UNIVERSIDADE DE ÉVORA

Automatic Extraction and Structure of Arguments in Legal Documents

Prakash Poudyal

Tese apresentada à Universidade de Évora
para obtenção do Grau de Doutor em Informática

Orientador	*Professor Paulo Quaresma*
Co-Orientadora	*Professor Teresa Gonçalves*

December 27, 2018

INSTITUTO DE INVESTIGAÇÃO E FORMAÇÃO

UNIVERSIDADE DE ÉVORA

Automatic Extraction and Structure of Arguments in Legal Documents

Prakash Poudyal

Tese apresentada à Universidade de Évora
para obtenção do Grau de Doutor em Informática

Orientador	*Professor Paulo Quaresma*
Co-Orientadora	*Professor Teresa Gonçalves*

December 27, 2018

INSTITUTO DE INVESTIGAÇÃO E FORMAÇÃO

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisors; Professor Paulo Quaresma and Professor Teresa Gonçalves for their continued support, motivation, enthusiasm, and immense knowledge. Their guidance has helped me throughout the research and writing of this thesis.

In addition to my supervisors, I would also like to thank the faculty members of the Department of Informatics, University of Evora: Professor Salvador Abreu, Professor Luís Miguel Rato, Professor José Saias, Professor Irene Rodrigues, and last but not least, Professor Pedro Salgueiro, all of whom have helped me by providing advice whether it be regarding research or in the lab.

I would like to thank Professor Marie-Francine Moens from Language Intelligence & Information Retrieval Lab (LIIR) Research lab, Department of Computer Science at KU Leuven, Belgium for providing the ECHR Dataset. I would like to thank Professor Renata Vieira, Pontifícia Universidade Católica do Rio Grande do Sul, Brazil and her team for helping to clarify various issues that I encountered. Lots of thanks to Professor Chris Reed and his team provided me with guidance and support while I was in Arg-Tech Centre, University of Dundee, Scotland. My sincere thanks also go to Professor Alessandro Moschitti from the University of Trento, Italy for solving several problems that I faced during the Tree Kernel experiments.

I must thank Professor Russell Alpizar-Jara, Professor Imme Berg, Professor José Carlos Oliveira, Mrs. Teresa Nogueiro, and Mrs. Gabriela Simões from the University of Evora, Portugal for taking care of me and being of immense help during my stay in Portugal.

I would like to thank Professor Bhola Thapa, Professor Subodh Sharma, Professor Sanjay Khanal, Mr. Mukunda Prasad Upadhaya, Professor Manish Pokharel, Professor Pursotam Kharel, Professor Bal Krishna Bal, and Professor Manoj Shakya of Kathmandu University (KU), Nepal who have tirelessly encouraged my research work.

I am grateful to have been befriended by Professor Cristina Água-Mel, University of Coimbra, Portugal, and Lol Grant from the U.S. who have always given me advice in academic writing and thesis editing.

A special thanks goes to my friends Roy Bayot, Masud Rana Rashel and Anish Byanjankar, for being a part of the project related discussions, and who also gave advice on several experimental endeavors, including thesis writing. I would also like to thank Tran Nam, Swarnendu, Abdel Raouf, Sajib, KP dai for their help.

My parents: Professor Hom Nath Poudyal, Mrs. Geeta Poudyal, and my brother, Dr. Youb Raj Poudyal, sister in law Mrs. Prativa Gautam and nephews Abinab and Arshab receive my deepest gratitude for the love and faith in me. Their words of encouragement, their confidence in my ability, and their affection motivated me to persist. My family in law members, Sukadev Nepal, Arati Nepal and Sushil Nepal, Prabha Nepal also deserve thanks for their constant support and love.

My loving wife, Rama Nepal, receives tons of thanks for her eternal affections and heartwarming encouragement, and my daughter Medhavi Poudyal who is missing me a lot and waiting for me to return with lots of chocolates.

Last but not the least, I would like to thank Erasmus Mundus scholarship (EMMA-WEST 2013) project for granting me a scholarship which made it possible for me to research at the University of Evora, Portugal.

Prakash Poudyal
December 4, 2018

Contents

List of Figures

List of Tables

List of Acronyms

AB Argument Builder

ACIA Appropriate Cluster Identification Algorithm

AEI Argument Element Identifier

AFs Argument Frameworks

AI Artificial Intelligence

AIF Argument Interchange Format

AML Argument Markup Language

AS Argument Structurer

BAF Bipolar Argument Framework

BOW Bag of Words

CBOW Continuous Bag of Words

CDC Context-Dependent Claim

CDCD Context Dependent Claim Detection

CFG Context-Free Grammar

CFWS Clustering based on Frequent Word Sequence

CFWMS Clustering based on Frequent Word Meaning Sequence

CNNs Convolutional Neural Networks

CRF Conditional Random Field

CSV Comma-Separated Value

DSCA Distribution of Sentence to the Cluster Algorithm

ECHR European Court of Human Rights

EPA Environmental Protection Agency

FCM Fuzzy c-means

GATE General Architecture for Text Engineering

IAAIL International Association for Artificial Intelligence and Law

IAT Inference Anchoring Theory

ICAIL International Conference on Artificial Intelligence for Law

ICLE International Corpus of Learning English

IDEA International Debate Education Association

JAPE Java Annotation Patterns Engine

LDA Latent Dirichlet Allocation

LIIR Language Intelligence & Information Retrieval Lab

LSTMs Long Short-Term Memory Networks

ML Machine Learning

NLP Natural Language Processing

OVA Online Visualisation of Argument

POS Part of Speech

RF Random Forest

RST Rhetorical Structure Theory

SA Stance Alignment

SMO Sequential Minimal Optimization

SRM Structural Risk Minimization

STS Semantic Text Similarity

SVM Support Vector Machine

TA Transition Application

TE Textual Entailment

TF-IDF Term Frequency-Inverse Document Frequency

UE Universidade de Évora

UKP Ubiquitous Knowledge Processing

VSM Vector Space Model

WEKA Waikato Environment for Knowledge Analysis

Sumário

A argumentação desempenha um papel fundamental na comunicação humana ao formular razões e tirar conclusões. Desenvolveu-se um sistema automático para identificar argumentos jurídicos de forma eficaz em termos de custos a partir da jurisprudência. Usando 42 leis jurídicas do Tribunal Europeu dos Direitos Humanos (ECHR), anotou-se os documentos para estabelecer um conjunto de dados "padrão-ouro".

