The first group of classifiers was trained on DS1, while the second was trained on DS2.

A possible improvement to this approach could be achieved by adding also *n*-grams to assess if they can offer a better representation of sentences. Moreover, it would be interesting to compare results from different kinds of tree representation to assess whether GRCTs are the best choice for this particular task.

One of the achievements of this study is the successful combination of two important datasets originally designed for other purposes.

Also, it is worth remarking that this study is context-independent and focused on the structures of argumentative evidences without considering the specific context in which arguments are placed.

Finally, the main achievement of this work is to show that TKs can differentiate between different kinds of supporting evidences with high performances, which can facilitate the discrimination among different Argument Schemes (e.g. Argument from Expert Opinion), a crucial sub-task in the Argumentation Mining pipeline.

Chapter 3

Detecting Argumentative Opposition

Original title: **Classifying Argumentative Stances of Opposition Using Tree Kernels**

Abstract

The approach proposed in this study aims to classify argumentative oppositions. A major assumption of this work is that discriminating among different argumentative stances of support and opposition can facilitate the detection of Argument Schemes. While using Tree Kernels for classification problems can be useful in many Argument Mining sub-tasks, this work focuses on the classification of opposition stances. We show that Tree Kernels can be successfully used (alone or in combination with traditional textual vectorizations) to discriminate between different stances of opposition without requiring highly engineered features. Moreover, this study compare the results of Tree Kernels classifiers with the results of classifiers which use traditional features such as TFIDF and *n*-grams. This comparison shows that Tree Kernel classifiers can outperform TFIDF and *n*-grams classifiers.

3.1 Introduction

Publicly open reviews on bills are used in many legal systems. In fact, in some systems, it is mandatory to open public reviews during the legislative process to encourage people's participation and engagement.

Interestingly, web portals for collecting opinions and comments from citizens are becoming more and more frequent, and the idea of supporting people participation and engagement has been embraced by many famous social media.

However, there are still important obstacles when trying to understand the argumentative threads of online debates, since they are often presented as a flat flow of textual interactions.

In this regard, it is extremely difficult for decision makers to extract useful information from debates with hundreds of posts. Similarly, it is difficult to extract a useful map of pros and cons from a given online debate.

In this sense, one of the most ambitious aims for the future of artificial intelligence is to automatically recognize arguments and counter-arguments in debates, along with argumentative fallacies.

This work presents a method which uses Tree Kernels to classify argumentative stances of opposition facilitating the detection of Argument Schemes such as the well-known "Slippery Slope" argument that produces polarization and emphasizes debates.

3.2 Methodology

3.2.1 The Argument Mining Pipeline

The main target of Argument Mining (AM) is to analyze arguments, including their components and the relations connecting these components (9; 47). With an increasing number of works written and the interest of important private actors like IBM, this field has attracted a growing attention in the last few years (9), achieving important results and applications. These applications have been successfully implemented in a wide range of domains, since AM is physiologically multidisciplinary and facilitates cooperation among fields (e.g. Information Extraction, Knowledge Representation, Legal Reasoning, Sentiment Analysis). Importantly, being language the main means by which humans express their arguments, there is a close relation between AM and Natural Language Processing. Also, AM is closely related to Opinion Mining, with the latter trying to detect *what* people say and the former trying to understand *why* (25).

Lippi and Torroni (47) describe AM as a pipeline composed of three steps: the first step is the identification of argumentative data (which must be distinguished from non-argumentative data); the second step is the detection of the boundaries of argumentative components; the third step consist of predicting the relations among argumentative units and among arguments.

Importantly, the last two steps strictly depend on the underlying argumentative model, e.g. the most frequently used two-role model proposed by Walton (86) (which considers argumentative units as "claim" and "premise"), or the more complex five-role model proposed by Toulmin (81) (which considers fact, warrant, backing, rebuttal and qualified claim).

Cabrio and Villata (9) proposed a simpler two-step pipeline, which is the one that we will refer to in this work. In their pipeline, the first step is the identification of arguments, which involves not only the differentiation between argumentative and non-argumentative data but also the identification of the roles of argumentative components (claims, premises, etc.) and their boundaries. The second step involves the prediction of the heterogeneous nature of argument relations (e.g., *supports*, *attacks*) and the connection between premises/evidences and conclusions/claims.

Ideally, after the above mentioned steps, a last phase can be that of fitting the map of argumentative components into an Argument Scheme (e.g., argument from Analogy, "Slippery Slope" argument, argument from Example).

From the one side, this work tries to classify argumentative stances of opposition. On the other side, it tries to facilitate the detection of those argumentative stances whose classification is more likely to be related to a specific Argument Scheme. Particularly, we targeted the well-known "Slippery slope" argument and we evaluated the ability of Tree Kernel methods to distinguish this scheme from other kinds of opposition stance.

3.2.2 Tree Kernels Methods

A kernel function can be considered as a *similarity measure* that perform an implicit mapping $\varphi : \mathcal{X} \rightarrow \mathcal{V}$ where $\mathcal{X}$ is a input vector space and $\mathcal{V}$ is a high-dimensional space. The function can be represented as follows:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{V}} \tag{3.1}$$

Importantly, the above $\langle ., . \rangle_{\mathcal{V}}$ must necessarily be considered an inner product, while x and x' belong to $\mathcal{X}$ and represent the labelled and unlabelled input respectively.

If we consider a binary classification task with a training dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ composed of n examples, where $y \in \{c_1, c_2\}$ (with c_1 and c_2 being the two possible outputs of a binary classification), the final classifier $\hat{y} \in \{c_1, c_2\}$ can be calculated in the following way:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i k(x_i, x') = \sum_{i=1}^{n} w_i y_i \varphi(x).\varphi(x') \tag{3.2}$$

Where the weights w_i are learned by the trained algorithm.

Since Tree Kernels belong to the family of kernel methods, they can be considered a *similarity measure* too. In particular, they are designed to calculate similarities between tree-structured documents.

Importantly, there are different kinds of Tree Kernel functions, which operate on different segments of the tree-structured documents. In fact, different TK functions make calculations by watching at different substructures of the given tree-structured data. In this study, data was segmented into Partial Trees (PTs) fragments, where each node is composed of any possible sub-tree, partial or not. The reason for this choice is that PTs are able to provide a high generalization (57).

The resulting function, called Partial Tree Kernel (PTK), can be calculated as follows (57):

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \tag{3.3}$$

In the above equation, T_1 and T_2 are the two trees involved in the calculation of the similarity, while N_{T_1} and N_{T_2} are their respective sets of nodes and $\Delta(n_1, n_2)$ is the number of common fragments in node n_1 and node n_2. More information about tree fragments can be found in Nguyen et al. (60) and Moschitti (57).

Another important aspect is the selection of the kind of tree structure that will represent the textual data. A description of different kinds of tree representation is presented in Croce et al. (13). In this work, we represented text as Grammatical Relation Centered Trees (GRCTs), which take into account not only Part-of-Speach Tags but also lexical terms.

As we will try to show in this work, there are different reasons for using Tree Kernels. On the one side, it seems that they can outperform traditional features, on the other side they can keep a high degree of generalization leveraging structural information.

3.3 Related Works

This work presents an approach which uses Tree Kernels methods to classify opposition stances. This method is also presented as a way to facilitate the detection of Argument Schemes, one of the most complex sub-task in AM.

Noticeably, only few studies tried to detect Argument Schemes. In this regard, Feng and Hirst (17) managed to achieve an accuracy ranging from 63 to 91% in one-against-others classification and 80-94% in pairwise classification. Some years later, Lawrence and Reed (33) increased the previous performances, achieving F-scores ranging from 0.78 to 0.91. However, since the task of detecting Argument Schemes is very complex, the two above-mentioned works deployed a set of highly engineered features. The aim of this study is to give a further contribution in this part of the AM pipeline, showing a possible method for Argument Scheme discrimination which is also able to preserve high levels of generalization without requiring highly engineered features.

TKs have already been used successfully in question answering (18), metaphor identification (27), semantic role labelling (58) and other NLP-

related task. Although results of TKs methods have been strongly encouraging in all the above mentioned tasks, showing the ability of TKs to perform well, their application in AM has been relatively limited. One of the first uses in AM was proposed by Rooney et al. (72), who combined the use of TKs with Part-of-Speech tags. However, it was only after three years that somebody underlined the potential advantages of deploying TKs in the argument detection sub-task (which is the first step of the AM pipeline) (47). One year later, the same authors presented a web application which uses TKs and is capable of extracting arguments from text automatically (48). Still today, this is one of the few existing attempts to create a complete argument extraction tool.

This approach is the continuation of two previous works ((36) (39)), which aimed at discriminating among different kinds of argumentative stances of support and opposition. These two works are an attempt to give a contribution to the AM pipeline finding a working methodology capable to discriminate among stances of support/opposition by using Tree Kernels. The underlining assumption is that being able to classify different kinds of support and opposition is a crucial aspect also in the discrimination of different Argument Schemes.

In the present paper, our previously achieved findings will be extended, with a deeper analysis which involves the performances of twenty classifiers.

3.4 Corpus and Annotation

The analyzed corpus is composed of a group of 638 annotated sentences gathered from public available data. The annotation process is still ongoing under the supervision of experts of domain and our aim is to further extend the amount of annotated sentences. Particularly, these sentences have been extracted from the opinion of voters in the "Opinion Poll" section available in the official website of Nevada Legislature. More specifically, these sentences are taken from the opinion against the Senate Bill 165(SB165), about Euthanasia. Each comment of opposition against SB165 has been segmented into sentences using an automatic sentence segmentation tool.

After a preliminary empirical analysis, each sentence of the corpus has been manually annotated following an annotation scheme which is designed to allow different degrees of granularity in the classification process. Table 3.1 shows the list of classes with some examples, while Table 3.2 describes how these classes have been grouped in super-classes to create different levels of granularity. Thanks to this flexible annotation, the ability of TKs to perform fine-grained differentiation at each level of granularity has been tested.

So far, the classes PERSONAL EXPERIENCE, NOT PERSONAL EXPERIENCE, JUDGEMENTS SIMPLE and JUDGEMENT MORAL have not been used, but they could be useful when the annotation process will be completed and the corpus will be expanded.

As can be seen in Table 3.2, the first level is the least granular, since it discriminates between just two categories: SLIPPERY SLOPE sentences and

Table 3.1: The annotation classes with some examples.

Classes	Examples
SLIPPERY SLOPE	*- This would turn physicians into legal murderers.*
JUDGEMENT SIMPLE	*- This bill is terrible.*
JUDGEMENT MORAL	*- This bill is an affront to human dignity.*
MORAL ASSUMPTIONS	*- Only God should decide when a person is supposed to die.* *- Being a Christian, I cannot accept this bill.* *- This is totally against the Hippocratic Oath!*
STUDY STATISTICS	*- Our country already experienced 20% increase of suicide rate.*
ANECDOTAL *(PERSONAL EXPERIENCE)* *(NOT PERSONAL EXPERIENCE)*	*- The bible says that this is wrong.* *- My husband struggled a lot of years and* [...] *- In Oregon this bill created the chaos.*
OTHER/NONE	All the sentences that does not belong to the above classes

Table 3.2: The granularity levels and the grouping options.

Granularity 1	Granularity 2	Granularity 3	Granularity 4
SLIPPERY SLOPE	SLIPPERY SLOPE	SLIPPERY SLOPE	SLIPPERY SLOPE
OTHER/NONE	TESTIMONY	TESTIMONY	ANECDOTAL
			STUDY STATISTICS
	OTHER/NONE	JUDGEMENTS MORAL	JUDGEMENTS (simple + moral)
			MORAL ASSUMPTIONS
		OTHER/NONE	OTHER/NONE

the rest of classes. The second level is more granular, since it discriminates among three categories: SLIPPERY SLOPE, TESTIMONY, OTHER/NONE. The third level also involves JUDGEMENTS MORAL. The fourth level is the most granular since it discriminates among six categories: SLIPPERY SLOPE, ANECDOTAL, STUDY STATISTICS, JUDGEMENTS, MORAL ASSUMPTIONS, OTHER/NONE.

Importantly, during the annotation process we aimed to find out how people justify their opposition stances in a monologic debating context. In other words, the selected classes are the product of our empirical analysis on how people express their opposition. Since the focus of this annotation is *why* people are expressing a stance of opposition, all those comments which do not give any explanation for the opposition stance have been considered as part of the class OTHER/NONE (e.g. exhortations like “Please, vote no!”).

The number of sentences grouped by class is listed in Table 3.3.

3.5 The Experiment

For each sentence of the corpus a Grammatical Relation Centered Tree (GRCT) representation was created along with a TFIDF vectorization. More precisely, we attempted three different TFIDF vectorizations considering monograms, 2-grams and 3-grams, in order to assess the effects of *n*-grams on

Table 3.3: Number of sentences depending on class and granularity.

Classes	Gr.4	Gr.3	Gr.2	Gr.1
SLIPPERY SLOPE	82			
STUDY STATISTICS	26	133	423	556
ANECDOTAL *(PERSONAL EXPERIENCE)* *(NOT PERSONAL EXPERIENCE)*	107			
JUDGEMENT SIMPLE JUDGEMENT MORAL	54	140		
MORAL ASSUMPTIONS	86			
OTHER/NONE	283			

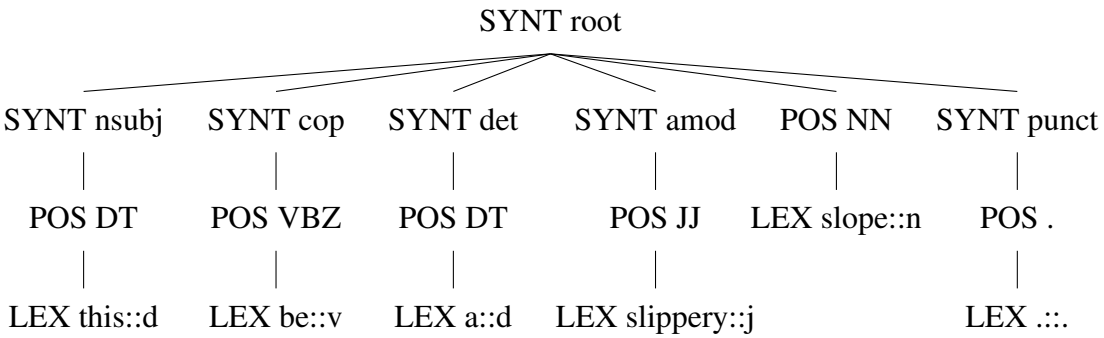


Figure 3.1: The GCRT representation for the sentence *"This is a slippery slope."*

the results.

In other words, each labelled document in the corpus has two typology of "representation": GRCT and TFIDF. For example, the sentence *"This is a slippery slope."* can be represented as the GCRT in Figure 3.1 and can have a TFIDF vectorial representation like one of those in Figure 3.2, which shows the monogram, 2-grams and 3-grams. This results in three combinations at each level of granularity and, thus, in a total of twelve possible combinations.

All these classifiers were trained on the GRCT and TFIDF representations by using KeLP (19). This operation was performed by dividing the corpus of 638 sentences into a test set of 191 sentences and a training set of 446 sentences and by using a One-vs-All classification, which is one of the most common approach for multi-class problems. Noticeably, KeLP allows to combine multiple kernel functions. In this work, the classification algorithm was built as a combination of a Linear Kernel and a Partial Tree Kernel (PTK) (57), with the first kernel related to the TFIDF vectors and the second kernel related to the GRCT representations. More details on kernel combinations can be found in Shawe-Taylor and Cristianini (75).

An important contribution of this study is that other two classifiers have been added at each level of granularity to better understand the real contribution of TKs and TFIDF. The first considers just monograms, the second

Figure 3.2: An example of TFIDF representation of a sentence (monograms and *n*-grams).

Monograms:
`is:0.2872 this:0.2944 slippery:0.6445 slope:0.6445`
2-grams:
`is:0.1913 this:0.1961 this_is:0.3442 slippery:0.4293 slope:0.4293 is_slippery:0.4962 slippery_slope:0.4374`
3-grams:
`is:0.1543 this:0.1581 this_is:0.2776 slippery:0.3462 slope:0.3462 is_slippery:0.4002 slippery_slope:0.3528 this_is_slippery:0.4351 is_slippery_slope:0.4002`

considers just TK (monograms were preferred to other *n*-grams simply because of their better performances).

3.6 Results

Table 3.4 shows the resulting scores for each classifier, grouped by granularity. Since we want to show the non-triviality of the proposed task, we added the performance of a stratified baseline. The stratified method has been chosen because it produces better results, compared to a majority baseline, at all levels of granularity and because it reflects the classes' distribution in the training set.

As expected, results show that F1 scores are lower at higher granularity. Importantly, we remark that classifiers 2 and 3 are probably the best ones in terms of balance among results and number of instances per class. Moreover, since the class OTHER is often responsible for the increase of the mean F1 value, it is important to consider not just the Mean F1 score. For example, Figure 3.3 shows the results for the mean F1 scores and the results for the F1 scores related to different classes (and sub-classes), particularly SLIPPERY SLOPE, which is the main target of this study. In this way it is possible to have a better understanding of the performance of the classifiers. The same Figure show the decrease of the mean F1 scores at higher granularity.

Finally, results show that TK-only classifiers can be equal or better than monograms (at granularity 2 and 3, respectively). Although monograms outperforms TK-only classifiers at granularity 1 and 4, the combination TK+*n*-grams is always the most performing.

The results for each level of granularity will be now discussed.

3.6.1 Results for Granularity 1

The classifiers at granularity level 1 show that the best performance can be achieved by combining the GRCT Tree Kernel with monograms. In fact, although the monograms classifier outperforms the TK-only classifier, with a mean F1 score of 0.79 and 0.76 respectively, the combination of TK and

monograms outperform all the other combinations, achieving a mean F1 score of 0.81.

However, the problem of these classifiers is that the number of SLIPPERY SLOPE instances (82) is too little compared to the instances of OTHER (556). In fact, the good result is mostly due to the F1 score related to the class OTHER (0.96), while the F1 score related to the SLIPPERY SLOPE class, our main target, is at 0.67.

3.6.2 Results for Granularity 2

The classifiers with granularity level 2 achieved more encouraging results. Interestingly, the results of the TK-only classifier and the monograms classifier are equal in this case, with a mean F1 score of 0.76. Again, the best results is achieved by combining TK and monograms, with a mean F1 score 0.77. In this case, the F1 score for SLIPPERY SLOPE reaches 0.70, while the score for TESTIMONY is 0.71. Even though the instances of OTHER are still too many compared to the number of instances of the other two classes, the numbers of instances is more balanced compared to granularity 1.

3.6.3 Results for Granularity 3

The granularity level 3 is maybe the one with the best balance in terms of number of instances. The classifiers of this group achieved a mean F1 score ranging from 0.65 to 0.69. Interestingly, the TK-only classifier outperformed the monograms classifier, achieving the best performance together with the TK+monograms combination. This means that TK can outperform traditional TFIDF representations.

However, for the purposes of this work, the TK+monograms combination is still preferred, since it produce a better performance on the SLIPPERY SLOPE and TESTIMONY classes.

3.6.4 Results for Granularity 4

The last group of classifiers is the most granular one. The main problem of this classifiers is that they were trained on a small number of instances per class, especially the classes STUDY STATISTICS and JUDGEMENTS, which have just 26 and 54 instances respectively. On the other side, an important achievement of this group of classifiers is that they produce an F1 score for the class SLIPPERY SLOPE which is comparable or superior to granularity 3, achieving good results also with the class ANECDOTAL.

3.7 Conclusions and Future Work

This study shows that Tree Kenels can outperform traditional features such as TFIDF. Importantly, we wanted to remark that one of the main advantages

of Tree Kernels is the possibility of leveraging structural information while preserving a high generalization.

The proposed method shows the ability of Tree Kernels to classify different kinds of opposition stance with relatively good results and without using highly engineered features, while at the same time presenting a working methodology for Argument Scheme discrimination.

The experiment was performed on a corpus of 638 short comments expressing opposition against the Nevada's Senate Bill 165, which aims to regulate Euthanasia .

Although results are encouraging, especially with the second and third groups of classifiers, there are still some obstacles when trying to deepen the degree of granularity. In the future, creating a chain of classifiers could help solve this problem, with a gradual and more complex advancement into granularity.

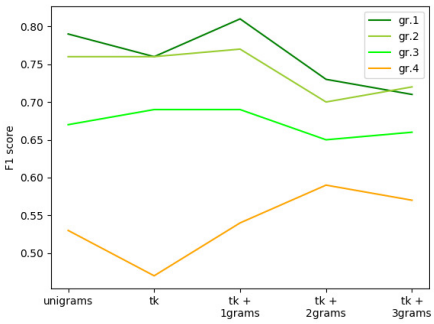
Moreover, we are working on the enlargement of the annotated data, since the imbalanced number of instances per class is a significant obstacle towards the achievement of better scores.

Despite the above-mentioned limitations, the present work shows that TKs can differentiate between argumentative stances and recognize stances that are related to the "Slippery Slope" argument. Still, the combination TK+n-grams outperforms the other classifiers.

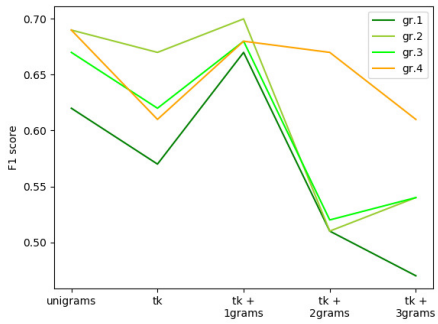
Table 3.4: The scores of the classifiers grouped by granularity (P = Precision, R = Recall, F1 = F1 score). Close to the class name, the number of instances is specified. SS = SLIPPERY SLOPE, O = OTHER, T = TESTIMONY, JM = JUDGEMENTS AND MORAL, ST = STUDY STATISTICS, A = ANECDOTAL, MA = MORAL ASSUMPTIONS, J = JUDGEMENTS.

Classes	1grams			TK			TK + 1grams			TK + 2grams			TK + 3grams			Stratified baseline
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Granularity 1																
SS (82)	0.77	0.52	0.62	0.71	0.48	0.57	0.75	0.60	0.67	0.90	0.36	0.51	0.89	0.32	0.47	
O (556)	0.93	0.98	0.95	0.93	0.97	0.95	0.94	0.97	0.96	0.91	0.99	0.95	0.91	0.99	0.95	
Mean F1		0.79			0.76		—>	**0.81**	<—		0.73			0.71		0.54
Granularity 2																
SS (82)	0.71	0.68	0.69	0.82	0.56	0.67	0.76	0.64	0.70	0.71	0.40	0.51	0.83	0.40	0.54	
T (133)	0.70	0.68	0.69	0.68	0.74	0.70	0.66	0.77	0.71	0.63	0.74	0.68	0.68	0.74	0.70	
O (423)	0.89	0.90	0.89	0.90	0.93	0.91	0.92	0.91	0.91	0.88	0.92	0.90	0.88	0.95	0.91	
Mean F1		0.76			0.76		—>	**0.77**	<—		0.70			0.72		0.31
Granularity 3																
SS (82)	0.70	0.64	0.67	0.76	0.52	0.62	0.73	0.64	0.68	0.65	0.44	0.52	0.69	0.40	0.54	
T (133)	0.68	0.79	0.73	0.61	0.82	0.70	0.63	0.85	0.72	0.60	0.82	0.69	0.60	0.82	0.69	
JM (140)	0.62	0.49	0.55	0.76	0.53	0.62	0.64	0.53	0.58	0.71	0.53	0.61	0.74	0.55	0.63	
O (283)	0.70	0.75	0.73	0.76	0.85	0.80	0.76	0.75	0.76	0.76	0.82	0.79	0.75	0.82	0.79	
Mean F1		0.67		—>	**0.69**	<—	—>	**0.69**	<—		0.65			0.66		0.20
Granularity 4																
SS (82)	0.67	0.72	0.69	0.67	0.56	0.61	0.64	0.72	0.68	0.65	0.68	0.67	0.63	0.61	0.61	
A (107)	0.56	0.77	0.65	0.54	0.85	0.66	0.55	0.88	0.68	0.59	0.88	0.71	0.55	0.85	0.67	
ST (26)	1.00	0.13	0.22	0.33	0.13	0.18	1.00	0.13	0.22	1.00	0.13	0.22	1.00	0.13	0.22	
J (54)	0.78	0.37	0.50	0.33	0.11	0.16	0.78	0.37	0.50	0.88	0.37	0.52	1.00	0.37	0.54	
MA (86)	0.50	0.36	0.42	0.63	0.36	0.45	0.52	0.43	0.47	0.68	0.54	0.60	0.70	0.50	0.58	
O (283)	0.66	0.76	0.71	0.70	0.86	0.77	0.76	0.76	0.76	0.76	0.85	0.80	0.74	0.86	0.79	
Mean F1		0.53			0.47			0.54		—>	**0.59**	<—		0.57		0.21

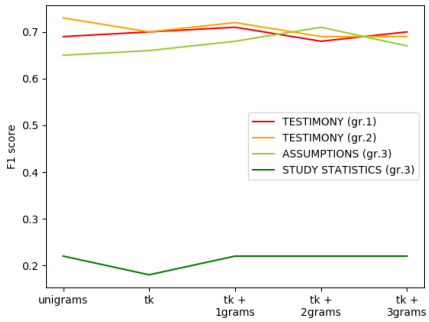
Figure 3.3: A comparison between classes' scores and Mean F1 scores (a,b,c,d,e) and the decrease of F1 over granularity (f).



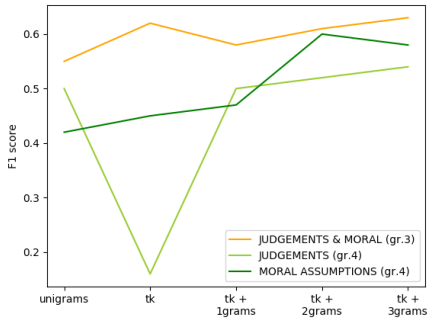
(a) Mean F1 scores for the 4 granularities



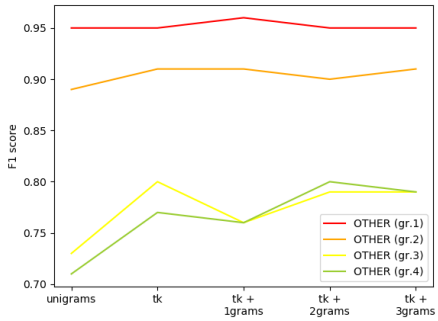
(b) F1 scores for SLIPPERY SLOPE



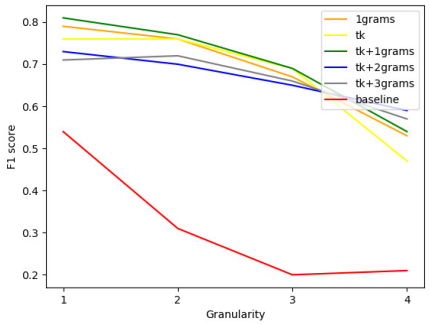
(c) TESTIMONY and its sub-classes



(d) JUDGEMENTS AND M. and its sub-classes



(e) Scores for the class OTHER



(f) Mean F1 scores decrease

Chapter 4

Combining Tree Kernels and Tree Representations

Original title: **Combining Tree Kernels and Tree Representations to Classify Argumentative Stances**

Abstract

This work investigates how the combination of different tree representations with different Tree Kernel functions influences the results of the classifications in two specific case studies. One case study is related to the classification of argumentative stances of support, the other one is related to the classification of stances of opposition. Results show that some Tree Kernels achieves not only higher results but also a higher level of generalization. Moreover, it seems that also the kind of tree representation influences the performances of classifiers. In this study, we thus explore this relation between tree representation and different Tree Kernels, considering also compositional trees.

4.1 Introduction

This study is related to the field of Argument Mining (AM), a relatively new research field focused on the analysis, detection and classification of argumentative structures, substructures and units from natural argumentation. On the one side, AM is a field which employs Natural Language Processing techniques. On the other side, AM is also connected to the field of Argumentation, which involves a wide range of logical and philosophical aspects. The aspects related to natural language and those related to Argumentation are both crucial when dealing with legal data, because legal texts contain several argumentative structures which are encoded in natural language, e.g. inferential logical-ontological rules or even stereotypical patterns of inference (known as Argumentation Schemes). These rules and argumentative patterns are composed of premises (evidences) and conclusions (claims) which can be classified using Machine Learning algorithms, facilitating the automatic detection of argumentative structures and rules in legal texts.

Due to the complexity of human language, AM scholars often need to create classifiers employing highly-engineered features capable of describing the complexity of argumentative structures in natural language. As suggested in Lippi and Torroni 2015 (46), the problem of having to engineer features to catch complex structures can be solved by employing classifiers that works directly on structures, such as Tree Kernel classifiers. Following this suggestion, recent studies have shown how Tree Kernels classifiers can discriminate between different kinds of argumentative stances of support (36) and opposition (39).

For example, Liga 2019 (36) described a first attempt to use Tree Kernel classifiers to discriminate argumentative stances of support in the context of a binary classification. On the other side, Liga and Palmirani 2019 (39) showed that a similar approach can be applied also to opposition stances and to a multi-class classification problem. The present work takes inspiration from these two studies and reproduces their settings by using different tree representations and tree kernel functions, to assess whether or not there is a specific combination that better suits the task of classifying/discriminating argumentative stances.

In Section 4.2, Tree Kernels classifiers will be introduced along with a description of different tree representations and Tree Kernel functions. In the Section 4.3, we will briefly describe the related works in the domain of AM, considering the studies which employed Tree Kernels in AM and describing the two above-mentioned studies, (36) and (39), from a general perspective. In Section 4.4, we will reproduce the settings of Liga 2019 (36) applying different tree representations and different Tree Kernel functions to the same scenario and analyzing the results of each classifier. In Section 4.5, we will do the same process, employing different tree representations and different Tree Kernel functions in the setting of the experiment in Liga and Palmirani 2019 (39). Lastly, in Section 4.6 we will open a short discussion and conclude the work.

4.2 Tree Kernels and tree representations

A Tree Kernel is simply a similarity measure. It works by comparing the similarity between tree-structured pieces of data. Supposing that we want to use Tree Kernel classifiers for textual data, there are two main elements to consider.

The first element is the kind of tree-structure to employ, namely the kind of tree representation we want to use. The second element is defining which fragments of the tree structures should be involved into the calculation of the similarity. The following sections briefly describe these two aspects.

4.2.1 Tree representations

The idea is that our data (e.g. textual data such as sentences) must be represented into a specific tree-structured shape to allow a Tree Kernel function to calculate the similarity between different pieces of tree-structured data. For example, a sentence can be converted into some kind of tree representation such as a dependency tree or constituency tree.

In the following part, we will shortly describe some of the most famous tree representations for the conversion of textual data into tree structures. They can be considered as particular kinds of Dependency Trees which combine grammatical functions, lexical elements and Part-of-Speech tags in different ways (13).

GRCT The Grammatical Relation Centered Tree (GRCT) representation is a very rich data representation (14). It involves grammatical, syntactical, lexical elements together with Part-of-Speech and lemmatized words. In this representation, after the root there are syntactical nodes (grammatical relations), then Part-of-Speech nodes and finally lexical nodes. In other words, a tree of this kind is balanced around the grammatical nodes, which determines the structure of dependencies.

LCT Also Lexical Centered Tree (LCT) representations involve grammatical, lexical and syntactical element, along with Part-of-Speech tags. However, the structure of the tree is different. In fact, it is "centered" over Lexical nodes, which are at the second level, immediately after the root. Part-of-Speech nodes and grammatical functions nodes are equally children of the lexical elements.

LOCT The Lexical Only Centered Tree (LOCT) representation contains just the lexical elements. Intuitively, the contribution of LOCT representation can be particularly determinant whenever the tasks to be achieved mostly depend on lexical elements.

cGRCT and cLCT The compositional Grammatical Relation Centered Tree (cGRCT) and the compositional Lexical Centered Tree (cLCT) representations are very similar to the the Grammatical Relation Centered Tree (GRCT) and the Lexical Centered Tree (LTC) representation. The difference here is that the representations allow compositional operators. This aspect will be explained more in depth in the section related to CSPTKs. In fact, cGRCTs and cLCT can be used with Compositionally-Smoothed Partial Tree Kernels (CSPTKs) which are designed specifically for the purpose of considering compositionality (5).

4.2.2 Tree Kernels

A kernel function can be considered as a *similarity measure* that perform an implicit mapping $\varphi : \mathcal{X} \rightarrow \mathcal{V}$ where $\mathcal{X}$ is a input vector space and $\mathcal{V}$ is a high-dimensional space. A general kernel function can be represented as follows:

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{V}} \tag{4.1}$$

Importantly, the $\langle ., . \rangle_{\mathcal{V}}$ in the above formula must necessarily be considered an inner product, while x and x' belong to $\mathcal{X}$ and represent the labelled and unlabelled input respectively. If we consider, for example, a binary classification task with a training dataset $\mathrm{D} = \{(x_i, y_i)\}_{i=1}^{n}$ composed of n examples, where $y \in \{c_1, c_2\}$ (with c_1 and c_2 being the two possible outputs of a binary classification), the final classifier $\hat{y} \in \{c_1, c_2\}$ can be calculated in the following way:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i k(x_i, x') = \sum_{i=1}^{n} w_i y_i \varphi(x).\varphi(x') \tag{4.2}$$

Where the weights w_i are learned by the trained algorithm.

When using Tree Kernels, the function must be adapted to allow the calculations over tree nodes. In this regards, a general Tree Kernel function can be calculated as follows (57):

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \tag{4.3}$$

In the above equation, T_1 and T_2 are the two trees involved in the calculation of the similarity, while N_{T_1} and N_{T_2} are their respective sets of nodes and $\Delta(n_1, n_2)$ is the number of common fragments in node n_1 and node n_2.

Importantly, $\Delta(n_1, n_2)$ can be seen as a function considering common fragments between trees. Depending on how this function is configured (i.e., which fragments are considered involved into the calculation of the similarity), different Tree Kernels can be obtained.

Given that our data is tree-structured, the second important element is the definition of the which fragments must be involved when calculating the similarity between trees. Defining which fragments to involve also means

defining the Tree Kernel function, because the names of the Tree Kernel functions usually derives from the fragment definition.

In the following part, some famous Tree Kernel functions will be shortly described; each of them defines, in a different way, which fragments should be involved into the calculation of the similarity.

STK In a SubTree Kernel (STK) (82), a fragment is any subtree, i.e. any node of the tree along with all its descendants.

SSTK A SubSetTree Kernel (SSTK) (12) considers as fragments the so-called subset-trees, i.e. it considers any node along with its partial descendancy. Since in SSTKs the only constraint of not breaking grammar production rules, and since fragments' leaves can be also non-terminal symbols, they can be considered a more general representation compared the previously mentioned STKs.

PTK A Partial Tree Kernel (57) is a convolution kernel that considers partial trees as fragments. Similarly to SSTKs, a partial tree is a fragment of a tree which considers a node and its partial descendancy. However, partial trees allow also partial grammar production rules. The fact that production rules can be broken (i.e. partial), makes PTs even more general than SSTs. This is the reason why PTKs should provide a higher ability to generalize.

SPTK A PTK can be also "Smoothed" PTK (SPTK) (14), which adds a further semantic layer into the calculation of node similarity. SPTKs allows to calculate similarities between dependency structures whose surfaces (i.e. the lexical nodes, or words) are partially or totally different. They introduce a lexical similarity which allows the generalization of tree structures through the semantic layer by representing words not just as mere symbols, but as semantic entities.

CSPTK The scenario can be further expanded considering Compositionally-Smoothed Partial Tree Kernels (CSPTK) (6), which apply a composition function between nodes to better represent contextual relations between words.

We have previously showed how compositional trees look like (i.e. cGRCTs and cLCTs), but we did not explain how compositionality works.

The key point of CSPTKs is that they can compute compositional trees integrating Distributional Compositional Semantics (DCS) operators into the kernel evaluation acting on both lexical leaves and non-terminal nodes. On the one side, SPTK already offer a modeling of lexical information, since they extend the similarity between tree structures allowing a smoothed function of node similarity, which makes them able to compare better trees which are semantically related even if their nodes and leaves differ.

However, SPTKs have an major limitation: they cannot consider compositional interactions between the lexical elements of the trees (i.e. between the words of the trees).

The meaning of the verb "to save" can be better captured only if we consider the verb in composition with the words it is refered to, namely "files" and "people". In this sense, CSPTK can better capture the role of more complex syntagmatic structures and compositions. Regarding the calculation of Δ, it is similar to SPTK, but the smoothing function is adapted according to (5; 6).