Foi então desenvolvido e testado um processo composto por 3 etapas para mineração de argumentos. A primeira etapa foi avaliar o melhor conjunto de recursos para identificar automaticamente as frases argumentativas do texto não estruturado. Várias experiencias foram conduzidas dependendo do tipo de características disponíveis no corpus, a fim de determinar qual abordagem que produzia os melhores resultados. No segundo estágio, introduziu-se uma nova abordagem de agrupamento automático (para agrupar frases num argumento legal coerente), através da utilização de dois novos algoritmos: o "Algoritmo de Identificação do Grupo Apropriado", ACIA e a "Distribuição de orações no agrupamento de Cluster", DSCA. O trabalho inclui também um sistema de avaliação do algoritmo de agrupamento que permite ajustar o seu desempenho. Na terceira etapa do trabalho, utilizou-se uma abordagem híbrida de técnicas estatísticas e baseadas em regras para categorizar as orações argumentativas.

No geral, observa-se que o nível de precisão e utilidade alcançado por essas novas técnicas é viável como base para uma estrutura geral de argumentação e mineração.

Palavras chave: Mineração de Argumentos, Domínio legal, Aprendização Automática, SVM, Fuzzy Clustering

Abstract

Automatic Extraction and Structure of Arguments in Legal Documents

Argumentation plays a cardinal role in human communication when formulating reasons and drawing conclusions. A system to automatically identify legal arguments cost-effectively from case-law was developed. Using 42 legal case-laws from the European Court of Human Rights (ECHR), an annotation was performed to establish a 'gold-standard' dataset. Then a three-stage process for argument mining was developed and tested.

The first stage aims at evaluating the best set of features for automatically identifying argumentative sentences within unstructured text. Several experiments were conducted, depending upon the type of features available in the corpus, in order to determine which approach yielded the best result. In the second stage, a novel approach to clustering (for grouping sentences automatically into a coherent legal argument) was introduced through the development of two new algorithms: the "Appropriate Cluster Identification Algorithm",(ACIA) and the "Distribution of Sentence to the Cluster Algorithm" (DSCA). This work also includes a new evaluation system for the clustering algorithm, which helps tuning it for performance. In the third stage, a hybrid approach of statistical and rule-based techniques was used in order to categorize argumentative sentences.

Overall, it's possible to observe that the level of accuracy and usefulness achieve by these new techniques makes it viable as the basis of a general argument-mining framework.

Keywords: Argument Mining, legal domain, Machine Learning, SVM, Fuzzy Clustering

1

Introduction

1.1 Overview

Historically, Dialectics and Philosophy are the ancient roots of the discipline of argumentation. Argumentation has always been considered an important branch of Philosophy and, with the passage of time and advances in technology, its relevance has grown exponentially in other fields such as Literature, Logic, Law, Mass Communication and Artificial Intelligence. Argumentation is the fundamental tool for human beings to argue and reach their objectives, without resorting to violence. The history of argumentation starts from ancient human civilization, and ideas of notable significance in this field began with the Greek philosopher Aristotle (384 - 322 B.C)[166]. He formulated a theory called *modal logic* that consists of three modes of persuasion: 'Pathos', relating to emotions and

values; 'Ethos', representing credibility or authority; and 'Logos', the logic behind the argumentation. Aristotle's work on Logos paved the way for modern computational linguistics and artificial intelligence.

During argumentation, facts, figures, and evidence, as well as logic are provided to support, attack and/or refute the opponent's argument. Presently, when social media is one of the most important discussion platforms available, the number of users expressing their opinions has grown enormously. Usually, such opinions are expressed through a sequence of reasoning that generates ideas and claims. Professor Chris Reed[1] said that *The ability to argue, to express our reasoning to others, is one of the defining features of what it is to be human* [151]. Similarly, MacEwan [110] states that *argumentation is the process of proving or disproving the proposition. Its purpose is to induce a new belief, to establish the truth or to combat error in the mind of another*. The simplest and most concise definition of an argument seems to have been provided by Schiappa and Nordin [166], stating that an argument is a claim supported by reasons.

Consider the example of the US Presidential Debate, which takes place every four years and is one of the most popular events in the US. The winning of a debate depends upon the patterns of argument e.g. in a supportive or a hostile way. The judge is the people of the nation and their role is to analyze the debate and select their candidate of choice. Their perception and acceptance of debate seem to depend partly upon the technology used for delivery. In the 19^{th} century, the media or communication medium was a newspaper, which is much less effective compared to today's media. An example of this kind of debate is Abraham Lincoln vs. Stephen Douglas which took place on October, 16^{th} 1854 [62]. The debate was so long that they ordered a break to let the audience go home, have dinner and then return to endure four more hours of talks. This exemplifies the huge investment in time and money that leaders need to make to communicate their agenda and beliefs. At that time, newspapers or magazines were the only resources for making the debate public knowledge, which was much less effective than radio or television. The debate is not only the text that is presented, it's a mode of arguing and expression that interacts with many factors such as environment, politics/diplomacy, language and presentation. With the development of improved communication tools and techniques, the amount of material broadcasted by radio, television, voice records, etc has increased. Furthermore, within the last few decades, the world has enjoyed a significant improvement in communications technology. As a result, specific information used in the debate can be captured and presented in a structured way (e.g. by highlighting) so that people can assimilate the information more quickly and in greater depth. For instance, arguments 'for' or 'against' used in a debate can be shown in a graphical format which enhances its understandability. The reverse process, in which arguments are extracted from people's comments, helps politicians to

[1]Director of ARG-tech, Centre for Argument Technology, University of Dundee

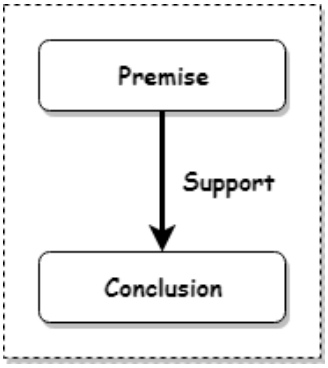

Figure 1.1: Simple Argument

understand their constituent's choices, which helps them to design effective policies. This idea of political debate is just an example of the usability and effectiveness of argumentation in professional life which is found to be similar across other sectors of society.

In the current era (21^{st} century), logic, philosophy and technology have grown collaboratively. Lippi and Torroni [106] mentioned that after the dissemination of Pollock [144], Simari and Loui [169] and Dung's [54] argumentation models move into the Intellectual technology zone. These scientists created a connection between Philosophy and Artificial Intelligence, and gave rise to a new field named *Computational Argumentation*.

There are two distinct approaches to computational argumentation: abstract and structured. Abstract argumentation [106] is represented as an atomic entity that provides a powerful framework to model and analyze the 'attack relation' between the arguments. Dung's argumentation theory [54] is the basis of abstract argumentation, details of which are explained in Section 2.1.1. On the other hand, structured argumentation deals with the internal structure of an argument divided into premise(s) and a conclusion as shown in Figure 1.1. This is an example of a simple argument (consisting of premise and conclusion). The internal structure of the argument should be identified automatically, which is one of the characteristics of argumentation (or argument) mining [106].

This most recent decade has witnessed the rapid development of argument mining in several fields (see Chapter 3). Every day, a massive number of electronic documents concerning news editorials, discussion forums and judicial decisions containing arguments are generated. During a discussion, facts, figures, and further evidence as well as logic are used to support or attack the arguments presented by an opponent. Usually, such opinions are expressed through an array of evidence to support the claim. In addition, premises are used to reinforce other premises to strengthen the focal point of the discussion/debate.

Psychology is the religion of the modern era. If people are unhappy, guilty, or confused about life, they go to see a psychologist. Last year, two million people in North America visited a psychologist because of personal and emotional problems.

Table 1.1: Sample format of Argument (adapted from [197])

A. Last year, two million people in North America visited a psychologist because of personal and emotional problems
B. If people are unhappy, guilty, or confused about life, they go to see a psychologist.
C. Psychology is the religion of the modern era.

Table 1.2: Argument Standardization (adapted from [197])

Figure 1.2: Serial Argument representing Table 1.2

Table 1.2 represents the argument Standardization transferred from the argument example presented in Table 1.1. In this situation, the argument is of a type known as 'serial'. As can be seen in Figure 1.2 (and Table 1.2), there are two premises (A and B) and one conclusion (C). The sentence 'B' has additional information for sentence 'A' to support the claim 'C' which says *Psychology is the religion of the modern era*. Thus, premises are the vehicle that supports the conclusion's reasoning and approval.

Argumentation is available everywhere, in every profession (like journalism, clinical practice, science, law and management) that uses the argument to reach their goal ultimately. During this process, appropriate argument and counter-arguments are generated, depending on discussion agendas. Thus, extracting arguments from text is a crucial but not easy task. One of the most prominent reasons is the structure of a corpus; even human beings find it difficult to detect and classify arguments. It is necessary to understand the theme of a text to understand its arguments.

Considering the simple structure of the European Court of Human Rights (ECHR) example presented in Table 1.3, two sentences belong to the same argument, as determined by the annotator. The first sentence (underlined) is the premise and the second sentence (in italics) is the conclusion. In

other words, the first sentence states the reason why the second sentence is valid. This can be made obvious when a human annotates the text, since links can be set up between arbitrary phrases in the corpus. A human annotator can draw upon context and background knowledge to see that the first sentence that contains the words "However, again" reflects the meaning being repeated, and thus link it to the second sentence.

However, again the Commission find no evidence in the case to substantiate this complaint. *It follows that this part of the application is also manifestly ill-founded within the meaning of Article 27 para. 2 (Art. 27-2) of the Convention*

Table 1.3: Example of single sentence as Argument

On the other hand, if identification or annotation responsibilities are given to an automatic classifier (i.e. an algorithm that implements classification), it needs to be able to annotate without background knowledge. Similarly, there is also the possibility of components of one argument (i.e the premise or conclusion) to be involved in another argument. In the Table 1.4 the sentence *It follows that he was not tired or punished again 'for an offense for which he has already been finally acquitted or convicted'* is the conclusion for argument number (4C) but also the premise for argument (5P). Considering the wide variation in the structure of an argument, extraction of arguments is quite challenging work.

Sentence	AS ID
The Commission finds that the applicant, who was convicted for speeding twice in the course of one journey, but over separate stretches of road, was convicted not of one but of two separate offences.	$4P$
It follows that he was not tried or punished again 'for an offence for which he has already been finally acquitted or convicted'	$4C, 5P$
This part of the application must therefore be rejected as being manifestly ill-founded within the meaning of Article 27 para. 2 (Art. 27-2) of the Convention.	$5C$

Table 1.4: Example of complex argument structure

To extract such arguments, a methodology that uses different theoretical Argumentation models, machine learning tools and techniques are proposed. Moreover, several techniques such as identifying features and preprocessing the corpus must be used, in addition to selection and application of an

appropriate machine learning algorithms, so that relevant features can be extracted to create a model. These models and tools, which will be discussed in Chapter 2, and are the basis for understanding the concept of argument.