4.3 Related works

So far, only few studies have employed Tree Kernels in the field of AM. One of the first studies that mentioned the use of Tree Kernels in this field is Rooney 2012 (72), where kernels are used with Part-of-Speech tags and with sequences of words with the aim of detecting whether or not a sentence is related to an argumentative element (i.e. premise, conclusion, or both).

In 2015, Lippi and Torroni wrote an important study in which Tree Kernels are employed in a context-independent scenario, with the aim to detect argumentative claims (46). In this case, the authors employed PTKs using constituency trees as tree representations. Similar approaches have been applied also in specific domains like the legal domain (48; 50) and the medical domain (56), where SSTK have been applied over constituency trees.

Outside the field of AM, Croce et Al. (13) focused on the combination of different Tree Kernels and tree representations. However, no study has proposed yet a comparison of this kind in AM, particularly in the classification of argumentative stances of support and opposition.

As already stated, this work is based on two previous studies, Liga 2019 (36) and Liga and Palmirani 2019 (39). Interestingly, the two above-mentioned studies try to discriminate among stances of support and opposition that can be related to specific Argumentation Schemes (86). While the first paper focused on the stances of support potentially related to the Argumentation Scheme from Expert Opinion (which is a specific kind of source-based Argumentation Scheme (35)), the second study has a particular focus on the argumentative stances related to the Argumentation Schemes from Negative Consequences and the Slippery Slope argument (85). In Liga 2019, Tree Kernels have been used in combination with TFIDF vectors to automatically discriminate between different kinds of argumentative support, while in Liga and Palmirani 2019 a similar methodology was employed to detect argumentative stances of opposition.

Importantly, these two studies explore the ability of different kinds of Tree Kernel to perform the tasks of classifying argumentative stances (the first study employs PTK (57), while the second study employs SPTK (14)) However, both studies employ the same kind of tree representation: namely, they only employ Grammatical Relation Centered Tree (GRCT representations). However, as already stated, there are other kinds of tree representation that may be employed in the same settings. For this reason, the present study reproduce the same scenarios of these two works by employing different kinds of tree representation to assess whether or not a particular kind of tree representation

can be more suitable for the task of classifying argumentative stances of support and opposition.

To the best of our knowledge, no study of argumentative stance classification has so far presented a comparative analysis of the use of different kinds of tree representations and Tree Kernels. In particular, this study will compare the performance of the 5 tree representations described in Section 4.2.1 At the same time, we will combine these representations with the 5 Tree Kernels described in Section 4.2.2.

We will now describe the two settings of the present paper. The first one, related to the first study (36), will be defined Setting One; the second one, related to the second study (39), will be defined Setting Two. The experiments have been performed using the JAVA framework KeLP (19), and a 70/30 train-test split ratio.

4.4 Setting One

The first study combined two famous AM datasets in the same setting ((1) and (4), namely). We will refer to these datasets as DataOne (4) and DataTwo (1). The aim of the study was to develop a series of classifiers able to differentiate between argumentative support coming from the opinion of an expert and argumentative support coming from studies or statistics.

Importantly, in this study, the ability of Tree Kernels to generalize over different data was explored by training the classifiers on one dataset and testing them on both datasets (36). For this reason, we will consider Setting One as divided into two scenarios.

In the first scenario, all classifiers are trained on the training subset of DataOne (and tested not only on the testing subset of the same dataset, but also on the whole DataTwo dataset). In the second scenario, all classifiers are trained on the training subset of DataTwo (and tested not only on the testing subset of the same dataset, but also on the whole DataOne dataset).

As shown in the left side of Table 4.1, which reports the number of instances per class per dataset, the two datasets are quite balanced. The label expert has 372 and 311 instances in DataOne and DataTwo respectively; while the label Study/statistics has 281 and 258 instances in DataOne and DataTwo respectively.

4.4.1 Results for Setting One

The top part of Table 4.2 is referred to the first scenario, while bottom part is referred to the second scenario. For reason of space, compositional trees are reported jointly with their non-compositional counterpart: compositional tree representations (cGRCT and cLCT) should be thus considered related only with CSPTKs.

The experiments were performed using the configurations of the decay

Table 4.1: Number of sentences in the two datasets, grouped by category group (left). Configuration of the kernel parameters for Setting One (right).

DataOne	n.
Expert	372
Study/statistics	281
Total	653

DataTwo	n.
Expert	311
Study/statistics	258
Total	569

Kernels	Decay factors	
STK	$\lambda = .3$	
SSTK	$\lambda = .2$	
PTK	$\lambda = .4$	$\mu = .4$
SPTK	$\lambda = .4$	$\mu = .4$
CSPTK	$\lambda = .4$	$\mu = .4$

factors reported in the right side of Table 4.1[1].

Setting One - Scenario A

Results of the top part of Table 4.2 related to GRCT/cGRCT (left sub-table, on the top), show a similar performance on the dataset DataOne (ranging from .85 to .87), but above all they show a growing degree of generalization over DataTwo: from a minimum of .72 (STK) to a maximum of .77 (CSPTK and SPTK). This shows that the degree of generalization increases when using PTKs and SPTKs compared to STKs and SSTKs.

A similar trend can be seen also with regard to LCT/cLCT (central sub-table), where the degree of generalization increase similarly from .72 (STK) to .77 (SPTK). However, performances are sliglty more polarized on DataOne (ranging from .84 to .88).

An even more polarized trend is reported on the sub-table on the right, related to LOCT. In this case, performance on the dataset DataOne range from .80 to .87 and this is the only case in which PTK outperform SPTK. Also the performances on DataTwo are more polarized compared to the other two sub-tables: for the LOCT sub-table scores range from .63 (STK) to .75 (SPTK).

Setting One - Scenario B

Results of bottom part of Table 4.2 related to GRCT/cGRCT (left sub-table), show that PTKs performed better than the other kernels: they reach a mean F1 score of .74 on DataTwo (which is the dataset on which the classifiers of this scenario have been trained) while all the other kernels range from .71 to .72. Also in this case, results over the other dataset (which in this scenario is DataOne) show a growing capability of generalization ranging from .79 (STK) to .84 (CSPTK and SPTK).

[1]Decay factors are meant to penalise long tree fragments, in order to mitigate the risk that their size might excessively affect similarity scores. In this paper, λ is the vertical decay factor, while μ is the horizontal decay factor.

The trend over LCT/cLCT representations (central sub-table) show a similar picture: PTK is the kernel with the best performance over DataTwo (with a Mean F1 score of .72) while the other kernels range between .68 and .71. Also in this case, performances over DataOne show a growing trend ranging from .80 to .84.

Also in this scenario, the sub-table on the right, related to LOCT, is the most polarized one: results on DataTwo range from .64 to .69, with PTK outperforming SPTK; while results on DataOne range from .71 to .83, with SPTK showing again the best ability to generalize over the other dataset.

4.5 Setting Two

In the second study (39), a similar approach has been employed on a dataset created ad hoc for the analysis of argumentative stances of opposition. The dataset has been created by extracting the comments of opposition that citizens wrote on the public website of Nevada Legislature against a bill aiming at regulating euthanasia. The limitation of this study is that the annotation has not been accomplished yet and the number of instances per class is still unbalanced. However, an interesting aspect of this dataset is that a granular labelling system has been proposed in order to assess the ability of Tree Kernel classifiers to detect different kinds of argumentative opposition in a multi-class setting. The original setting presents four levels of granularity in which the dataset can be divided. In the present paper, we are going to select just the granularities with the best balance in the number of instances, namely granularity 2 and granularity 3.

For this reason, similarly to what has been done with Setting One, also Setting Two has been divided into two scenarios. In the first scenario, all classifiers are trained and tested considering three labels (“Slippery Slope”, “Other” and “Testimony”). In the second one, all classifiers are trained and tested considering four labels (“Slippery Slope”, “Other”, “Judgements” and “Testimony”). The number of sentences grouped by class is listed on the left in Table 4.3, while the configurations of the decay factors used in the experiment are reported on the right in Table 4.3.

4.5.1 Results for Setting Two

The top part of Table 4.4 is referred to the first scenario, while bottom part is referred to the second scenario. It is important to underline that this results should be observed by watching not only at the F1 scores (which can be misleading, since they are trained upwards by the results of the class “Other”).

To partially overcome this problem, we should instead focus on the F1 scores of the single classes, as described in the following subsections.

Also in this case, compositional trees are reported jointly with their non-compositional counterpart and should be considered related only with CSP-TKs.

Table 4.2: Results for Scenario A (top) and Scenario B (bottom) of Setting One. Results report the mean F1 score of the binary classification ("Expert" vs "Study/Statistics").

SCENARIO A:

	GRCT/cGRCT		**LCT/cLCT**	
Kernel	DataOne	DataTwo	DataOne	DataTwo
STK	.86	.72	.84	.72
SSTK	.86	.74	.84	.73
PTK	.85	.76	.86	.75
SPTK	.86	**.77**	**.88**	**.77**
CSPTK	**.87**	**.77**	.87	.76

LOCT	
DataOne	DataTwo
.80	.63
.80	.66
.87	.73
.83	**.75**

SCENARIO B:

	GRCT/cGRCT		**LCT/cLCT**	
Kernel	DataOne	DataTwo	DataOne	DataTwo
STK	.79	.71	.80	.69
SSTK	.81	.71	.81	.68
PTK	.83	**.74**	.81	**.72**
SPTK	**.84**	.72	.83	.70
CSPTK	**.84**	.72	**.84**	.71

LOCT	
DataOne	DataTwo
.71	.64
.71	.64
.79	**.69**
.83	.67

Table 4.3: Number of sentences depending on class and granularity (above). Configuration of the kernel parameters for Setting Two (below).

Classes	Granularity 2	Granularity 3
Slippery Slope	82	82
Testimony	133	133
Judgements	part of Other	140
Other	423	283
Total	638	

Kernels	Decay factors	
STK	$\lambda = [.1 - .4]$	
SSTK	$\lambda = [.1 - .4]$	
PTK	$\lambda = [.3 - .4]$	$\mu = [.3 - .4]$
SPTK	$\lambda = [.3 - .4]$	$\mu = [.3 - .4]$
CSPTK	$\lambda = [.3 - .4]$	$\mu = [.3 - .4]$

Setting Two - Scenario A

Results of the top part of Table 4.4 related to GRCT/cGRCT (left sub-table), show that the the Mean F1 score is achieved by the PTK classifier (.76). However, it should be remarked that STK and CSPTK seem to perform better on the class "Testimony" (reaching a F1 score of .71).

In the central sub-table, related to LCT/cLCT, the SPTK and the CSPTK are the best ones both in terms of Mean F1 score (.73-.74) and in terms of balance between the scores for the "Slippery Slope" and "Testimony" classes (which is .59-.60 and .70-.71, respectively). Conversely, the STK, SSTK and PTK show nearly one decimal point less (floating between the values .50-.51).

Regarding the sub-table related to the LOCT representation, there is a clear superiority of the SPTK over the other kernels not only in terms of Mean F1 score (.75), but also in terms of balance between "Slippery Slope" and "Testimony" scores (.67 and .69, respectively). In fact, although PTK reaches .68 on the class "Slippery Slope", it stops at .58 on the class "Testimony".

Setting Two - Scenario B

Regarding the second scenario of the Setting Two (the one considering four labels), results can be seen in the bottom part of Table 4.4. The sub-table on the left, related to GRCT/cGRCT, shows that performances of PTK and CSPTK classifiers are slightly better in terms of mean F1 score (.69). However, when one considers the results of the classes separately, one can see that the SSTK classifier shows better results in the classification of the "Slippery Slope" class (.67), while the CSPTK over the cGRCT representation shows better results in the classification of "Judgements" (.66) and "Testimony" (.74).

Regarding the central sub-table (LCT/cLCT), it seems that the CSPTK classifier is the one that has the best performance in terms of mean F1 (.68).

Table 4.4: Results for Scenario A (top) and B (bottom) of Setting Two. SS = "Slippery Slope", O = "Other", J = "Judgements", T = "Testimony". In bold the maximum F1 scores for SS, J and T.

SCENARIO A:

GRCT/cGRCT

Kernel	SS	O	T	$\overline{F1}$
STK	.61	.90	**.71**	.74
SSTK	.62	.91	.67	.73
PTK	**.67**	.91	.69	.76
SPTK	.58	.87	.68	.71
CSPTK	.63	.90	**.71**	.75

LCT/cLCT

SS	O	T	$\overline{F1}$
.51	.90	.68	.70
.50	.92	**.71**	.71
.51	.90	.68	.70
.60	.89	.70	.73
.59	.91	**.71**	.74

LOCT

SS	O	T	$\overline{F1}$
.60	.87	.61	.70
.62	.89	.63	.71
.68	.86	.58	.71
.67	.89	**.69**	.75

SCENARIO B:

GRCT/cGRCT

Kernel	SS	O	J	T	$\overline{F1}$
STK	.62	.76	.55	.71	.66
SSTK	**.67**	.80	.60	.63	.67
PTK	.64	.79	.61	.71	.69
SPTK	.55	.79	.63	.68	.67
CSPTK	.57	.79	**.66**	**.74**	.69

LCT/cLCT

SS	O	J	T	$\overline{F1}$
.51	.76	.62	.69	.65
.50	.79	**.64**	.71	.66
.49	.80	.61	.75	.65
.52	.78	.58	.71	.65
.52	.79	.63	**.77**	.68

LOCT

SS	O	J	T	$\overline{F1}$
.62	.76	**.54**	.61	.63
.65	.77	.51	.60	.63
.59	.72	.47	.62	.60
.65	.75	.47	**.66**	.63

It is also the classifier that reach the best performance in the classification of the class "Testimony" (.77). Regarding the class "Judgements", the best performances are achieved by the SSTK (.64) and the CSPTK (.63), while the class "Slippery Slope" is the one with the worst performances, with SPTK and CSPTK as best classifiers (stopping at .52).

Regarding the sub-table on the right (related to the LOCT representation), all mean F1 show a similar performance ranging from .60 to .63. Moreover, it can be seen that the performances for the class "Judgements" are the worst ones, where the best score is achieved by the STK (.54). On the other side, SSTK and SPTK achieve the best scores with the class "Slippery Slope" (.65), while the "Testimony" class has the SPTK as most performing classifier (reaching .66).

4.6 Discussion and Conclusions

From the results, some clear patterns can be observed. In general, it seems that PTKs and above all (C)SPTKs collect the best performances. This appears particularly evident in Setting One, when watching the numbers of Table 4.2 from the top to the bottom.

Another trend that can be seen from Setting One is the growing degree of generalization (always watching from the top to the bottom of the tables).

Finally, another interesting trend can be observed on Setting One, namely the growing degree of polarization, watching from the left (GRCT/cGRCT) to the right (LOCT), related to the scores between the most and less performing kernels in each column. It seems that the last table (LOCT) is the most polarized one, reaching up to .13 points of difference between the most and less performing score (i.e. in the column DataTwo, Scenario A; and in the column DataOne, in Scenario B).

To the best of our knowledge, this study is the first attempt to offer a comparative analysis of the combination of five tree representations and five tree kernels in the classification of argumentative stances of opposition and support.

A limitation of this work is that of being related to the specific data employed. The contribution of different tree representation should be assessed also in different scenarios and with different kinds of argumentative data.

Moreover, it is important to investigate the relation between the type of tree representation, the tree kernel function employed and the targeted argument to be classified. In other words, are there tree representation that can express better specific kinds of argument? Are there tree kernel functions that better calculate the similarities between these argumentative representations? This study suggests that the answer to these questions is positive, showing a first attempt of investigation in this direction.

Chapter 5

Transfer Learning to Classify Argumentative Evidences

Original title: **Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification**

Abstract

This work describes a simple Transfer Learning methodology aiming at discriminating evidences related to Argumentation Schemes using three different pre-trained neural architectures. Although Transfer Learning techniques are increasingly gaining momentum, the number of Transfer Learning works in the field of Argumentation Mining is relatively little and, to the best of our knowledge, no attempt has been performed towards the specific direction of discriminating evidences related to Argumentation Schemes. The research question of this paper is whether Transfer Learning can discriminate Argumentation Schemes' components, a crucial yet rarely explored task in Argumentation Mining. Results show that, even with small amount of data, classifiers trained on sentence embeddings extracted from pre-trained transformers can achieve encouraging scores, outperforming previous results on evidence classification.

5.1 Introduction

In the last few years, the use of Transfer Learning methodologies generated in remarkable hype in the State of the Art of many Natural Language Processing tasks. Particularly, the Transformer known as "Bidirectional Encoder Representations from Transformer" (BERT) has shown extremely good results, establishing several new records in terms of metrics results (15). In 2018, BERT obtained new state-of-the-art results on eleven NLP-related tasks. In a couple of years dozens of variants have been developed, establishing other new records not just in English but also in other languages (e.g., the Italian versions, GilBERTo[1] and umBERTo[2], the French camemBERT (55)).

Despite the high celebrity recently achieved by Transfer Learning techniques, these methodologies have been applied relatively few times in Argumentation Mining (61; 68). To the best of our knowledge, this is the first work that explicitly assesses Transfer Learning performances with the aim of discriminating argumentative components related to Argument Schemes (86). On the one side, the approach show to be capable of discriminating argumentative stances of support and opposition related to some famous argumentative patterns (Argumentation Schemes) such as Argument from Expert Opinion, and Argument from negative consequences, showing better results compared to previous studies. On the other side, the approach show that it is possible clustering Argumentation Schemes according to the criteria of the pragmatical dimension, which is a crucial aspect described in the most recent literature about Argumentation Scheme classification (52; 35). In summary, the approach show an ability to classify argumentative evidences not only at fine-grained levels (e.g., different instances of Argument from Expert Opinion) but also at the level of large clusters (like the Argumentation Schemes coming from an external source, a class which according to some classification approaches can be used as first dichotomic criterion of discrimination among schemes (52; 35)).

Section 5.2 will describe the Transfer Learning methodology and the two main settings for the experiments. Section 5.3 will describe the datasets used for the experiments in the two scenarios. Sections 5.4 and 5.5 will show the experimental results on the two scenarios. Section 5.6 will describe the related works. In Section 5.7, some final considerations will conclude the paper.

5.2 Methodology

Transfer Learning methods are generally divided in two approaches: the first approach is called fine-tuning and it consists of using a pre-trained neural architecture (i.e., a Transformer architecture trained on thousands of inputs) as a starting point to perform further training steps on a downstream task (training, thus, the neural architecture on downstream data). The second

[1] https://github.com/idb-ita/GilBERTo
[2] https://github.com/musixmatchresearch/umberto

approach, instead, is that of using a pre-trained neural architecture just to extract the outputs that the Neural Architecture generate for a given input at a specific stage of the neural architecture. For example, a sentence can be used as input and the output generated by the neural architecture can be extracted and used as sentence embeddings, that can represent our sentence in other downstream tasks (noticeably, the extraction of the generated output to be used as embedding can be performed at different stages of the neural architecture, not necessarily at the final layer). In this paper, the second approach will be employed: a famous pre-trained architecture will be selected, some sentences will be used as inputs for this neural architecture, and the output coming from the neural architecture will be employed as sentence embeddings to represent our data in a series of downstream classification tasks.

For the pre-trained embeddings we will employ three pre-trained models: the first one is the famous neural transformer called BERT (15) (specifically, we will use the uncased base version). The second and third models are two recent models which are derived from BERT, namely: distilBERT(73) and RoBERTa(51) (uncased). While BERT base consists of 12 layers, 768 hidden dimensions, 12 self-attention heads and nearly 110M parameters, RoBERTa base consists of 12 layers, 768 hidden dimensions, 12 self-attention heads and 125M parameters. Finally, distilBERT consists of 6 layers, 768 hidden dimensions, 12 self-attention heads and 66M parameters.

To extract the embeddings from the neural models, each input sentence must be firstly tokenized according to the requirements of the given model. Typically, with BERT, a [CLS] and a [SEP] special tokens are inserted at the beginning and at the end of the input (we are interested in the first one which is the token holding the classification output we are interested to extract from the input sentence). Moreover, the length of each input sentence is set to a max length: all sentences longer than that limit are shortened, while all sentences shorter than that limit are padded with the special [PAD] token. This process makes sure that all inputs have the same length before entering the neural architecture. After the tokenization, inputs are passed into the neural architecture of a BERT transformer, while deactivating the calculation of gradients.

After having transformed each input sentence of the test sets into tokens and having used these tokens as inputs for the BERT neural architecture, the resulting extracted embeddings have been used, in turn, as input of a classification using two classification procedure: a Support Vector Machine (SVM) classifier and a Logistic Regression classifier (LRC). Notice that for the experiment on D3 our SVM employed a Linear Support Vector Classifier (Linear SVC), while in all other experiments we employed a standard Support Vector Classifier (SVC).

The classification method is One vs All. Which means that the classification has been performed per each class, considering one class against all the other classes, a typical approach in multiclassification and multilabel scenarios. Finally, all classifiers have been evaluated on the relative testing set.

The experiments have been divided into different scenarios:

1. Baseline scenario: in this scenario, the classification was performed on the same setting of two previous works, taken as baselines for comparison.

2. Extended scenario: in this scenario, the classification was performed on new settings, using an extended version of two datasets from the baseline scenario.

5.3 Data

The experiments of this work have been applied to the datasets listed in Table 5.1, reporting reports also the number of instances for each dataset. These datasets have been selected because their annotations describe classes of argumentative evidence directly related to specific Argumentation Schemes. Importantly, during the experiments, all datasets have been split into train and test sets, following a standard 80/20 ratio.

Table 5.1: Description of all datasets used in this paper.

Dataset	Reference	Classes	Instances
Baseline datasets:			
D1	Al Khatib et al. 2016 (only 2 classes extracted as in (36))	Study, Testimony	653
D2	Aharoni et al. 2014 (only 2 classes selected as in (36))	Study, Expert	569
D3	Liga and Palmirani 2019	Slippery Slope, Testimony, Other	638
Extended datasets:			
D1+	Al Khatib et al. 2016 (3 classes extracted following (36))	Study, Testimony, Anecdotal	2253
D2+	Aharoni et al. 2014	Study, Expert, Anecdotal	1291
D2++	Rinott et al. 2015	Study, Expert, Anecdotal	4692

Regarding the baseline scenario, D1 and D2 are a portion of Al Khatib et al. 2016 and Aharoni et al. 2014 respectively, two important dataset designed by IBM. Only two classes from the original datasets have been selected, reproducing the scenario in (36) in order to have baseline scenarios for our classifiers. D3 is a small dataset (only 638 sentences) from Liga and Palmirani 2019. It is a dataset which has different levels of granularity, depending on

how many classes are considered. In this case we selected granularity three, which contains three labels.

Regarding the extended scenario, the dataset D1+ is an extension of D1: instead of extracting just two classes, it considers three classes. The inputs of the dataset from Al Khatib et al. 2016 (4) are actually structured in a very fragmented way, so we needed to rebuild the sentences following the approach suggested in (36). Similarly, D2+ is an extension of D2 (instead of being a selection of just two classes, it considers three classes). Finally, D2++ is an extended version of the same dataset which, having many more instances, can be a useful benchmark for this kind of classifications.

Importantly, the datasets which have been employed in this work are among the few available datasets containing instances of argumentative evidences which can be related to Argumentation Schemes. Namely, the dataset in Al Khatib et al. 2016 (4) shows instances of argumentative evidences labelled as Study, Testimony and Anecdotal: these evidences support argumentative claims which refer to source-based opinions, this means that they belong to different types of source-based arguments. One of the most famous example of source-based Argumentation Scheme is the well-known Argument from Expert Opinion; another famous scheme is the Argument from witness testimony (more details about this kind of schemes can be found in (35)).

The datasets in Aharoni et al. 2014 (1) and Rinott et al. 2015 (69) present similar source-based Argumentation Schemes (however, this time the labels are Study, Expert and Anecdotal). In this case, the cluster of argumentative evidences labelled with the class Expert are likely to be compatible with the evidences of an Argumentation Scheme from Expert Opinion.

The dataset in Liga and Palmirani 2019 (39) offers instead only one class of evidences which is related to source-based arguments (Testimony) while another class is related to a cluster of evidences which can be related to the Argument from Negative Consequences and the Slippery Slope Arguments.

These three datasets can thus be used to assess whether classifiers are able to discriminate between different cluster of argumentative evidences. Since these argumentative evidences are strictly related to specific clusters of Argumentation Schemes, the ability of classifiers to discriminate different clusters of argumentative evidences is, in our opinion, a crucial step towards Argumentation Scheme discrimination.

5.4 Results for the Baseline Scenario

The classifications in this Section show that the proposed approach is able to outperform recent results in the Argumentation Mining literature. With this purpose, recent results on D1, D2 and D3 are reported (36; 39) and used as baseline for our classifiers.

In this paper, all F1 scores per class are calculated as the mean macro F1 scores, taken from each One-vs-All classification. All these scores are finally averaged and reported as mean F1 (per each classifier, i.e. SVM and LR).

Table 5.2: Results on the baseline classifiers (D1, D2, D3) considering mean F1 scores (macro) and two kinds of classifier. SVM = Support Vector Machine; LR = Logistic Regression; BS = Baseline. The columns whose mean F1 value has an asterisk refers to a Linear Support Vector Classifier. In bold are all the mean F1 scores which overcome the mean F1 of the baseline. The three grey columns represent the best classifiers for the baseline scenario.

Classes	Bert Base		DistilBERT		RoBERTa		BS
	SVM	LR	SVM	LR	SVM	LR	
D1 (Al Khatib et al. 2016)							
Study	.94	.92	.97	.97	.91	.89	.91
Testimony	.93	.91	.97	.96	.89	.86	.92
mean F1	**.94**	**.92**	**.97**	**.96**	.90*	.88	.92
D2 (Aharoni et al. 2014)							
Study	.78	.71	.72	.74	.75	.79	.69
Expert	.75	.68	.67	.72	.73	.77	.78
mean F1	**.76**	.69	.69*	**.73**	**.74***	**.78**	.73
D3 (Liga and Palmirani 2019)							
Slippery Slope	.75	.71	.79	.76	.82	.60	.70
Testimony	.90	.92	.93	.94	.93	.73	.71
Other	.85	.86	.87	.87	.87	.85	.91
mean F1	**.83***	**.82**	**.86***	**.86**	**.87***	.73	.77

As can be seen from Table 5.2, results outperform previous results for the same scenario, showing the ability of Transfer Learning techniques to achieve high performances. As indicated by the bold numbers in Table 5.4, for D1, D2 and D3 there are always at least four classifiers out of six which outperform the baseline.

5.5 Result for the Extended Scenario

The next series of experiments have been performed on an extended version of two datasets from the baseline scenario (D1 and D2), to assess how performances change in a multiclass scenario.

Table 5.3 shows a clear trend, with Logistic Regression on DistilBERT being the best solution for both the dataset extending D1 (i.e., D1+) and the datasets extending D2 (i.e., D2+ and D2++).

Regarding the classifications on D1+, one can see that the best performances are achieved by the Logistic Regression classifier (LR) trained on sentence embeddings extracted using DistilBERT. To have a better understanding of these results, the confusion matrix of the best classifier in this scenario (i.e., Logistic Regression on DistilBERT) are reported with the confusion

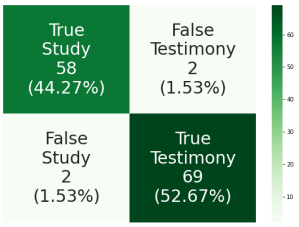
Table 5.3: Results on D1+, D2+ and D2++ considering mean F1 scores (macro) and two kinds of classifiers. SVM = Support Vector Machine; LR = Logistic Regression; BS = Baseline. The columns whose mean F1 value has an asterisk refers to a Linear Support Vector Classifier. In bold are the top mean F1 scores. The three grey columns represent the best classifiers for the extended scenario.

Classes	Bert Base		DistilBERT		RoBERTa	
	SVM	LR	SVM	LR	SVM	LR
D1+ (Al Khatib et al. 2016)						
Study	.83	.85	.83	**.87**	.86	.77
Testimony	.77	.81	.81	**.82**	.78	.70
Anecdotal	.81	.81	.82	**.84**	.83	.77
mean F1	.80	.82	.82*	**.84**	.82*	.75
D2+ (Aharoni et al. 2014)						
Study	.89	.90	**.91**	**.91**	.90	.85
Expert	.91	.91	.92	**.93**	.90	.84
Anecdotal	.92	**.93**	.92	**.93**	.92	.92
mean F1	.91*	.91	**.92***	**.92**	.91*	.87
D2++ (Rinott et al. 2015)						
Study	.93	**.94**	**.94**	**.94**	.92	.90
Expert	.92	.92	**.93**	**.93**	.91	.88
Anecdotal	.91	**.93**	.90	.92	.87	.85
mean F1	.92*	**.93**	.92*	**.93**	.90*	.88

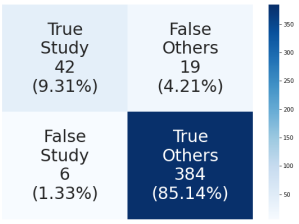
matrix from the best classifier of the baseline scenario (i.e., Support Vector Machine from DistilBERT embeddings from Table 5.2) in Figure 5.1.

Regarding the classifications on D2+ and D2++, one can see that the best performances are achieved by the Logistic Regression classifier (LR) trained on sentence embeddings extracted using DistilBERT and Bert Base. Also in this case, to have a better understanding of the results, the confusion matrices of the best classifiers in this scenario (i.e., Logistic Regression from DistilBERT embeddings and from Bert Base) are reported with the confusion matrix from the best classifier of the baseline scenario (i.e., Logistic Regression from RoBERTa embeddings from Table 5.2) in Figure 5.2.

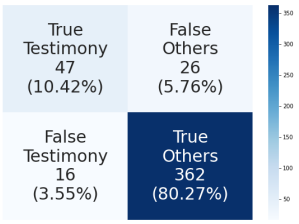
Notice that while confusion matrices for D1 and D2 (in green) show a binary classification, the other confusion matrices in blue (relative to D1+, D2+ and D2++) show a one-vs-all classification. These blue matrices show that classifiers are able to recognize classes also in a multiclass scenario. While Figure 5.1 shows an unbalance (which is probably due to the predominance of the class anecdotal), results in Figure 5.2 seems more balanced: the diagonal



D1 (study vs testimony)
SVM on DistilBERT



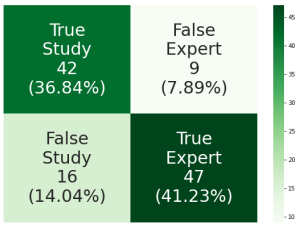
D1+ (study vs others)
LR on DistilBERT



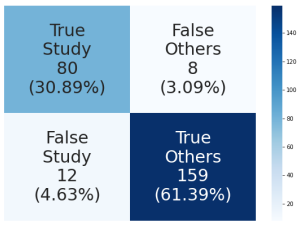
D1+ (testimony vs others)
LR on DistilBERT

Figure 5.1: Confusion matrices for D1 (in green) and D1+. The number of instances and the relative percentages are reported.

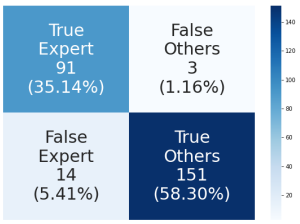
is always a 30/60 ratio, indicating the goodness of predictions.



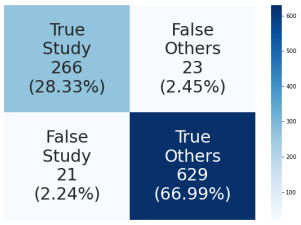
D2 (study vs expert)
LR on RoBERTa



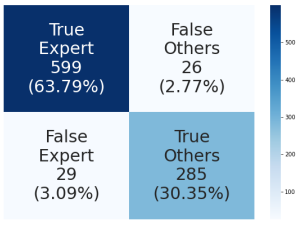
D2+ (study vs others)
LR on DistilBERT



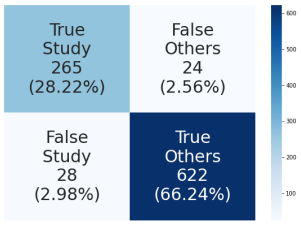
D2+ (expert vs others)
LR on DistilBERT



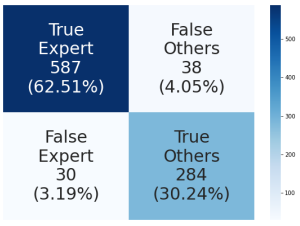
D2++ (study vs others)
LR on DistilBERT



D2++ (expert vs others)
LR on DistilBERT



D2++ (study vs others)
LR on Bert Base



D2++ (expert vs others)
LR on Bert Base

Figure 5.2: Confusion matrices for D2 (in green), D2+ and D2++. The number of instances and the relative percentages are reported.

5.6 Related works

Unfortunately, datasets specifically designed in a way that allow a direct link between classes and specific Argumentation Schemes are very few. A promising and growing resource, in this sense, is the corpora in AIFdb (34) thanks also to the contribute of tools like OVA+ (66) which recently added a very important component for Argumentation Scheme annotation called Argument Scheme Key (35).

Moreover, although there have been different works of text classification in Argumentation Mining, only few studies focused on classification tasks aiming at facilitating the discrimination of Argumentation Schemes.

Rinott et al. 2015 (69) achieved important results on evidence detection employing the dataset D2++. However, the approach is mostly context-dependent, while the present work is not considering the context. In Liga 2019 (36), the classification has been performed using Tree Kernels classifiers on D1 and D2, containing argumentative evidences of support among which it is possible to find evidences directly related to the Argument from Expert Opinion. The work is however limited to a binary classification. A similar approach, in a multiclass scenario, is described in Liga and Palmirani 2019 (39), where Tree Kernels are employed on D3, a small dataset which considers argumentative evidences of opposition among which one can find, for example, the Slippery Slope Argument. Considering these two works as baselines, the approach presented in this paper seems capable of outperforming the previous achievements.

5.7 Conclusion

The datasets analyzed in this work are composed of argumentative evidences which are directly related to different clusters of arguments. For example, many instances which can be found in the datasets of this paper are directly related to the cluster of source-based arguments. Other instances of argumentative evidences are instead specifically related to the Argumentation Scheme from Expert Opinion, while others are related to the cluster which includes the Argument from Negative Consequences and the Slippery Slope Arguments (which do not belong to the cluster of source-based arguments).

We believe that the ability to discriminate different clusters of argumentative evidences is a crucial step in the classification of Argumentation Schemes. For example, the discrimination of clusters of Argumentation Schemes can be performed in a pipeline of binary classifications starting from source-based versus non-source-based arguments and continuing towards more specific binary classifications (similarly to the path of dichotomous choices followed by ASK, the annotation system recently elaborated in (35), which offers a valuable system of classification of Argumentation Schemes).

In general, the results presented in this paper seem encouraging, showing that pre-trained embeddings can outperform previous results in the field of

Argumentation Mining related to the classification of argumentative evidences. An interesting aspect is that the proposed classifiers show encouraging results not only in the discrimination among different source-based argumentative evidences, but also in classifications involving source-based versus non-source-based argumentative evidences (i.e. with dataset D3).

However, further analysis is needed to verify if and how Transfer Learning techniques can discriminate argumentative evidences in such a way that they can facilitate Argumentation Scheme discrimination. In this regard, the present paper is just a preliminary exploration of a promising possible approach. In future works, other Transfer Learning techniques should be assessed too. For example, it could be useful to assess the performances between the two main Transfer Learning techniques: sentence embeddings and fine-tuning. Also, other pre-trained models should be employed and compared (e.g., Xlnet(89), Albert(32)).

A long-term goal is being able to connect natural language argumentative evidences to their specific Argumentation Schemes, which can be a further step in the development of an artificial Natural Argumentation Understanding.

Chapter 6

Transfer Learning for Argumentative Sequence Labelling

Original title: **Argumentative Sequence Labelling Using Transfer Learning**

Abstract

This work presents an approach for Argumentative Sequence Labelling using Transfer Learning. Specifically, a famous pre-trained neural architecture, BERT, has been employed using the Transfer Learning technique known as "fine-tuning" and employing two different data formats for sequence labelling (BIO and BILUO). The neural architecture has been fine-tuned on two famous corpora to recognize not only the boundaries of argumentative units, but also the specific types of argumentative component. The resulting model not only outperforms the results of previous models, but it is also easier to implement, since it does not require highly-engineered features. An evaluation at token-level and another at span-level are performed, as well as a preliminary error analysis.