The application of argument mining is immensely useful and its importance has already been mentioned in political debates. The technique of argument mining can be used on a variety of diverse areas so as to make more precise search engines, web applications, and social media. The software industry sector could also use this technique to capture primary issue evidence, locate an argument in the corpus, and even relate arguments used by various parties to the issue under scrutiny. Furthermore, these techniques could be used in recommendation systems by analyzing users' comments available on the web and on social media; after analyzing the users' comments, a recommendation system can deliver relevant information/product to him/her. Extracting arguments from the user's comments helps to know the negative side (i.e against) or positive side (i.e in support) of the product, and also provides other, relevant information. Its also very important to other sections such as in news editorials; newspaper editorials are essential to understanding the main theme of a paper. A newspaper not only disseminates the news but also spreads the views and the arguments of people regarding the current topic of discussion in the political and cultural literature. As the technology grows, the trend of discussion and debate rises exponentially.

1.2 Motivation

Since argumentation is an important component of human communication, its effect has increased in the current era. With the continued advancement of technology and availability of numerous news portals, blogs, forums and social media, the number of people expressing their opinion has grown dramatically. As a result, a massive amount of comment, feedback and opinion containing arguments is collected every day from various sectors: politics, journalism, clinics, judicial court, social media, etc. Due to this massive amount of information, it is difficult to get the precise and specific contents as required. In addition to this, it is tedious and time consuming to scan through the enormous amount of information available on the web. The users' needs are so diverse that some require general information while others may need more specific information. Search engines and web portal do their best to provide the relevant information, but it may not always show correct information. To overcome such issues, the search engines and web portals must be very precise and need to have a critical analyzes capability. The solution to making this a practical reality is Artificial Intelligence.

Acting is the ability of Intelligence to utilize knowledge from a source of data. In this sense,

investigating a new method or approach that can obtain new information from existing sources of data is always an interesting question to explore. Furthermore, information that holds reasoning and claims is an interesting and challenging topic, allowing us to discover the patterns inside the structure of the sentence: this is argument mining. The developed system should qualify fully as an artificial intelligence, in that exhibits the capacity to act. Here 'action' is defined as the synthesis of uniquely original knowledge from existing knowledge. Exploring novel modes of such syntheses is, needless to say, interesting and relevant [52]. Many researchers have worked aiming to identify arguments from different domains, Stab *et al.* [172, 173, 174, 175, 176, 177] who worked in annotating and identifying arguments from persuasive essays; Lawrence *et. al* [97, 98, 99] proposed a way of recognizing general arguments using a unique different approach; Ghosh *et al.* [68] identifies argumentative text portions and the nature of the argumentation from web portals such as discussion forums, blogs, and web page comments and Boltuzic and Snajder [31] conducts the investigation into argument-based opinion mining from online discussions.

Similarly, there have been huge contributions to the research of Argumentation, Law and Artificial Intelligence within the last 30 years. Researchers such as Douglas Walton [196], Trevor J M Bench-Capon [18], Paul E. Dunne [56], Henry Prakken [146], Floris Bex [23], Thomas F. Gordon [69], Bart Verheij [192], Katie Atkinson [12] investigate mostly abstract argumentation in the legal domain. In addition, Kevin Ashley [136], Raquel Mochales Palau [100], Marie-Francine Moens [123] pioneered work on argument mining in the legal domain. Their work is widely cited, but after their contribution, the legacy seems not to be continued, as it needs to be. Even though, judiciary process are one of the most refined domains of argumentation, the need for a system to identify arguments cost-effectively from case-law is rather urgent. As part of their work, lawyers and other stakeholders of the court need to know the type of arguments that have been used in previous cases, which may have set a precedent, making this important for effective, efficient and uniform judgment in a particular case. After the development of an argument mining system, lawyers, judges, agents of defenders, plaintiffs and any other stakeholders of the court will benefit greatly when searching for previous arguments. This will help court users to identify the specifically advocated type of argument and other court activities related to argumentation. Considering the relevance of argumentation in everyday life and its ubiquity in the judiciary system, there is a need for some kind of 'schema', which is able to perform analysis and detect specific arguments from previous cases. This need was the motivation for developing a framework to identify arguments from the legal domain automatically.

The results of this can be helpful to lawyers and judges in court because every single judgment contains many references to past case-law of a related nature. Hence, they form a reference for future judgments of the same kind, which is essential for the judiciary mechanism and scholars. The system should be capable of synthesizing judicial pronouncement documents so that they become

more useful and less time-consuming to consult. This natural way of accessing previous information may contribute towards the efficiency of the decision-making process, thereby reducing the time required for the perusal of judicial documents of a similar nature, which will both speed up the legal proceedings mechanism and hopefully reduce its cost. The system can also facilitate scholars during their study and research. Therefore, the primary purpose of this study is to identify and evaluate the structure of arguments present in legal documents. Furthermore, the application and techniques proposed in this thesis are not limited to legal documents, but can also be used for automating the processing of other types of plain text documents, so that this technique of argument mining can be used to develop more precise search engines, web applications, and is also benefits for large social media companies as well as other web development firms. The software industry sector could also use these techniques to capture the primary issue evidence, locate an argument in the corpus, and even relate arguments used by the various parties under scrutiny.

Thus, the main focus of this work is to develop a system that identifies and classifies the arguments present in a legal corpus. The research aims to create a system that identifies arguments through their structure. The planned approach is entirely novel and consequently in itself a major contribution to the argument-mining research field. To accomplish the task of producing dataset, case-law files from the European Court of Human Rights (ECHR)[2] annotated by Mochales-Palau and Moens [122] were selected. The corpus is composed of 42 case-law files that include 20 decisions and 22 judgments Categories. Details of the corpus are described in Chapter 4.

1.3 Problem Statement

Computational argumentation is a recent and rapidly growing area of research in Computational Linguistics. Most of the existing state-of-the-art approaches are focused on one or more of the following three consecutive subtasks:

- Identification of argumentation sentences [29, 58, 71, 105, 121, 159, 164, 172, 176],
- Location of boundaries of the argument [36, 99, 121, 172, 164]
- Determination of the argumentation structure [28, 77, 96, 139, 140, 158, 161, 195].

These tasks are dependent on each other; the results of the first module are used as input to the second module, and the results of the second module are used in turn as input to the third. There

[2] http://hudoc.echr.coe.int/sites/eng

is a clear similarity of approach that is common to most of the previous research work, which we aim to improve on in this work. As a result, new features were proposed resulting in improvements to stages 1 and 3, and a new approach devised for stage 2. The following research question will be addressed:

RQ1: Is it possible to find better-performing discrete features and to use machine learning algorithms to identify component arguments / argumentative sentences in the legal domain?

After identifying which sentences are relevant to an argument, it is necessary to bundle these sentences into arguments (defined as a set of related argumentative sentences). Hence, the second stage of the work is to identify the boundaries of the arguments, a technique extensively explored in the AI & Law literature (see: Chapter 3). To find the boundaries of an argument, the relation between the premise(s) and conclusion are identified, which indicates which premise(s) belong to the conclusion. However, there can be major disagreements over which sentences are conclusions and which are premises, largely because the conclusion in one argument can also simultaneously be the premise of another argument. As a consequence, the accuracy of finding the relation between the components (specifically, premise and conclusion) of arguments is low. Similarly, in [41] another disparity is found, that in some of the cases, the facts (i.e. non-argumentative sentences) are also considered to be components of argument. Since typical Stage 2 processing, is limited to considering only argumentative sentences as input, this is also problematic and led us to our next research interest.

RQ2: Is it possible to find a better way to group the argumentative sentence into arguments, rather than the commonly used method of grouping by detecting argument boundaries?

There are very few studies of Computational Argumentation in the legal domain. One of the pioneering works was from Mochales and Moens [121]. They proposed specific Context-free Grammars (CFGs) to detect the argument structure, which was applied to a very limited portion of case-law. Unfortunately, the accuracy of their data was comparatively low, perhaps accounting for the subsequent dearth of further investigation into argument mining in the legal domain. On the other hand, production of case-law is snowballing, which demands a system that is able to analyse and detect specific arguments from previous cases. Moreover, the components of one argument can also simultaneously be a component of another argument making the complexity of the structures required to model an argument also high. To deal with this situation, a classifier that identifies arguments automatically is needed. This, in turn, required addressing the following question.

RQ3: "Is it possible to create a system that identifies arguments through their structure obtainable from legal documents?"

In summary, developing and evaluating a novel approach to argument mining, which will replace the boundary technique with newer, emerging technology, and allow automated processing of legal arguments, in a more accurate and sustainable way, is one of the focal points of the work.

1.4 Objectives

The main goal of this research is to develop a system that is able to identify arguments from legal documents automatically. This goal includes several other sub-goals:

- **Annotate the ECHR Corpus:** Since the ECHR Corpus is not available electronically, one of the subtasks was to annotate the components of an argument (i.e., as premise or conclusion) in the ECHR case-law to form a 'gold-standard' dataset.

- **Find an appropriate Classifier to identify legal sentences:** The structure of any corpus varies depending upon the subject matter; simultaneously most of the corpora on a given subject tend to be quite similar in structure, so there is an advantage of using a domain-specific classifier. Since the selection of a domain-specific classifier is of such importance, it raises the secondary objective of selecting the most promising technology for automatically classifying legal documents, which led us to look at machine learning approaches.

- **Propose and test an argument clustering system:** As noted in the problem statement, there are various limitations with the current boundary detection technique, and a proposal to develop a technology that will automatically bundle sentences into arguments is made.

- **Find Discriminant Features:** Features are an important representation of text input to the classifier. The quality of features greatly affects whether the models will perform well or not. Usually, features are represented numerically, but there are also relevant features that are represented by more complex structures such as trees or graphs. As there are numerous ways to represent features, selecting the right features for this task is one of the most important actions to perform. Therefore another goal is to identify the discriminant features that help to determine the components (i.e. premise and conclusion) of arguments from a narrative, legal text.

1.5 Research Methodology

The primary goal of the investigation is to develop a system that identifies arguments from legal documents. This task is divided into two: Task One (1) transferring the printed European Court of Human Rights (ECHR) to a digital version and Task two (2) is the major work on identifying arguments within legal documents. Let's briefly describe each of these divisions.

Task One: Development of the ECHR Corpus This phase was about developing an ECHR Corpus. The main task was to transfer the printed ECHR Corpus to an electronic version. Preprocessing was performed to render the corpus into a standard format. The main purpose of doing this was to enable the classifier to develop a model such that it identifies information with greater accuracy. Details of the annotation are explained in Chapter 4. Here, the steps to accomplished the task are shown.

1. Case-law documents were preprocessed by applying several types of techniques to improve the quality of the corpus. First, the initial section of the case-law which consists of the name of stakeholders, the index numbers and section title are removed.

2. The case-law text is split into sentences.

3. Annotation is performed on the sentences using five Categories:

    - Case-law File Number: The ID number of the case-law file
    - Case-law Type: Distinguish the sentence as either Judgment case-law or Decision case-law type
    - Section Type (Other and Law): sentence position considerations
    - Sentence Number: Sentences are sequentially numbered in each case-law file
    - Argumentation (YES/NO): Sentences are tagged as argumentative or not.
    - Argument Component Notation: The components of the arguments are noted with 'C' for conclusion, 'P' for premise along with the argument number.