6.1 Introduction

Transformers like BERT have been extremely popular in the last two years achieving several records in the State of the Art of Natural Language Understanding (15). However, their use in the field of Argumentation Mining (34) has been relatively small so far. In this work, a Transfer Learning approach has been applied to the task of detecting argumentative spans. More precisely, we want to assess the ability of Transfer Learning to facilitate the task of labelling argumentative sequences.

Transfer Learning methodologies, particularly the fine-tuning and contextual-embeddings techniques, have been recently used in many NLP tasks, achieving remarkable steps forward in artificial Natural Language Understanding. BERT and its derivations have been among the most successful natural language models employed for Transfer Learning in the last couple of years. The reason for the success of BERT is the fact that it is able to encode important information about language. To do so, BERT has been pre-trained on a large amount of data (mostly from wikipedia) performing tasks that are designed to force the neural architecture to learn language features. As explained in (15), one of these tasks is the Masked Language task: this task forces BERT's neural architecture to predict randomly masked tokens. Thanks to this simple idea and thanks to its attention-based mechanisms (20), BERT is able to learn many features of human language, and this knowledge is incorporated in its neural architecture. As shown in (15), this knowledge can be then transferred (hence the expression Transfer Learning) to downstream tasks in two ways: the first option is to use BERT to output the embeddings produced by its neural architecture, using these embeddings as features for downstream tasks; the second option is to fine-tune the pre-trained neural architecture, namely performing new training epochs on it, while using downstream data.

Since fine-tuning has been surprisingly efficient with many NLP-related tasks, including sequence labelling tasks such as Named Entity Recognition (15), we think it can be useful to assess BERT's performances on argumentative sequences employing the fine-tuning method to transfer learning from the pre-trained model to the downstream task of labelling argumentative sequences.

In Section 6.2, some related works will be mentioned. Section 6.3 will describe the datasets employed in this work. In Section 6.4, the proposed methodology will be presented. Section 6.5 will describe the achieved results, while Section 6.6 will offer a discussion about the results. Finally, Section 6.7 will conclude the work.

6.2 Related works

While the tasks of sequence labelling and tagging are well known in NLP (54; 3), only few attempts have been performed in the field of Argumentation Mining, especially with regard to the labelling of argumentative spans. To

the best of our knowledge, the first attempt to label argumentative sequences has been proposed by (77), where the modeling of argumentative sequences employs highly-engineered features including Structural, Syntactic, Lexical-Syntactic and Probabilistic elements, while the classification employs a CRF (31) implemented in CRFsuite (62) with an averaged perceptron (12). In this study, (77) adapted the standard BIO format to the purpose of the Argumentative Sequence Labelling, using the labels Arg-B (for those tokens that are at beginning of an argumentative span), Arg-I (for all other argumentative tokens) and Arg-O (for tokens that are not within an argumentative span) with the Argument Annotated Essays Corpus (77). The resulting classification of these three labels achieved a macro averaged F1 score of 0.867.

(2) carried out further experiments, showing that results can be improved by combining all the features provided by (77) with a Bi-LSTM (bidirectional long short-term memory) neural network, and showing that the Bi-LSTM outperforms SVM and CRF classifiers. Importantly, the classification consider not only the Argument Annotated Essays Corpus but also other two famous corpora: the Webis-Editorials-16 corpus (4) and the Argument Annotated UserGenerated Web Discourse corpus (24).

The above-mentioned studies employs highly-engineered features and their models (the CRF and the BiLTSM) are designed for the labelling of argumentative spans in general (without considering the differences between argumentative components, which is considered as a separate task). Moreover, the two models only consider the BIO format, despite the fact that other formats showed better learning performances in some cases, e.g. the BILUO format, which considers also the last token of spans and the spans with just one token (67).

The main novelties of our paper, are: (a) the assessment of the performances of fine-tuning as a Transfer Learning methodology for Argumentative Sequence Labelling on two famous datasets: the Argument Annotated Essays Corpus (79) and the Argument Annotated User-Generated Web Discourse corpus (24); (b) the assessment of the performances of two different data formats for the sequence labelling: BIO and BILUO; (c) the division of the experiment in two separate sub-tasks to assess whether Argumentative Sequence Labelling can be performed not only to detect argumentative vs non-argumentative spans, but also to detect the spans of different argumentative components, e.g. premises, claims (somehow combining sequence labelling and text-classification); (d) an evaluation at span-level, instead of the classic token-level evaluations proposed in previous works. The research questions addressed in this paper are:

1) Can Transfer Learning outperform previous Argumentative Sequence Labelling scores without employing highly-engineered features?

2) Can different labelling formats (e.g. BIO, BILUO) affect these scores significantly?

3) Can we use the same methodology to discriminate argumentative components at a more granular level (distinguishing, for example, premises and

claims)?

6.3 Data

Essays This corpus is composed of 402 persuasive essays written by students and annotated by three experts. Particularly, the annotation considers three types of argumentative units (premises, claims and major claims) considering all the other spans as non-argumentative spans. The authors split the corpus into 322 essays for training and 80 essays for test. This split has been preserved also in the present paper.

Web Discourse This corpus consists of 340 comments written in online newspapers, forums and blogs. In this case, the annotation has been performed by considering a five different types of argumentative unit, which are similar to the famous Toulmin argument model (81): claim, premise, rebuttal, backing and refutation. Since the dataset does not provide any training-test split, we followed the classic 80/20 split, also proposed by (2).

These two datasets have a very different nature: the first corpus is composed by well-structured arguments written by students that were asked to write their arguments about specific topics, the second one shows less predictable ways of expressing arguments, as usual with comments from the Internet.

6.4 Methodology

We divided our experiments into two parts. The first one reproduces the same task of sequence labelling discussed in (77) and (2). The second one is instead an extension which increases the complexity of the labelling task. Table 6.8, in the Appendix A, describes this complexity showing the number of targeted token-level labels considered in the two tasks.

Task 1 focuses on the labelling of sequences as belonging to an argument or not: in this case, each tokens has been labelled as belonging or not to an argumentative span. We created a script to convert each corpus into BIO and BILUO format. For both corpora, the BIO format is represented by three labels (B-ARG, I-ARG and O), while the BILUO format is represented by five labels B-ARG, I-ARG, L-ARG, U-ARG and O (however, no U-ARG have been found in the two corpora).

Task 2 presents exactly the same neural architecture, but we trained it for a more complex task: labelling each tokens as belonging to specific argumentative components. This means that each token is considered as being or not part of a specific argumentative component, such as premise, claim, and so on. In this case, we converted each corpus into a BIO and BILUO using the argumentative components specifically belonging to them. This means that we applied the prefix (B-, I- and L-) to three labels in the case of the Essays corpus (Claim, Premise, MajorClaim) and to five labels in the case of the Web

Discourse corpus (Claim, Premise, Rebuttal, Backing, Refutation). In other words, Task 2 is more complex because the neural architecture tries to classify more types of sequences. To understand this complexity, we can consider the prefixes of the chosen format (B-, I- in the case of the BIO format; B-, I-, L-, U- in the case of the BILUO format) and multiply the number of prefixes by the number of argumentative components in the two corpora (three in the case of the Essays corpus, five in the case of the Web Discourse corpus), plus the label O.

Before feeding our model with the textual data, we used spaCy [1] to automatically separate each document into different sentences. We noticed that this process of separation improves significantly the learning results. Regarding the employed neural architecture, we implemented Google's BERT [2], a famous attention-based (20; 76) Transformer (15). More specifically, we used the pretrained BERT base-uncased (we will sometimes refer to it as BERT_{base}) which is a neural architecture consisting of 12 encoder layers, 768 hidden units and 12 attention heads and it is pretrained on a large amount of data (including Wikipedia), resulting in 110M parameters. To use BERT, each sentence of the corpora has been tokenized using wordpiece (87) tokenization as required by BERT (15). Moreover, we truncated and padded all sentences to a fixed length, we trained our model in 4 epochs, with a batch size of 32 and a learning rate of 5e-5.

Finally, to assess the ability of our model to understand argumentative spans, we evaluated the results of the classification at token-level, considering Precision, Recall and F1 scores for the correct classification of each tokens. Moreover, we followed a stricter methodology at span-level, to evaluate the exact matches of the classified spans, following the evaluation methodology proposed CoNLL-2000 shared task (80).

6.5 Results

This section reports the results for the two parts of the experiment. The pre-trained neural architecture is exactly the same in both in Task 1 and Task 2 and we compared it with a baseline, to show whether a simple pre-trained BERT model can improve previous results. More specifically, we considered as baseline the previous scores achieved by the BiLSTM model proposed by (2) on the same two corpora. While (2) employed just the BIO format, we employed both the BIO and BILUO format. Table 6.1 reports the average macro F1 score of our BERT model for the Essay and Web-Discourse corpora considering both the BIO and BILUO formats.

Importantly, our simple BERT implementation is able to reach and outperforms the results of the highly-engineered BiLSTM baseline (which answers our first research question). When considering the Essays corpus in the BIO format, our model almost reaches the Baseline at 87.43; when considering the

[1] https://spacy.io/
[2] github.com/google-research/bert

	BERT$_{base}$		**Baseline** BiLSTM
Corpus	BIO	BILUO	BIO
Essays	.8743	**.8897**	.8854
Web Discourse	**.6026**	.5266	.5498

Table 6.1: Comparison with the baseline. Mean F1 scores (macro) for the token labelling considering the BIO format (3 classes: B-ARG, I-ARG and O) and the BILUO format (5 classes: B-ARG, I-ARG, L-ARG, U-ARG and O). In bold, the macro F1 scores that outperformed the baseline.

BILUO format, our model slightly outperforms the baseline at 88.97. However, the situation is different when considering the Web Discourse: using the BILUO format, our model performs slightly worse than the baseline at 52.66, while using the BIO format, our model outperforms the baseline with a more evident improvement at 60.26. This is a confirmation that the chosen format can affect the ability of neural architectures to learn (as has been showed for example in (67)), which is also an answers for our second research question.

6.5.1 Task 1: Argumentative Span Detection

Table 6.2 is a more complete report, showing the F1 scores per class for the two datasets. This report provides a better understanding of the ability of the model to classify tokens correctly. For example, we can see that using the BIO format BERT is capable of discriminating correctly between tokens that are at the beginning of an argumentative span with an F1 score of 0.87 while it can recognize inner tokens with a F1 score of 0.92. Interestingly, when using the BILUO format, performances improve. In this case, the support for I-ARG is split (1,264 tokens are considered as L-ARG tokens, concluding an argumentative span), however BERT seems capable to recognize the difference between B-, I- and L- tokens with an average macro F1 of .89 (i.e., .8897 as showed in Table 6.1).

6.5.2 Task 2: Argumentative Component Span Detection

Not surprisingly, results for the Task 2 show lower scores. Tables 6.4 reports the achieved scores for the two datasets considering BIO and BILUO formats.

In the case of the Essays corpus the labelling involved 3 types of argumentative components, while in the case of the Web Discourse corpus the labelling involved 5 types of argumentative components. Noticeably, we attempted two ways for classifying the spans of the argumentative components of the Essays corpus. In the first method, we simply considered all three classes already mentioned (Claim, Major Claim and Premise). In the second method, we considered Claim and Major Claim as the same label, to see if the classifier could recognize the difference between tokens belonging to claims and tokens

Task 1 Essays corpus				
BIO format	P	R	F1	support
B-ARG	.86	.88	.87	1,266
I-ARG	.91	.94	.92	18,750
O	.86	.81	.83	9,412
macro avg:			.87	
weighted avg:			.89	
BILUO format	P	R	F1	support
B-ARG	.87	.88	.88	1,266
I-ARG	.91	.94	.93	17,484
L-ARG	.90	.92	.91	1,266
O	.88	.82	.85	9,412
macro avg:			.89	
weighted avg:			.90	
Task 1 **Web Discourse corpus**				
BIO format	P	R	F1	support
B-ARG	.42	.60	.49	201
I-ARG	.59	.66	.63	7,615
O	.72	.66	.69	10,325
macro avg:			.60	
weighted avg:			.66	
BILUO format	P	R	F1	support
B-ARG	.44	.49	.46	201
I-ARG	.60	.52	.56	7,414
L-ARG	.43	.42	.43	201
O	.68	.74	.71	10,325
macro avg:			.54	
weighted avg:			.64	

Table 6.2: Complete results for the token-level classification of the Task 1 for the two datasets: detecting argumentative spans on the two corpora using BIO and BILUO formats.

belonging to premises.

The left side of Table 6.3 shows the results of the token classification on the Essays corpus considering the BIO and BILUO format for all the three types of argumentative components of the Essays corpus (Claim, Major Claim, Premise). In right side of Table 6.3, instead, we considered Claim and Major Claim as belonging to the same category (a general Claim category). Interestingly, the average macro F1 score improves when considering Major Claim and Claim as a unique class: it increases from .65-.66 (for BIO and BILUO formats, respectively) to .72-.70. Moreover, in both cases, the BIO format outperforms the BILUO format. It is important to notice that even if the support for some classes is relatively small, results seems encouraging (especially in the when considering just two classes). These scores show that BERT can not only recognize argumentative spans, but also understand what kind of argumentative span.

In Table 6.4, the classification of the tokens belonging to argumentative components is related to the Web Discourse corpus. In this case, the model tried to classify 11 (in the BIO format) and 16 (in the BILUO format) token-level labels. The average F1 score is quite low for both the formats, showing that the model struggles in the classification. This is understandable if considering the low support for some classes: there are probably not enough instances for the model to learn. In fact, the labels with the smallest supports are those that achieve the lowest F1 scores. The weighted F1 describe better this unbalanced scenario (.55/.56 for BILUO and BIO respectively). Also in this case, this corpus shows better performances with the BIO format.

6.5.3 Span-level Evaluation

Although some of the results achieved in this work are encouraging, the mere token-level evaluation is not a sufficiently strong measure of evaluation. A token-level evaluation can just provide a view on the ability of our model to classify tokens correctly; however, it does not say anything about the exact span matching. In order to provide our results with a more robust evaluation, an approach proposed at the CoNLL-2000 shared task (80) can be followed. This approach is well known in the the field of Sequence Labelling and Named-Entity Recognition because allows for the evaluation of precision, recall and F1 scores for the exact matching of spans, not just for the tokens. The original script performing these calculations has been originally written in Perl [3], however we implemented it using the python library seqeval [4]. Tables 6.5, 6.6, 6.7 describe the results on the exact matching for all the classes considered in Task 1 and Task 2.

The left side of Table 6.5 considers all the 1,266 argumentative spans extracted from the Essays corpus, showing that the actual matching spans achieve an F1 score of .77 (BIO format) and .79 (BILUO format). Even if this result is lower than the previously mentioned token-level mean F1 scores (.87

[3] https://www.clips.uantwerpen.be/conll2002/ner/bin/conlleval.txt
[4] https://github.com/chakki-works/seqeval

Task 2 **Essays corpus** (considering three classes)				
BIO format	P	R	F1	support
B-CLAIM	.42	.49	.45	304
B-MAJORCLAIM	.69	.68	.69	153
B-PREMISE	.73	.69	.71	809
I-CLAIM	.43	.49	.46	3,920
I-MAJORCLAIM	.72	.71	.72	1,970
I-PREMISE	.81	.79	.80	12,860
O	.85	.82	.83	9,412
macro avg:			.66	
weighted avg:			.75	
BILUO format	P	R	F1	support
B-CLAIM	.40	.49	.44	304
B-MAJORCLAIM	.68	.67	.68	153
B-PREMISE	.73	.69	.71	809
I-CLAIM	.38	.51	.44	3,616
I-MAJORCLAIM	.73	.66	.69	1,817
I-PREMISE	.80	.77	.78	12,051
L-CLAIM	.43	.51	.46	304
L-MAJORCLAIM	.73	.67	.70	153
L-PREMISE	.75	.73	.74	809
O	.87	.81	.84	9,412
macro avg:			.65	
weighted avg:			.74	

Task 2 **Essays corpus** (considering two classes)				
BIO format	P	R	F1	support
B-CLAIM	.62	.62	.62	457
B-PREMISE	.71	.73	.72	809
I-CLAIM	.61	.62	.61	5,890
I-PREMISE	.78	.81	.79	12,860
O	.86	.80	.83	9,412
macro avg:			.72	
weighted avg:			.76	
BILUO format	P	R	F1	support
B-CLAIM	.57	.62	.60	457
B-PREMISE	.72	.70	.71	809
I-CLAIM	.58	.62	.60	5,435
I-PREMISE	.79	.77	.78	12,049
L-CLAIM	.59	.64	.61	457
L-PREMISE	.76	.72	.74	809
O	.84	.83	.84	9,412
macro avg:			.70	
weighted avg:			.76	

Table 6.3: Task 2 for the Essays corpus, using BIO and BILUO formats and considering classes Claim, MajorClaim and Premise (the first Table) and Claim and Premise (the second Table).

Task 2 Web Discourse corpus				
BIO format	P	R	F1	support
B-CLAIM	.71	.14	.23	36
B-PREMISE	.30	.42	.35	106
B-BACKING	.45	.12	.19	43
B-REBUTTAL	.00	.00	.00	12
B-REFUTATION	.00	.00	.00	4
I-CLAIM	.40	.42	.41	680
I-PREMISE	.44	.45	.45	4,247
I-BACKING	.32	.26	.29	2,089
I-REBUTTAL	.32	.15	.20	453
I-REFUTATION	.00	.00	.00	146
O	.68	.71	.70	10,325
macro avg:			.26	
weighted avg:			.56	

Task 2 Web Discourse corpus				
BILUO format	P	R	F1	support
B-CLAIM	.43	.08	.14	36
B-PREMISE	.33	.35	.34	106
B-BACKING	.50	.12	.19	43
B-REBUTTAL	.00	.00	.00	12
B-REFUTATION	.00	.00	.00	4
I-CLAIM	.35	.34	.35	644
I-PREMISE	.43	.38	.40	4,141
I-BACKING	.31	.23	.27	2,046
I-REBUTTAL	.11	.08	.10	441
I-REFUTATION	.00	.00	.00	142
L-CLAIM	.33	.31	.32	36
L-PREMISE	.33	.36	.34	106
L-BACKING	.00	.00	.00	43
L-REBUTTAL	.00	.00	.00	12
L-REFUTATION	.00	.00	.00	4
O	.67	.76	.71	10,325
macro avg:			.20	
weighted avg:			.55	

Table 6.4: Task 2 for the Web Discourse corpus, using BIO (first table) and BILUO (second table) formats and considering classes Claim, Premise, Rebuttal, Backing, Refutation.