Task Two: Identify arguments from legal documents The proposed methodology is divided into three modules: Argumentative Element Identifier, Argument Builder, and Argument Structurer. The working procedure of each of these modules is described in following sub-sections.

1.5.1 Argument Element Identifier

The first module of the work is to identify a set of discriminating features that will help to classify argumentative and non-argumentative sentences. The following steps show the methodology applied.

1. Extract Structural, Contextual, Syntactic and Lexical features from the Corpus
2. Calculate numeric values for each feature using the TF-IDF measure.
3. Research the effects on performance of top-ranked features and parameters on the classifier. Selects the best features by using the Gain Ratio approach [149].
4. Determine experimentally which is the best machine learning algorithm for classifying argument sentences from the legal documents.
5. Perform Tree Kernel Experiments in the SVM light tool using Syntactic parser features.
6. Compare performance results via statistical tests (Paired T-Test)
7. Compare results from all experiments and select the best approach to identify argumentative sentences.

1.5.2 Argument Builder

After identifying which sentences contribute to the argument, it is necessary first to organize these sentences into an argument (i.e. a set of related argumentative sentences). A clustering technique (discussed in Chapters 5 and 6) is proposed that gathers argumentative sentences into a cluster of potential arguments. The following steps show the methodology applied to fulfill the aim of this module:

1. Extract n-gram (word, character), Word2vec, 'Sentence Closeness' and 'Combine Feature' (Combining all these three features) from the corpus.
2. Identify the optimum number of arguments in the case-law file using the work of Xie and Beni [205] and Latent Dirichlet Allocation (LDA) [30] exploited by Juan *et al.* [38] techniques.
3. Since the same argumentative sentence can be shared by several arguments, the Fuzzy clustering algorithm is used to provide a membership value ranging from 0 to 1 for each cluster. The value is calculated from the features provided.

4. A "Distribution of Sentence to the Cluster Algorithm" (DSCA) is developed to transfer the above mentioned membership value to cluster form (transforming soft clustering to hard clustering).

5. An "Appropriate Cluster Identification Algorithm" (ACIA) algorithm is developed to find the best mapping between the system's clusters and the gold-standard dataset clusters. The algorithm maps the argument predicted to the one that is closest to the gold-standard.

6. System Evaluation is then performed by comparing the arguments obtained from the ACIA recommendation with arguments from the gold-standard datasets.

7. The results obtained from argument clustering are compared and analyzed, and its limitations were discussed.

1.5.3 Argument Structurer

The Argument Structurer (AS) module deals with the discovery of the internal structure of the arguments; i.e., their identification as either premise or conclusion. The following steps show the methodology applied.

1. The argumentative sentences are divided into three Categories:
   - **Single Annotation:** a sentence annotated as either a premise or conclusion.
   - **Overlap Annotation:** sentences annotated as premise/conclusion of one argument and also premise/conclusion of another argument.
   - **Sentence Partition:** sentences consisting of more than one argument component.
2. The task is divided into two phases: A classification-based approach and a rule-based approach.
3. The classification-based approach is applied to classify sentences into Single Annotation, Overlap Annotation, and Sentence Partition.
4. The rule-based approach is then applied to determine the accuracy of the discourse indicators that separated the components of the arguments in Partition Categories.
5. In this final step, the performance yielded by the discourse markers which separate the component of the arguments in a sentence were compared.

1.6 Key Contributions

The main goal is to develop a system that identifies and classifies legal arguments present in a legal corpus. The main achievements of the research are:

1. Annotation of the ECHR corpus. The task was to transfer the printed manually annotated ECHR corpus into an electronic version.

2. Find the most viable and best-suited machine learning algorithm to detect arguments sourced from legal documents.

3. Find discriminant features which signal the components of arguments (i.e., premise and conclusion) in the legal domain.

4. A technique to cluster argumentative sentences to form an argument. This approach is a novel one in the field of argument mining. Prior to this work, most of the research tried to identify the boundaries of the argument by finding a relation (i.e. 'support' or 'attack') between the components of an argument, and between the arguments themselves.

5. Development of the "Distribution of Sentence to the Cluster Algorithm" (DSCA) that transfers membership values generated from a fuzzy cluster algorithm into the form required for clustering form (i.e., hard clustering).

6. Development of the "Appropriate Cluster Identification Algorithm" (ACIA) algorithm to find the best mapping to the gold-standard datasets.

7. Development of an evaluation system to measure the performance of the proposed system.

8. The system is capable of processing judicial pronouncement documents so that they can become more useful and less time-consuming to work with. This natural way of accessing prior cases will contribute towards the efficiency of the decision-making process, thereby reducing the preparation time required to search existing case-law and thus speeding up legal proceedings.

9. The result from this study can be helpful to lawyers and judges of the court because every single judgment contains many references to relevant past case-law. They also become reference material for future judgments, making them essential to judiciary scholars. This system will help professionals to analyze cases in greater depth; as a consequence, it is hoped that the quality of judicial decisions will improve.

1.7 Publications

Prakash Poudyal, Teresa Gonçalves, Paulo Quaresma "**Using clustering techniques to identify arguments in legal documents**", Twelfth International Workshop on Juris-informatics (JURISIN 2018), November 12 - 13, 2018 Raiosha, Hiyoshi Campus in Keio University, Yokohama, Japan

Prakash Poudyal, Teresa Gonçalves, Paulo Quaresma "**An architecture for the automatic identification of arguments in legal documents**", The 2^{nd} International Workshop on Methodologies for Research on Legal Argumentation (MET-ARG) at WAW 2018 Warsaw Argumentation Week September 15^{th}, 2018, Warsaw

Prakash Poudyal, Teresa Gonçalves, Paulo Quaresma "**Experiments On Identification of Argumentative Sentences**", 2016 10^{th} International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, China, 2016.

Prakash Poudyal "**Automatic Extraction and Structure of Arguments in Legal Documents**", Proceedings of the Second Summer School of Argumentation: Computational and Linguistic Perspectives (SSA '16), CoRR, abs/1608.02441

Prakash Poudyal "**A Machine Learning Approach to Argument Mining in Legal Documents**" VI Workshop on Artificial Intelligence and the Complexity of Legal Systems (AICOL), JURIX 2015, 9th-11th December 2015, University of Minho, Braga Portugal, pages 443–450. Springer, (**Best Paper in Doctoral Consortium**)

Prakash Poudyal, Teresa Gonçalves, Paulo Quaresma "**The influence of training data on the performance of the SVMCMM algorithm**" ,JIUE'2014 - Jornadas de Informtica da Universidade de Evora, Universidade de Evora, ISBN 978-989-97060-2-6, February 2014.

1.8 Organization of the thesis

The thesis is divided into seven (7) chapters. Chapter 2 presents an overview of the theoretical concepts that are needed to understand the issues addressed in this thesis. Chapter 3 provides a review of state of the art that is relevant to address the proposed issues. Chapter 4 provides a description of the ECHR Corpus, including its historical background and describes the process of annotation. Chapter 5 presents the architecture of the proposed method of this thesis. Chapter 6

presents the experimental results and analysis of the architecture. Finally, Chapter 7 discusses what has been achieved and presents some ideas for extension of this thesis in the future.

1.9 Summary

The main goal of this document is to provide detailed information about the approach developed to handle an existing problem in the legal domain: argument mining and structuring. The chapter describes the general overview of the work and the interconnection between artificial intelligence law and argument theory. In particular, the chapter presents the research questions and their associated objectives, a brief summary of the research methodology and the main contributions to this field of research.

2

Theoretical Concepts

This chapter describes fundamental concepts, tools, and techniques used in this thesis. These concepts will help the reader to understand why argument mining is important and to become acquainted to the approaches used in computational argumentation research. Furthermore, the information available in this chapter should also be useful as a reference document for the researcher who is interested in computational argumentation. The chapter is divided into five (5) sections. In section 2.1, Argumentation Models, Argumentation Standardization, Argumentation Diagram, Argumentation Quality, Argumentation Schemes and Argument Interchange Format, Argument Visualization and analysis tools are described; section 2.2 presents different preprocessing techniques and theoretical aspects used to represent text through numeric values; section 2.3 discusses various machine learning algorithms and tools; section 2.5 introduces the evaluation measuring units and finally, section 2.6 concludes the chapter by discussing statistical analysis tools.

2.1 Argumentation Theory

Argumentation theory is a rich interdisciplinary field that uses concepts from philosophy, law, communication, psychology and artificial intelligence. The notion of argumentation is found from Aristotle's [180] works onwards. His work on traditional logic rhetoric and dialectic plays a significant role to establish the foundation of the computational argumentation. In the 21^{st} century, logic, philosophy, and technology have a created a symbiotic relationship. With the passage of time and its rapid development technology importance has grown exponentially in fields such as literature, mass, communication, logic, law and Artificial Intelligence [86].

Before going through theories and the fundamental classical principle of argumentation, let's recall the definition of Argument given by the founders of the argumentation. Ketcham [93] defines *argumentation as the art of persuading others to think or act in a definite way. It includes all writing and speaking which is persuasive in form.* Fox *et. al* [64] said that arguments could be considered as the tentative proofs of the proposition. Overall, the consensus appears to be that argument is all about claim with reasoning.

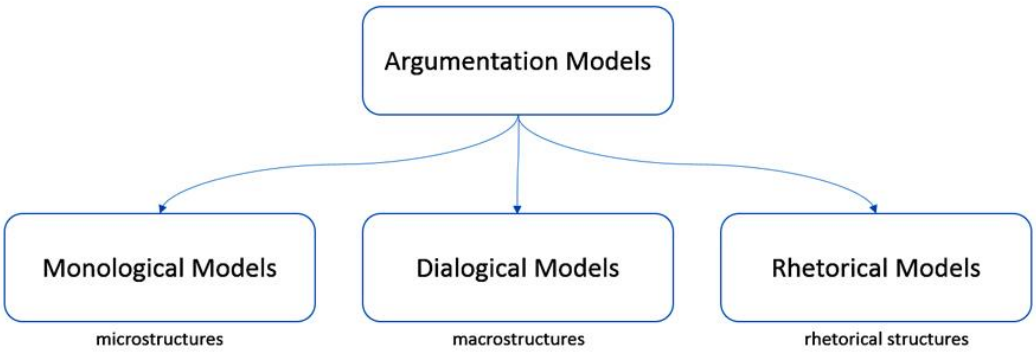

Figure 2.1: Taxonomy of Argumentation Model (adapted from [112])

2.1.1 Argumentation Models

Bentahar *et. al* [19] proposed a taxonomy of models that is horizontally divided into three categories as shown in Figure 2.1: monological models (also called micro-level models), dialogical models (macro-level models) and rhetorical models. These models are used in different domains such as legal reasoning, multi-agent systems, and natural language communication. The three models are discussed below.

Monological model

Monological models focus on the internal structure of an argument. Their role is to define the function of argument components, creating relations or links between them, and also to define the type of reasoning being used. As an example, Toulmin's model [184] falls into the Monological model category because it defines six argument components. Similarly, Walton *et. al* [198] classify elements of an argument into premises and a conclusion, and add inferences from the premises to the conclusion. Since arguments rely on reasons, coupled with inference, in order to substantiate a claim for approval, it is important to understand how the components of arguments are interlinked or combined, and how the inferential process works. The environment also influences the reasoning when arguing that a claim should be approved. These things are all used as indicators of the strength of an argument in the monological model.