Span-level evaluation - Taks 1 (Essays corpus)				
BIO format	P	R	F1	support
ARG	.73	.82	.77	1,266
BILUO format	P	R	F1	support
ARG	.74	.84	.79	1,266

Span-level evaluation - Taks 1 (Web Discourse corpus)				
BIO format	P	R	F1	support
ARG	.12	.34	.18	201
BILUO format	P	R	F1	support
ARG	.06	.19	.10	201

Table 6.5: Span-level report of the Task 1 for the Essays corpus (the first Table) and for the Web Discourse corpus (the second Table).

and .89 for the BIO and BILUO format respectively), it might be a more robust way of assessing the performance of Argumentative Sequence Labelling.

The right side of Table 6.5 shows that the actual matching for the 201 argumentative spans is lower than expected. In this case, the difference from the token-level evaluation is even more evident (scores plummeted from .60 and 54 to .18 and .10 for BIO and BILUO respectively). Again, we wonder whether this sharper decrease is due to the less structured composition of the argumentative text from the web, compared to the students' essays.

Regarding Task 2, Table 6.6 shows the exact span matching for the Essays corpus when considering three and two classes. When using three classes, BERT achieves a macro F1 score of .47 (for both BIO and BILUO formats), while, also in this case, the classification joining the Claims and Major Claims classes performs better, achieving .49 (BILUO) and .51 (BIO).

Table 6.7, finally, describes the exact span matching for the five classes of the Web Discourse corpus classified in Task 2. Not surprisingly, results are significantly low for the same reasons mentioned before: small number of instances per class and probably the presence of less well-structured (or more variable) argumentative structures compared to the Essays corpus.

6.6 Discussion on the results

The research questions mentioned before can now be addressed: results show, in fact, that a simple BERT model can reach and even outperform the previous models. This is important, because previous records employed highly engineered models, while we are using a simple pre-trained model without changing its neural architecture. We also assessed that the choice of BIO and BILUO can indeed affect results. However, it seems that the Web

Span-level evaluation - Taks 2 (Essays corpus) considering three classes				
BIO format	P	R	F1	support
CLAIM	.19	.39	.26	304
MAJORCLAIM	.36	.59	.45	153
PREMISE	.50	.61	.55	809
macro avg:			.47	
micro avg:			.45	
BILUO format	P	R	F1	support
CLAIM	.22	.43	.29	304
MAJORCLAIM	.36	.53	.43	153
PREMISE	.50	.62	.55	809
macro avg:			.47	
micro avg:			.46	

Span-level evaluation - Taks 2 (Essays corpus) considering two classes				
BIO format	P	R	F1	support
CLAIM	.36	.55	.44	457
PREMISE	.49	.64	.55	809
macro avg:			.51	
micro avg:			.51	
BILUO format	P	R	F1	support
CLAIM	.35	.53	.42	457
PREMISE	.48	.60	.53	809
macro avg:			.49	
micro avg:			.49	

Table 6.6: Span-level report of the Task 2 for the Essays corpus, using BIO and BILUO formats and considering classes Claim, MajorClaim and Premise (the first Table) and classes Claim and Premise (the second Table).

Span-level evaluation - Taks 2 (Web Discourse corpus)				
BIO format	P	R	F1	support
CLAIM	.04	.11	.06	36
PREMISE	.06	.21	.10	106
BACKING	.03	.09	.04	43
REBUTTAL	.03	.08	.04	12
REFUTATION	.44	.49	.46	4
macro avg:			.07	
micro avg:			07	

Span-level evaluation - Taks 2 (Web Discourse corpus)				
BILUO format	P	R	F1	support
CLAIM	.04	.11	.06	36
PREMISE	.06	.21	.10	106
BACKING	.03	.09	.04	43
REBUTTAL	.03	.08	.04	12
REFUTATION	.44	.49	.46	4
macro avg:			08	
micro avg:			.08	

Table 6.7: Span-level report of the Task 2 for the Web Discourse corpus, using BIO and BILUO formats and considering classes Claim, Premise, Backing, Rebuttal and Refutation.

Discourse corpus is the one that is most affected by this change. In this regard, we wonder whether this is due to the composition of the argumentative data coming from the web (which are probably less well-structured than the Essays corpus, or structurally more variable). Further studies are needed to investigate this aspect. Finally, answering to the third research question, we showed that the sequence labelling achieves encouraging results also at more granular levels, discriminating among different kinds of argumentative components. Span-level evaluation, however, show poorer scores.

6.6.1 Preliminary Error Analysis

We are currently performing an Error Analysis which shows that BERT can recognize patterns of language commonly employed in natural arguments (e.g. the use of connectors such as "In my view," or "Finally,") and also the beginning and the conclusion of argumentative spans are detected with precision (please, see Appendix B). However, missing information about the context unavoidably affects results: for example, some sentences which seems argumentative (especially in other contexts) but are not (w.r.t. the topic of the discussion) might generate false positives (see the first false positive reported in Appendix B). Other false positives might be generated by connectors such as "because" (see the second false positive reported in Appendix B). Finally, there are cases in which the argumentative sentence is detected but the match is not perfect (BERT wrongly adds or misses argumentative spans).

6.7 Conclusion

This study outperformed previous results in the State of the Art of Argumentative Sequence Labelling, showing that BERT can reach and outperform previous benchmarks on the Argument Annotated Essays corpus and on the Argument Annotated User-Generated Web Discourse corpus. More precisely, we divided the work in two Tasks: in the first one, we focused on the recognition (at token-level) of argumentative spans vs non-argumentative spans,

while in the second task we focused on a more fine-grained classification of the tokens as belonging to specific argumentative components.

Importantly, we showed that the choice of the labelling format (e.g., BIO, BILUO) can affect scores, although the extent of such influence seems related to the underlying data employed. In this regard, further research is required to understand what kind of format are more performing and how these performances are related to the underlying argumentative data. Furthermore, we showed that BERT is able not only to recognize sequences of argumentative tokens (considering argumentative vs non-argumentative), but also to recognize what kind of argumentative components are involved (premise, claim, rebuttal, etc.).

Although results are encouraging, we are skeptical about token-level evaluations. We thus proposed to use a more robust methodology to evaluate Argumentative Sequence Labelling tasks following the suggestion from CoNLL-2000 shared task. Using such approach, we extended our token-level evaluations with span-level evaluations, showing the actual ability of BERT to recognize exact matches of argumentative spans for all the proposed experiments.

In future, the performances of other Transformers might be assessed. In any case, we think that this work can be a starting point for future research employing more complex Transfer Learning architecture for Argumentative Sequence Labelling.

Appendix A. Description of the span classes.

Task 1 (Argumentative span detection)			
Corpora	Prefixes	Token-level classes	Number of classes
Essays (in BIO format)	B- I-	B-ARG I-ARG O	3
Essays (in BILUO format)	B- I- L- U-	B-ARG I-ARG L-ARG U-ARG† O	4†
Web Discourse (in BIO format)	B- I-	B-ARG I-ARG O	3
Web Discourse (in BILUO format)	B- I- L- U-	B-ARG I-ARG L-ARG U-ARG† O	4†
Task 2 (Argumentative component span detection)			
Corpora	Prefixes	Token-level classes	Number of classes
Essays (in BIO format)	B- I-	B-CLAIM, B-MAJORCLAIM, B-PREMISE, I-CLAIM, I-MAJORCLAIM, I-PREMISE O	7
Essays (in BILUO format)	B- I- L- U-	B-CLAIM, B-MAJORCLAIM*, B-PREMISE, I-CLAIM, I-MAJORCLAIM*, I-PREMISE L-CLAIM, L-MAJORCLAIM*, L-PREMISE U-CLAIM†, U-MAJORCLAIM*†, U-PREMISE† O	10† 7*†
Web Discourse (in BIO format)	B- I-	B-CLAIM, B-PREMISE, B-BACKING, B-REBUTTAL, B-REFUTATION I-CLAIM, B-PREMISE, I-BACKING, I-REBUTTAL, I-REFUTATION O	11
Web Discourse (in BILUO format)	B- I- L- U-	B-CLAIM, B-PREMISE, B-BACKING, B-REBUTTAL, B-REFUTATION I-CLAIM, I-PREMISE, I-BACKING, I-REBUTTAL, I-REFUTATION L-CLAIM, L-PREMISE, L-BACKING, L-REBUTTAL, L-REFUTATION U-CLAIM†, U-PREMISE†, U-BACKING†, U-REBUTTAL†, U-REFUTATION† O	16†

Table 6.8: Description of the span classes for the two parts of the experiments, depending on corpus and format. The dagger (†) refers to the fact that no U-tokens have been actually found. The asterisk (*) refers to the fact that the classes Claim and Major Claim can be joint into a unique class, producing 7 total token-level labels instead of 10.

Appendix B. Error Analysis

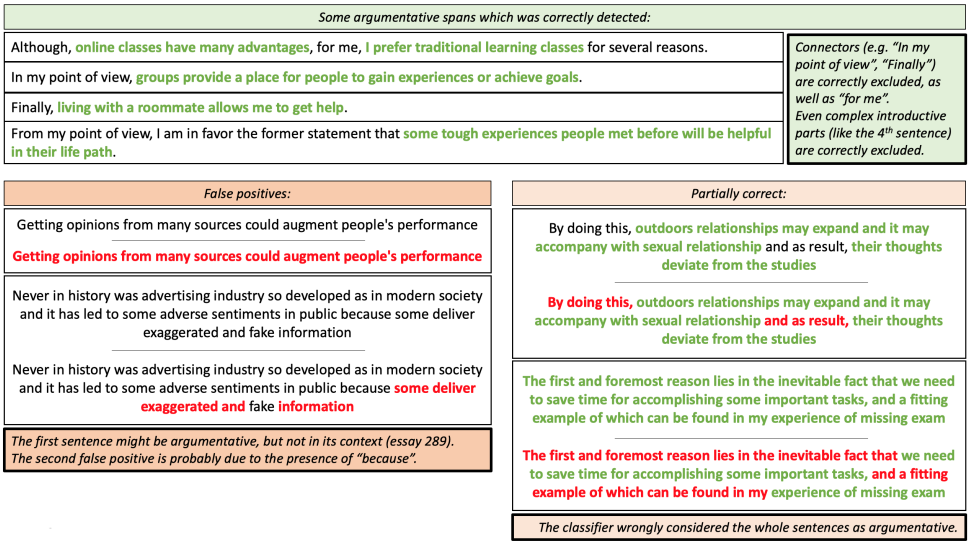


Figure 6.1: A preliminary error analysis on the Essay corpus for task 1. We selected 8 results: 4 correct, 2 partially correct, 2 false positives. Regarding false positives and partially correct sentences, both the true spans and the predicted span are reported: true spans are on the top, while their relative predictions are immediately below them. The red color is just used within predictions, to show errors (both false positives and false negatives). Please, compare predictions with the sentences above them. While false positive are those predictions where the classifier wrongly detected an non-existent argumentative span, partially correct sentences are those sentences where the match between true spans (on the top) and their relative predictions (below them) is not perfect, which means that the classifier either added or missed an argumentative span.

Chapter 7

Transfer Learning for Deontic Rule Classification

Original title: **Transfer Learning for Deontic Rule Classification: the Case Study of GDPR**

Abstract

This work focuses on the automatic classification of deontic sentences. It presents a novel Machine Learning approach which combines the power of Transfer Learning with the information provided by two famous LegalXML formats. In particular, different BERT-like neural architectures have been fine-tuned on the downstream task of classifying rules from the European General Data Protection Regulation (GDPR) by using Akoma Ntoso and LegalRuleML. This work shows that fine-tuned language models can leverage the information provided in LegalXML documents to achieve automatic classification of deontic sentences and rules.

7.1 Introduction

The ability to automatically detect deontic rules directly from natural language sentences is a crucial long-term goal in the field of Artificial Intelligence and Law (AI&Law), and in legal argumentation (7; 88). One of the obstacles of this kind of tasks is the lack of available data designed *ad hoc* for the classification of deontic rules. Since the annotation of this kind of datasets is time-consuming and requires experts of domain, the process of creating datasets to automatically recognize deontic rules can be costly.

Another obstacle, related to the first one, is that datasets might be too small to train Machine Learning classifiers, especially when dealing with deep neural architectures. In this regard, the Computational Linguistics communities have recently experienced a big step forward in many State of the Art challenges thanks to the so-called Transfer Learning methods, where pre-trained neural architectures are employed in downstream tasks. In this sense, BERT (15) was one of the most famous examples of successful pre-trained neural architectures, used in many downstream tasks even with very small datasets (40).

On the one side, this work wants to show the potential of using LegalXML documents as source of data. On the other side, it wants to exploit Transfer Learning ability to have good performances on downstream tasks even when dealing with relatively small datasets. Finally, this work tackles the automatic classification of deontic rules directly from natural language, an AI&Law task which has been approached by the community only marginally.

7.2 Methodology

The approach proposed in this work is twofold. On the one side, it wants to prove that Akoma Ntoso (63) and LegalRuleML (8) can be combined to feed Machine Learning algorithms with reliable data for the classification of deontic rules. On the other side, it wants to test the use of Transfer Learning on the task of deontic rule classification. The first aspect (i.e., the combination of Akoma Ntoso and LegalRuleML) is related to the methodology that has been used to extract the legal knowledge and data. The second aspect (i.e., the usage of Transfer Learning as Machine Learning algorithm) is related to the methodology for the classification. The combination of these two methodological aspects (i.e., method of data/knowledge extraction and method of classification) can be defined as a Hybrid AI approach, since it combines symbolic knowledge with sub-symbolic knowledge (22; 71).

7.2.1 Data extraction method

Regarding the first aspect, the idea is that combining Akoma Ntoso and LegalRuleML is a powerful and convenient solution to extract labelled data for the classification of rules and deontic modalities. In fact, while LegalRuleML describes the logical sphere and contains information about the deontic rules,

it also contains information about where to find these rules in the respective legal resource (where natural language can be found). While LegalRuleML is an optimal representation of the legal logical sphere, Akoma Ntoso is an optimal representation of the legal natural language. In fact, Akoma Ntoso contains crucial pieces of information not only about the legal document, but also about the structure of natural language where the pieces of deontic information are located. This can facilitates the reconstruction of the natural language sentences, especially in those cases where the deontic information is split in different structural portions within the legal source. In sum, LegalRuleML describes the logical sphere of legal rules and connect it to the legal source, while Akoma Ntoso is a rich and complete representation of the content of legal sources. In this work, these two formats have been used to create a dataset, where natural language sentences are taken (and sometimes reconstructed) from Akoma Ntoso, while the classes are extracted from LegalRuleML.

7.2.2 Classification method

Regarding the second aspect, related to the classification methodology, the idea is that Transfer Learning methods can have good performances even with small datasets. Transfer Learning generally consists in the use of neural architectures which have been pre-trained on a huge amount of data. On the one side, the results of this process of pre-training a neural architecture over a huge amounts of data generates language models which can achieve remarkable results in many NLP tasks; on the other side, the "knowledge" acquired by this pre-trained neural architectures during the training, can be "transferred" (hence the name "Transfer Learning") on downstream, more specific, tasks, which can even use small datasets. Importantly, there are two major ways of using Transfer Learning: a famous approach is to use the pre-trained neural architecture to extract embeddings to represent our data, these embeddings can then be classified using a common classification algorithm (this approach is more similar to what we described in (40)). Another approach is that of fine-tuning the pre-trained neural architecture on a downstream task, which means that the output the classification will be generated by the output layer of the neural network. In this work, we used this second approach.

7.3 Related Works

The first studies which tackled the classification of deontic elements focused on the deontic elements as parts of a wider range of targets. Among these first attempts to classify obligations (among other targets) from legal texts there is (30), which focused on the regulations of Italy and US. Their method employed word lists, grammars and heuristics to extract obligations among other targets such as rights and constraints. Another work which tackled the classification of deontic statements is (84), which focused on the German tenancy law and classified 22 classes of statements (among which there were also prohibitions

and permissions). The method used active learning with Multinomial Naive Bayes, Logistic Regression and Multi-layer Perceptron classifiers, on a corpus of 504 sentences. In (21), the authors used Machine Learning to extract six classes of normative relationships: prohibitions, authorizations, sanctions, commitments and powers.

Perhaps the first study which mainly focused on the deontic sphere is (59). This work was focused on the financial legislation to classify legal sentences using a Bi-LSTM architecture, with a training dataset containing 1,297 instances (596 obligations, 94 prohibitions, and 607 permissions). The work also inspired (10), which introduced a hierarchical Bi-LSTM with self-attention to extract sentence embeddings, with the goal to detect contractual obligations and prohibitions.

Since the publication of BERT (15), a growing number of studies employed Transfer Learning methods. To the best of our knowledge, the first study which employed BERT for the classification of deontic sentences is (28). While (10) focused on just prohibitions and obligations, (28) also focused on permissions, using BERT and achieving an average precision and recall of 90% and 89.66% respectively. Another recent work is (74), which used four pre-trained architectures (BERT, DistilBERT, RoBERTa, and ALBERT) but focused just on the binary detection duties vs non-duties.

Also our work presents a Transfer Learning approach of Machine Learning, which combines the symbolic information of LegalXML formats with the sub-symbolic power provided by different pre-trained language models (among which the famous BERT (15)). Moreover, leveraging the information channeled by the biggest LegalRuleML Knowledge Base available, we present four different scenarios of classification:

1. Rule vs Non-rule
2. Deontic vs Non-deontic
3. Obligation vs Permission vs None
4. Obligation vs Permission vs Constitutive Rule vs None

The novelty and the power of Transfer Learning methodologies jointly with the combined use of Akoma Ntoso and LegalRuleML are two major contributions of this study, along with the design of the experimental settings in 4 different classification scenarios. Another point which is worth mentioning is that LegalXML formats such as Akoma Ntoso and LegalRuleML are documents which are written by legal experts. In other words, for the task of detecting deontic classes, the extraction of data from this kind of documents arguably offers a more convenient and robust solution compared to the use of datasets which are only partially related to the deontic sphere.

7.4 Data

The data used in this study consists of 707 sentences extracted from the European General Data Protection Regulation (GDPR). To extrapolate this dataset, we used the *DAta Protection REgulation COmpliance* (DAPRECO) Knowledge Base (70), which is the LegalRuleML representation of the GDPR and the the biggest knowledge base in LegalRuleML (8), as well as the biggest knowledge base formalized in Input/Output Logic (53). The current version of the DAPRECO[1] includes 966 formulæ in reified Input/Output logic: 271 obligations, 76 permissions, and 619 constitutive rules. As explained in (70), the number of constitutive rules is much higher than permissions and obligations because constitutive rules are needed to trigger special inferences for the modelled rules. This means that constitutive rules are an indicator of the existence of a rule, without giving information about deontic modalities.

Importantly, DAPRECO also contains the connections between each formula and the corresponding structural element (paragraphs, point, etc) in the Akoma Ntoso representation of the GDPR[2]. In other words, using a LegalRuleML knowledge base like DAPRECO and the corresponding Akoma Ntoso representation, it is possible to connect the logical-deontic sphere of legal documents (in this case the 966 Input/Output formulæ provided by DAPRECO) to the natural language statements in the legal text (provided by the Akoma Ntoso representation of the GDPR).

Importantly, this combination of Akoma Ntoso and LegalRuleML facilitate also the reconstruction of the exact target in terms of natural language. For example, many obligations of legal texts are split into lists, and Akoma Ntoso is useful to reconstruct those pieces of natural language into a unique sentence. For example, Article 5 of the GDPR[3] states:

> *Article 5*
>
> **Principles relating to processing of personal data**
>
> 1. Personal data shall be:
>
> (a) processed lawfully, fairly and in a transparent manner in relation to the data subject ('lawfulness, fairness and transparency');
>
> (b) collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further

[1]The DAPRECO knowledge base can be freely downloaded from its repository: https://github.com/dapreco/daprecokb/blob/master/gdpr/rioKB_GDPR.xml

[2]The Akoma Ntoso representation of the GDPR is currently accessible from https://github.com/guerret/lu.uni.dapreco.parser/blob/master/resources/akn-act-gdpr-full.xml, where it can be freely downloaded

[3]https://eur-lex.europa.eu/eli/reg/2016/679/oj#d1e1807-1-1

processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes ('purpose limitation');

(c) adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed ('data minimisation');

[...]

As can be seen, *paragraph 1* of *Article 5* is a list composed of an introductory part ("Personal data shall be:") and different points. To be concise, only the first three points of *paragraph 1* are reported here, namely *point a*, *point b* and *point c*. From the point of view of the natural language, each deontic sentence is split between the introductory part (which contains the main deontic verb "shall") and the text of each point. While the introductory part contains the main deontic verb, the actual deontic information is contained within each point.

The Akoma Ntoso formalization for *point a* would be:

```
<article eId="art_5">
    <num> Article 5 </num>
    <heading eId="art_5__heading">
        Principles relating to processing of personal data
    </heading>
    <paragraph eId="art_5__para_1">
        <num> 1. </num>
            <intro> <p> Personal data shall be: </p> </intro>
            <point eId="art_5__para_1__content__list_1__point_a">
                <num> (a) </num>
                <content>
                    <p> processed lawfully, fairly and in a
transparent manner in relation to the data subject ('lawfulness,
fairness and transparency');

                    </p>
```

```
        </content>
    </point>
    [...]
```

In DAPRECO, which uses the LegalRuleML formalization[4], a series of <LegalReference> elements can be found, which contain the structural portion where the deontic formulas are located, referenced by using the Akoma Ntoso naming convention[5]. For example, the reference of the above mentioned *point a* can be found in DAPRECO as:

```
<LegalReference
refersTo = "gdprC2A5P1p1ref"
refID = "GDPR:
art_5__para_1__content__list_1__point_a"/>
```

Where the "refersTo" attribute indicates the internal ID of the reference, and the "refID" attribute indicates the external ID of the reference using the Akoma Ntoso naming convention. The prefix "GDPR" stands for the Akoma Ntoso uri of the GDPR, namely "/akn/eu/act/regulation/2018-05-25/eng@2018-05-25/!main#".

In turn, this <LegalReference> element is then associated to its target group of logical statements, which collects the group of logical formulas related to this legal reference (so, in this case, related to *point a* of the first paragraph of *Article 5*). Such association is modelled as follows:

```
<Association>
    <appliesSource keyref="#gdprC2A5P1p1ref" />
    <toTarget keyref="#statements1" />
</Association>
```

Where the attribute "keyref" of the target connects the source to the collection of statements whose "key" attribute is "statements1":

[4] https://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/legalruleml-core-spec-v1.0.html

[5] https://docs.oasis-open.org/legaldocml/akn-nc/v1.0/csprd01/akn-nc-v1.0-csprd01.html

```
<Statements key="statements1">

    <ConstitutiveStatement key="statements1Formula1">
        <Rule closure="universal">
            <if>[...]</if>
            <then>[...]</then>
        </Rule>
    </ConstitutiveStatement>

    <ConstitutiveStatement key="statements1Formula2">
        <Rule closure="universal">
            <if>[...]</if>
            <then>[...]</then>
        </Rule>
    </ConstitutiveStatement>

</Statements>
```

Importantly, each statement in natural language can have more than one formula in the logical sphere. This is the reason why the element <Statements> here shows a collection of two logical formulæ.

To finally associate the portion of natural language sentences extracted from Akoma Ntoso to a class related to the logical sphere, the identification keys of the two formulæ can be tracked into the <Context> element.

```
<Context key="context_1"
type="rioOnto:obligationRule">
    <inScope
keyref="#statements1Formula1" />
    [...]
</Context>

<Context key="context_3"
type="rioOnto:constitutiveRule">
    <inScope
keyref="#statements1Formula2" />
```

```
    [...]
</Context>
```

As can be seen, the first formula (identified as “statements1Formula1”) is associated with the ontological class “obligationRule”, while the second formula (identified as “statements1Formula2”) is associate with the ontological class “constitutiveRule”. In other words, the portion of natural language expressed in *point a* of the 1_{st} paragraph of *Art. 5* of the GDPR is represented in the logical sphere as a constitutive rule and an obligation rule.

The full path from the natural language sphere (located in the Akoma Ntoso) to the logical sphere (i.e. the LegalRuleML formalization) where the deontic classes are located is described in Figure 7.1. The figure further explains how the combination of Akoma Ntoso and LegalRuleML can be employed to extract labelled data.

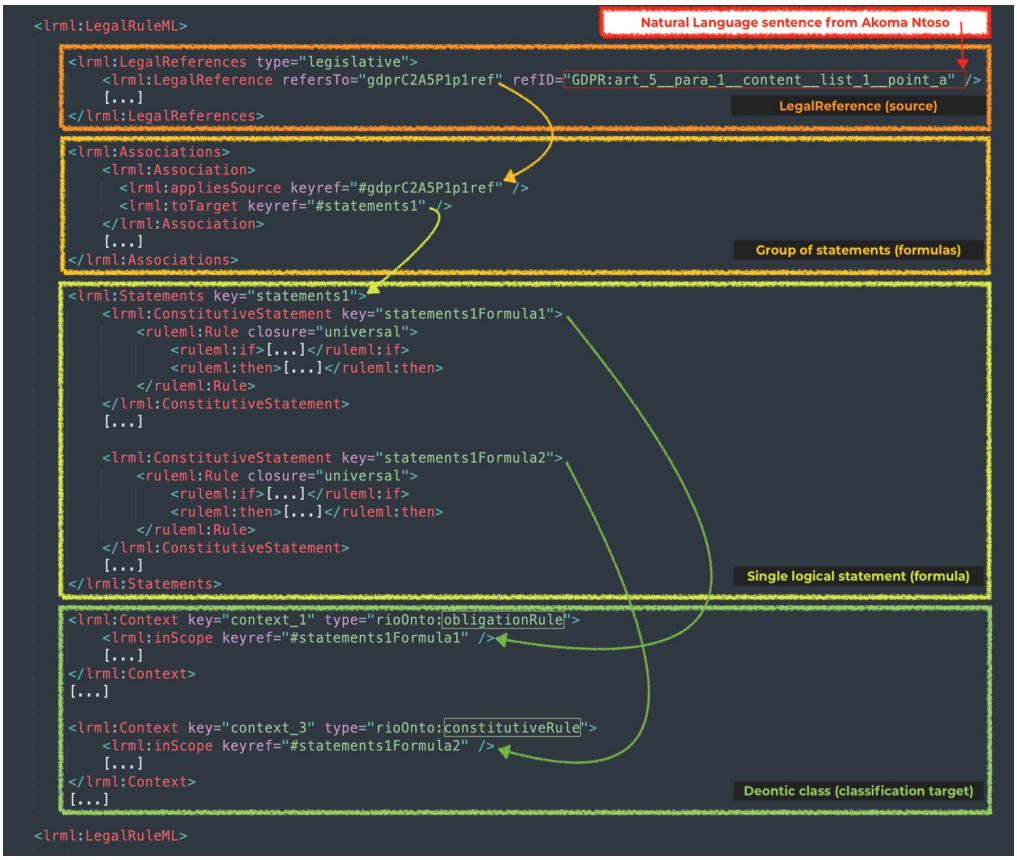


Figure 7.1: Description of the process of class extraction from Akoma Ntoso and DRAPRECO.

7.5 Experiment settings and results

At the end of the process of extraction, we achieved a total of 707 labelled sentences, which have been reconstructed whenever they were split into lists (thanks to the structural information provided by Akoma Ntoso). The labels of these sentences are the same as those provided by DAPRECO with the addition of a 'none' category:

1. obligationRule;
2. permissionRule;
3. constitutiveRule;
4. none;

The class obligationRule is referred to those sentences which have at least one obligation in their related formulæ. The class permissionRule is referred to those sentences which have at least one permission in their related formulæ. The class constitutiveRule is referred to those sentences which have at least one constitutive rule in their related formulæ. These labels allowed 4 different experimental settings, as shown in Table 7.1:

Table 7.1: Number of instances per class per scenario.

	Classes	Instances
Scenario 1	rule	260
	non-rule	447
Scenario 2	deontic	204
	non-deontic	503
Scenario 3	obligationRule	156
	permissionRule	44
	none	503
Scenario 4	obligationRule	156
	permissionRule	44
	constitutiveRule	56
	none	447

Scenario 1 is a binary classification task and aims at discriminating between rule and non-rule instances. In this scenario, all labels other than "none" are considered rule, while "non-rule" is just an alias for "none". Scenario 2 focus on a binary classification between deontic instances (i.e., any sentence labelled as either obligationRule or permissionRule) and non-deontic instances (i.e., all instances which are labelled neither as "obligationRule" nor as "permissionRule"). Scenario 3 is a multiclassification which considers the classes obligationRule, permissionRule and none (with "constitutiveRule" considered as part of the latter). Scenario 4 is a multiclassification which considers the classes obligationRule, permissionRule, constitutiveRule and

Table 7.2: Results for the four experimental scenarios. P = precision; R = recall, F1 = F1-score, S/T = Support/Total ratio

	Classes	BERT			DistilBERT			LegalBERT			S/T
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
1	rule	.74	.95	.83	.77	.95	.85	.75	.77	.76	39/260
	non-rule	.96	.81	.88	.97	.84	.90	.87	.85	.86	68/447
		Accuracy .86 Macro avg .86 Weighted avg .86			Accuracy .88 Macro avg .87 Weighted avg .88			Accuracy .82 Macro avg .81 Weighted avg .82			Total: 107/707
2	deontic	.74	.90	.81	.82	.90	.86	.80	.77	.79	31/200
	non-deontic	.96	.87	.91	.96	.92	.94	.91	.92	.92	76/507
		Accuracy .88 Macro avg .86 Weighted avg .88			Accuracy .92 Macro avg .90 Weighted avg .92			Accuracy .88 Macro avg .85 Weighted avg .88			Total: 107/707
3	obligationRule	.74	.83	.78	.74	.83	.78	.63	.92	.75	24/156
	permissionRule	.50	.83	.62	.36	.67	.47	.56	.83	.67	6/44
	none	.97	.88	.92	.94	.84	.89	1.0	.82	.90	76/503
		Accuracy .87 Macro avg .78 Weighted avg .88			Accuracy .83 Macro avg .71 Weighted avg .84			Accuracy .84 Macro avg .77 Weighted avg .85			Total: 106/703
4	obligationRule	.70	.79	.75	.80	.83	.82	.84	.67	.74	24/156
	permissionRule	.60	.50	.55	.40	.67	.50	.17	.67	.28	6/44
	constitutiveRule	.36	1.0	.53	.47	.89	.62	.89	.89	.89	9/56
	none	1.0	.73	.84	.94	.76	.84	.96	.79	.87	67/447
		Accuracy .75 Macro avg .67 Weighted avg .78			Accuracy .78 Macro avg .69 Weighted avg .80			Accuracy .76 Macro avg .69 Weighted avg .81			Total: 106/703

none. For the multi-classifications (i.e. Scenario 3 and 4) four statements have been removed, since the classes obligationRule and permissionRule overlapped.

As far as the experimental settings are concerned, the dataset was divided into 70% for the training phase, 15% for the test and 15% for the validation; and for all instances, a max length of 30 was applied.

Regarding the Transfer Learning architecture, 3 pre-trained language model have been fine-tuned, namely BERT (15), DistilBERT (73), and the LegalBert trained on EurLex (11). These three neural architectures were fine-tuned by adding two linear layers with a ReLu activation fuction and with a dropout of 0.2 after each activation, and a final output layer was added for the classification, through a softmax activation function. The fine-tuning process of these 3 neural architectures was performed in 10 epochs using learning rate 1e-3, and a batch size of 32.

The final results on the validation set are reported in Table 7.2, where it can be seen that DistilBERT outperforms the other classifiers in the two binary classifications, with an average score reaching .88 in the first scenario and .92 in the second scenario.

The results for the third and fourth scenarios are less straightforward and show that BERT slightly outperforms other classifiers in the third scenario, while LegalBERT outperformed the other models in the fourth scenario.

The main problem for the multiclassifications, is the class unbalance and the restricted amount of instances for some classes.

7.6 Conclusions

The contribution of this work is showing how Transfer Learning methods can leverage the information provided in LegalXML to train classifiers capable of automatically classifying deontic sentences and rules.

Since we were not interested in the internal elements of the logical formulæ, we just addressed the ontological classes of each rule, modelled within DAPRECO. However, in the future we want to create classifiers that directly address the internal components of each rule, trying to find a match between portions of natural language and portions of rules. Also, we would like to create a stronger connection with the ontological sphere by using PrOnto (64), strengthening this hybrid AI approach, which combines symbolic knowledge with sub-symbolic methods.

In general, the ability to connect each internal component (or at least some) of the deontic formulæ contained in DAPRECO directly to the portion of natural language where the component is communicated or expressed is a crucial future direction, and an important step towards the long-term goal of filling the gap between natural language and the logical-inferential sphere, which would generate a more reliable and explainable Artificial Intelligence.

Chapter 8

Conclusion and Future Work

In this Thesis, a collection of works is presented, aiming at detecting patterns and rules from argumentative and legal texts. Different methodologies have been employed to achieve our goals, with a hybrid approach which aimed at combining symbolic and sub-symbolic Artificial Intelligence, searching for ways to connect "top-down" structured knowledge (typically related to symbolic AI) with "bottom-up" data-driven methods (typically associated with sub-symbolic AI).

As described in the introduction of this Thesis (see Section 1.1.3), the collection of works described in the previous chapters addresses the following main research questions:

- (Q1) Can NLP methods detect and classify **argumentative support** and **opposition**?
- (Q2) Can we bridge the gap between natural language and the logical-inferential sphere by using **hybrid approaches**?
- (Q3) Can NLP methods **detect rules**?

These research questions are not trivial and pose a number of obstacles both in terms of available resources and with regards to the methodology to use.

Among the major issues, we mentioned the following ones:

- (Problem P1) Ontological complexity of argumentative patterns;
- (Problem P2) Scarcity of available data both for the detection of argumentative patterns and for the detection of rules;
- (Problem P3) Complex features are often needed;
- (Problem P4) Difficulty of connecting natural language to the logical-inferential sphere.

We showed that Q1 and Q3 have a positive answer. In fact, the presented methodologies are capable of classifying argumentative evidences of support and opposition, as well as rules and deontic modalities. Moreover, to answer research question Q3, we showed how to leverage structured LegalXML data to feed different neural architectures, and this hybrid approach is one of the potential solutions to shorten the distance (or gap) between natural language and the logical-inferential sphere.

The achievements of each paper, as well as the connections between the papers and the relative research questions and challenges they tackle, are reported in the Table 8.1. From a general point of view, the aim of all the papers of this Thesis is to give some little contributions towards an important long-term direction: using machine learning and NLP to unlock automatic reasoning directly on natural language. While we are aware that this goal is too ambitious for a simple PhD project, we still tried to tackle some of the challenges that this long-term goal involves, that can be broadly grouped under the research questions Q1, Q2, Q3, as well as issues P1, P2, P3, P4.

In the first part of the project, we dealt with arguments and argumentative patterns. In particular, we showed that Machine Learning classifiers can discriminate among different argumentative patterns (and this can be referred to as a classification task). In the **first paper** (36), we showed that argumentative support can be successfully discriminated (which is an answer to **Q1**) also when classifying argumentative patterns which are ontologically related (which tackles **P1**). Also, it showed how to combine two famous IBM datasets, augmenting the amount of data for the training phase (which tackles **P2**). Above all, we employed a methodology which combined TFIDF and Tree Kernels, which allowed us to avoid highly engineered features (which tackles **P3**).

In the **second paper** (39), we showed that Tree Kernels can discriminate also among different types of argumentative opposition (which is an answer to **Q1**). This is true also when classifying argumentative patterns at different levels of granularity (which tackles **P1**). Importantly, in this work a brand new dataset has been annotated (thus tackling **P2**)

In the **third paper** (41), we assessed the combination of different Tree Kernel functions with different types of tree representation. While, on the one side, this showed a more solid understanding of the potential of Tree Kernel methods (**Q1**), it also presented an important assessment of the potential of some tree-structured data representations, thus going towards the direction of **Q2**, especially when considering the use of "smoothed" and "compositional" trees, which allow for a *hybrid* (see Section 1.1.2) connection between tree structures and the semantic sphere.

In the **fourth paper** (40), we showed one of the most important and successful NLP methodologies of the last years: Transfer Learning. More precisely, we used Transfer Learning to search for a better and better answer to **Q1** by exploring three different pre-trained language models and two classification algorithms. Moreover, this paper uses a Transfer Learning method which is based on the extraction of sentence embeddings from the pre-trained

neural architecture. This provide us with a "learned" semantic representation of the sentences, generated by the pre-trained neural architecture, and this can be considered a hybrid approach which is partially sub-symbolic (which goes towards the direction of **Q2-P4**). Importantly, the Transfer Learning method, which outperformed all our previous works, is capable of working in downstream tasks even when operating on very small datasets (**P2**).

The **fifth paper** (44), describe another type of Transfer Learning method which is dedicated to the detection of argumentative spans. This is not a classification task like the previous works. Instead, it is a "Sequence Labelling" task applied to argumentative components. Since its aim is to detect the internal components of arguments (i.e. premises and conclusions), this work goes towards the direction of bridging the gap between natural language and the logical-inferential sphere (thus answering to **Q2** and tackling **P4**).

The **sixth paper** (45), employs a hybrid method which leverage symbolic knowledge to generate labelled data (facing **P2**) for the detection of rules and deontic modalities (answering **Q3**). Moreover, it provides a combination of symbolic knowledge with sub-symbolic Machine Learning methods (which goes towards the direction of **Q2** and tackles **P4**).

Although both argumentative patterns and rules suffer from the same gap between natural language and the logical-inferential sphere, in the last paper of this Thesis, we showed that the situation is more favourable in the legal domain, thanks to the existence of some LegalXML formats. In fact, in the legal domain, the combination between these formats (formalized by legal experts) provides the scientific community with a robust way to tackle the gap between legal natural language and legal reasoning. We demonstrated this by combining Akoma Ntoso (a format which is capable of representing exhaustively natural language and internal structures of legal documents) and LegalRuleML (a format which is capable of encoding the logical sphere of legal documents into formulæ) in the context of a sub-symbolic neural classification for the detection of rules, permissions and obligations.

Although also the results of this last part of the Thesis were quite encouraging (which confirms the capability of Tranfer Learning methods to achieve good performances in many experimental settings), we must point out that the classification of a sentence as being a rule or not is still not enough to unlock the **long-term goal of reasoning automatically from natural language**. To achieve this long-term goal more efforts are needed. In this regard, in future works, we will focus, on the matching between legal rules' internal components and their relative span of text within natural language. In other words, we will try to combine the approach proposed in the paper about the argumentative sequence labelling described in Section 1.3.5 and Chapter 6 (where we successfully identified the exact spans of premises and conclusions within argumentative sentences) together with the approach of the paper about the classification of rules described in Section 1.3.6 and Chapter 7 (where we designed a Hybrid AI method which combined the symbolic information of LegalXML formats with a sub-symbolic, data-drive neural classification).

Achievement	Target
Argumentative Evidences Classification and Argument Scheme Detection Using Tree Kernels	
Tree Kernels (SPTK) can perform fine-grain discrimination between different kinds of argumentative evidence, while avoiding the need of engineering sophisticated features	Q1 P1 P3
Classifiers combining Tree Kernel and TFIDF show better performances compared to simple TFIDF classifiers in the classification of argumentative stance of support	Q1
Two famous IBM datasets are combined, not only to increase the amount of data for the training of the algorithm, but also to assess the generalization ability of classifiers over different data	Q1 P2
Classifying Argumentative Stances of Opposition Using Tree Kernels	
Tree Kernels (PTK) can classify also argumentative stances of opposition at different levels of granularity	Q1 P1
The combination with *n*-grams does not show any significant improvement	Q1
A new dataset has been created specifically for the task	P2
Combining Tree Kernels and Tree Representations to Classify Argumentative Stances	
5 different Tree Kernels are used	Q1
5 different Tree Representations are used, including "smoothed" and "compositional" trees, which can connect the structural layer of trees with a semantic layer	Q1 Q2
The most performative combination of Tree Kernels and Tree Representations are showed	Q1
The performance of the combinations seems to show that PTKs and (C)SPTKs outperform other Tree Kernels while providing a greater degree of generalization	Q1
Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification	
Transfer Learning classifiers outperform Tree Kernel classifiers	Q1 Q2
Three pre-trained architecture (Bert, Roberta, DistilBERT) are compared together with two different classification algorithms (Logistic Regression and Support Vector Machines): the combination of DistilBERT sentence embeddings with a Logistic Regression classifier	Q1 Q2
Contextual embeddings produced by pre-trained language models can provide powerful 'learned' semantic representations	Q2-P4
Transfer Learning can reach the best performances even on small datasets	P2
Transfer Learning for Argumentative Sequence Labelling	
A fine-tuning Transfer Learning method show that it is possible to apply Argumentative Sequence Labelling, locating the exact span of argumentative components	Q2-P4
To have a stronger evaluation of the method, the experiment was performed at different levels (i.e., token-level and span-level) and two labelling strategies have been used (i.e., BILUO and BIO)	Q2-P4
Transfer Learning for Deontic Rule Classification	
LegalXML structures and meta-data can be leveraged for the extraction of deontic data and the subsequent detection of deontic rules	Q3 P2
Combining LegalRuleXML and Akoma Ntoso with sub-symbolic classifications can help to fill the gap between the sphere of natural language and the logical sphere	Q2-P4

Table 8.1: Research questions and problems targeted by each study.

Bibliography

[1] Aharoni E., Polnarov A., Lavee T., Hershcovich D., Levy R., Rinott R., Gutfreund D., Slonim N., 2014, A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics, Proceedings of the First Workshop on Argumentation Mining, pp 64–68

[2] Ajjour Y., Chen W.-F., Kiesel J., Wachsmuth H., Stein B., 2017, Unit segmentation of argumentative texts, Proceedings of the 4th Workshop on Argument Mining, pp 118–128

[3] Akbik A., Blythe D., Vollgraf R., 2018, Contextual string embeddings for sequence labeling, Proceedings of the 27th International Conference on Computational Linguistics, pp 1638–1649

[4] Al Khatib K., Wachsmuth H., Kiesel J., Hagen M., Stein B., 2016, A news editorial corpus for mining argumentation strategies, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 3433–3443

[5] Annesi P., Croce D., Basili R., 2013, Towards Compositional Tree Kernels, Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, pp 15–23

[6] Annesi P., Croce D., Basili R., 2014, Semantic compositionality in tree kernels, pp 1029–1038

[7] Ashley K. D., 2017, Artificial intelligence and legal analytics: new tools for law practice in the digital age. Cambridge University Press

[8] Athan T., Boley H., Governatori G., Palmirani M., Paschke A., Wyner A., 2013, Oasis legalruleml, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, pp 3–12

[9] Cabrio E., Villata S., 2018, Five Years of Argument Mining: a Data-driven Analysis., IJCAI, pp 5427–5433

[10] Chalkidis I., Androutsopoulos I., Michos A., 2018, Obligation and prohibition extraction using hierarchical rnns, arXiv preprint arXiv:1805.03871

[11] Chalkidis I., Fergadiotis M., Malakasiotis P., Aletras N., Androutsopoulos I., 2020, LEGAL-BERT: The muppets straight out of law school, arXiv preprint arXiv:2010.02559

[12] Collins M., Duffy N., 2002, Convolution kernels for natural language, Advances in neural information processing systems, pp 625–632

[13] Croce D., Moschitti A., Basili R., 2011a, Structured lexical similarity via convolution kernels on dependency trees, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp 1034–1046

[14] Croce D., Moschitti A., Basili R., 2011b, Semantic convolution kernels over dependency trees: smoothed partial tree kernel, Proceedings of the 20th ACM international conference on Information and knowledge management, pp 2013–2016

[15] Devlin J., Chang M.-W., Lee K., Toutanova K., 2018, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805

[16] Dragoni M., Villata S., Rizzi W., Governatori G., 2016, Combining NLP approaches for rule extraction from legal documents

[17] Feng V. W., Hirst G., 2011, Classifying arguments by scheme, Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp 987–996

[18] Filice S., Moschitti A., 2018, Learning pairwise patterns in Community Question Answering, Intelligenza Artificiale, 12, 49

[19] Filice S., Castellucci G., Croce D., Basili R., 2015, Kelp: a kernel-based learning platform for natural language processing, Proceedings of ACL-IJCNLP 2015 System Demonstrations, pp 19–24

[20] Galassi A., Lippi M., Torroni P., 2019, Attention, please! a critical review of neural attention models in natural language processing, arXiv preprint arXiv:1902.02181

[21] Gao X., Singh M. P., 2014, Extracting normative relationships from business contracts, Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pp 101–108

[22] Gomez-Perez J. M., Denaux R., Garcia-Silva A., 2020, Hybrid Natural Language Processing: An Introduction. Springer International Publishing, Cham, pp 3–6, doi:10.1007/978-3-030-44830-1_1, https://doi.org/10.1007/978-3-030-44830-1_1

[23] Grennan W., 1997, Informal logic: Issues and techniques. McGill-Queen's Press-MQUP

[24] Habernal I., Gurevych I., 2017, Argumentation mining in user-generated web discourse, Computational Linguistics, 43, 125

[25] Habernal I., Eckle-Kohler J., Gurevych I., 2014, Argumentation Mining on the Web from Information Seeking Perspective., ArgNLP

[26] Hinton M. D., 2018, Slippery Slopes and Other Consequences, Logic and Logical Philosophy, 27, 453

[27] Hovy D., Shrivastava S., Jauhar S. K., Sachan M., Goyal K., Li H., Sanders W., Hovy E., 2013, Identifying metaphorical word use with tree kernels, Proceedings of the First Workshop on Metaphor in NLP, pp 52–57

[28] Joshi V., Anish P. R., Ghaisas S., 2021, Domain adaptation for an automated classification of deontic modalities in software engineering contracts, Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 1275–1280

[29] Katzav J., Reed C. A., 2004, On argumentation schemes and the natural classification of arguments, Argumentation, 18, 239

[30] Kiyavitskaya N., Zeni N., Breaux T. D., Antón A. I., Cordy J. R., Mich L., Mylopoulos J., 2008, Automating the extraction of rights and obligations for regulatory compliance, International Conference on Conceptual Modeling, pp 154–168

[31] Lafferty J., McCallum A., Pereira F. C., 2001, Conditional random fields: Probabilistic models for segmenting and labeling sequence data

[32] Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R., 2019, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942

[33] Lawrence J., Reed C., 2016, Argument Mining Using Argumentation Scheme Structures., COMMA, pp 379–390

[34] Lawrence J., Reed C., 2020, Argument Mining: A Survey, Computational Linguistics, 45, 765

[35] Lawrence J., Visser J., Reed C., 2019, An online annotation assistant for argument schemes, Proceedings of the 13th Linguistic Annotation Workshop, pp 100–107

[36] Liga D., 2019a, Argumentative evidences classification and argument scheme detection using tree kernels, Proceedings of the 6th Workshop on Argument Mining, pp 92–97

[37] Liga D., 2019b, Comparing Tree Kernels performances in argumentative evidence classification, CLADAG 2019

[38] Liga D., Palmirani M., 2019a, Classifying argumentative stances of opposition using Tree Kernels, Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp 17–22

[39] Liga D., Palmirani M., 2019b, Detecting "slippery slope" and other argumentative stances of opposition using tree kernels in monologic discourse, International Joint Conference on Rules and Reasoning, pp 180–189

[40] Liga D., Palmirani M., 2020a, Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification, Proceedings of the 20th Workshop on Computational Models of Natural Argument, p. 11

[41] Liga D., Palmirani M., 2020b, Combining tree kernels and tree representations to classify argumentative stances, International Workshop on Artificial Intelligence for Legal Documents (AI4LEGAL2020), pp 12–23

[42] Liga D., Palmirani M., 2020c, Argumentation Schemes as Templates? Combining Bottom-up and Top-down Knowledge Representation., CMNA@COMMA, pp 51–56

[43] Liga D., Palmirani M., 2020d, Uncertainty in Argumentation Schemes: Negative Consequences and Basic Slippery Slope., CLAR, pp 259–278

[44] Liga D., Palmirani M., 2022a, Argumentative Sequence Labelling Using Transfer Learning, (forthcoming)

[45] Liga D., Palmirani M., 2022b, Transfer Learning for Deontic Rule Classification, (forthcoming)

[46] Lippi M., Torroni P., 2015a, Context-independent claim detection for argument mining

[47] Lippi M., Torroni P., 2015b, Argument mining: A machine learning perspective, International Workshop on Theory and Applications of Formal Argumentation, pp 163–176

[48] Lippi M., Torroni P., 2016, MARGOT: A web server for argumentation mining, Expert Systems with Applications, 65, 292

[49] Lippi M., Lagioia F., Contissa G., Sartor G., Torroni P., 2015, Claim detection in judgments of the EU Court of Justice, AI Approaches to the Complexity of Legal Systems, pp 513–527

[50] Lippi M., Pałka P., Contissa G., Lagioia F., Micklitz H.-W., Sartor G., Torroni P., 2018, CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service, Artificial Intelligence and Law, pp 1–23

[51] Liu Y., et al., 2019, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692

[52] Macagno F., Walton D., Reed C., 2017, Argumentation Schemes. History, Classifications, and Computational Applications, History, Classifications, and Computational Applications (December 23, 2017). Macagno, F., Walton, D. & Reed, C, pp 2493–2556

[53] Makinson D., Van Der Torre L., 2000, Input/output logics, Journal of philosophical logic, 29, 383

[54] Manning C. D., 2011, Part-of-speech tagging from 97% to 100%: is it time for some linguistics?, International conference on intelligent text processing and computational linguistics, pp 171–189

[55] Martin L., Muller B., Suárez P. J. O., Dupont Y., Romary L., de la Clergerie É. V., Seddah D., Sagot B., 2019, CamemBERT: a Tasty French Language Model, arXiv preprint arXiv:1911.03894

[56] Mayer T., Cabrio E., Lippi M., Torroni P., Villata S., 2018, Argument mining on clinical trials, Computational Models of Argument: Proceedings of COMMA 2018, 305, 137

[57] Moschitti A., 2006, Efficient convolution kernels for dependency and constituent syntactic trees, European Conference on Machine Learning, pp 318–329

[58] Moschitti A., Pighin D., Basili R., 2008, Tree kernels for semantic role labeling, Computational Linguistics, 34, 193

[59] Neill J. O., Buitelaar P., Robin C., Brien L. O., 2017, Classifying sentential modality in legal language: a use case in financial regulations, acts and directives, Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law, pp 159–168

[60] Nguyen T.-V. T., Moschitti A., Riccardi G., 2009, Convolution kernels on constituent, dependency and sequential structures for relation extraction, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, pp 1378–1387

[61] Niven T., Kao H.-Y., 2019, Probing Neural Network Comprehension of Natural Language Arguments, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 4658–4664

[62] Okazaki N., 2007, CRFsuite: a fast implementation of Conditional Random Fields (CRFs), http://www.chokkan.org/software/crfsuite/

[63] Palmirani M., Vitali F., 2011, Akoma-Ntoso for legal documents, Legislative XML for the semantic Web, pp 75–100

[64] Palmirani M., Martoni M., Rossi A., Bartolini C., Robaldo L., 2018, Pronto: Privacy ontology for legal compliance, pp 142–151

[65] Pollock J. L., 1995, Cognitive carpentry: A blueprint for how to build a person. Mit Press

[66] REED M. J. J. L. C., 2014, OVA+: An argument analysis interface, Computational Models of Argument: Proceedings of COMMA, 266, 463

[67] Ratinov L., Roth D., 2009, Design Challenges and Misconceptions in Named Entity Recognition, Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pp 147–155

[68] Reimers N., Schiller B., Beck T., Daxenberger J., Stab C., Gurevych I., 2019, Classification and Clustering of Arguments with Contextualized Word Embeddings, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 567–578

[69] Rinott R., Dankin L., Alzate C., Khapra M. M., Aharoni E., Slonim N., 2015, Show me your evidence-an automatic method for context dependent evidence detection, Proceedings of the 2015 conference on empirical methods in natural language processing, pp 440–450

[70] Robaldo L., Bartolini C., Lenzini G., 2020, The DAPRECO knowledge base: representing the GDPR in LegalRuleML, Proceedings of the 12th Language Resources and Evaluation Conference, pp 5688–5697

[71] Rodríguez-Doncel V., Palmirani M., Araszkiewicz M., Casanovas P., Pagallo U., Sartor G., 2020, Introduction: A Hybrid Regulatory Framework and Technical Architecture for a Human-Centered and Explainable AI, pp 1–11

[72] Rooney N., Wang H., Browne F., 2012, Applying kernel methods to argumentation mining, Twenty-Fifth International FLAIRS Conference

[73] Sanh V., Debut L., Chaumond J., Wolf T., 2019, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108

[74] Shaghaghian S., Feng L. Y., Jafarpour B., Pogrebnyakov N., 2020, Customizing Contextualized Language Models for Legal Document Reviews, 2020 IEEE International Conference on Big Data (Big Data), pp 2139–2148

[75] Shawe-Taylor J., Cristianini N., et al., 2004, Kernel methods for pattern analysis

[76] Spliethöver M., Klaff J., Heuer H., 2019, Is It Worth the Attention? A Comparative Evaluation of Attention Layers for Argument Unit Segmentation, arXiv preprint arXiv:1906.10068

[77] Stab C. M. E., 2017, PhD thesis, Technische Universität Darmstadt

[78] Stab C., Gurevych I., 2014, in Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers. pp 1501–1510

[79] Stab C., Gurevych I., 2017, Parsing argumentation structures in persuasive essays, Computational Linguistics, 43, 619

[80] Tjong Kim Sang E. F., Buchholz S., 2000, Introduction to the CoNLL-2000 Shared Task Chunking, Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop

[81] Toulmin S., et al., 1958, The uses of argument

[82] Vishwanathan S., Smola A. J., et al., 2004, Fast kernels for string and tree matching, Kernel methods in computational biology, 15, 113

[83] Wachsmuth H., Da San Martino G., Kiesel D., Stein B., 2017, The impact of modeling overall argumentation with tree kernels, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 2379–2389

[84] Waltl B., Muhr J., Glaser I., Bonczek G., Scepankova E., Matthes F., 2017, Classifying Legal Norms with Active Machine Learning., JURIX, pp 11–20

[85] Walton D., 2015, The basic slippery slope argument, Informal Logic, 35, 273

[86] Walton D., Reed C., Macagno F., 2008, Argumentation schemes

[87] Wu Y., et al., 2016, Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144

[88] Wyner A., Peters W., 2011, On rule extraction from regulations, pp 113–122

[89] Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R. R., Le Q. V., 2019, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems, pp 5754–5764

[90] van Engers T. M., van Gog R., Sayah K., 2004, A case study on automated norm extraction, Legal Knowledge and Information Systems. Jurix, pp 49–58