Toulmin's Argument Model

Stephen Toulmin's method [184] is an informal method of reasoning that moves from statement of evidence to a conclusion. Two argument components (premise and conclusion) are used in this work, but Toulmin defines six (produced directly from the book 'Elements of Argumentation'. [22]), namely:

- *Facts:* Facts are items of information that are specific to a given context. (e.g. the 'name' and the 'age' of a patient).

- *Warrant:* A Warrant is the information that is used to decide if a claim is qualified. A claim is qualified if the Warrants holds and the rebuttal does not.

- *Backing:* Backing is the support for the Warrant. It provides explanations and reasons for accepting the claim. The reasons might be drawn from diverse areas such as ethics, morals, attitudes, authorities or law.

- *Rebuttal:* The Rebuttal is the counter agent of the 'Warrant'. It lists the reasons the warrant might be inapplicable and presents a counter-argument to the Warrant.

- *Qualified claims:* Qualification is the conclusion after weighing the Backing against the Rebuttal, to see which one has the most merit. If the judgment goes in favor of Backing, the claim is considered to be Qualified.

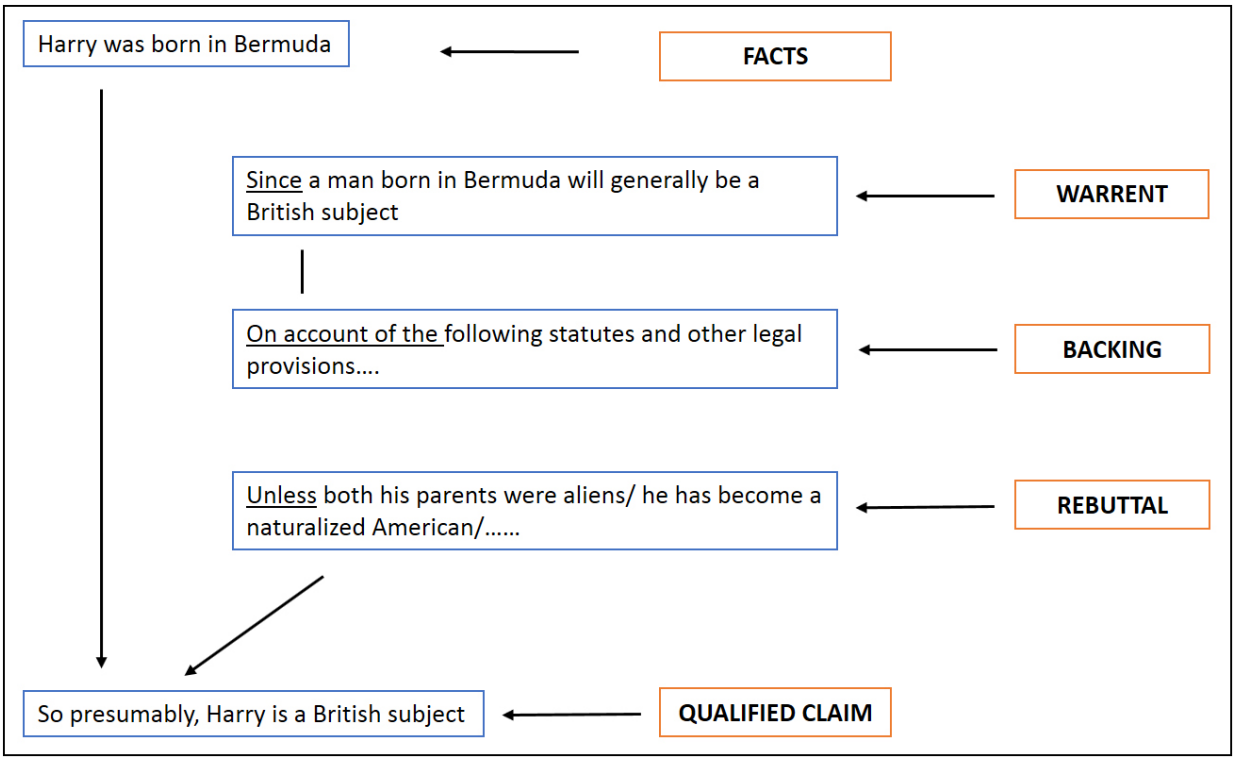

Figure 2.2: Toulmin's Argument Model (adapted from [22])

Figure 2.2 an example of Toulmin's model and is produced directly from [46]. The main point of this particular argument is to determine the citizenship of Harry. The example claims that *Harry is a British citizen* because he was born in Bermuda (Fact). The 'Warrant' which supports the 'Claim' is that *Since a man born in Bermuda will generally be a British subject*. The 'Backing' is this case would be any legal documents e.g. birth certificate which support the 'Claim', and the 'Rebuttal', which is a counter agent to the 'Warrants', is that Harry's parents were considered to be aliens.

It was found that Toulmin's model is not only used in the computational linguistic but also in other domains such as to identify argument patterns. This model was used by Chambliss *et.al* [45, 80] to investigate 20 documents in a classroom setting to find the components of arguments and argument patterns, and by Simosi [170] to solve conflicts, as well as by other researchers in a computer-supported collaborative learning research [60, 66, 178, 200].

Bench-Capon [16, 80] introduced the 'presupposition component' as an additional component to Toulmin's structure. It addresses the construction of textual arguments which form the basis for a link between monological and dialogical models. Dunn [55] and Simosi [170] claim that the Toulmin model can be applied in general (traditional) argumentation, however, Freeman [65] criticized the use of the Toulmin's model to represent the arguments that appear in real-world scenarios.

Dialogical Models

The arguments generated during the dialog process are categorized as dialogical models where the internal structure of the argument is ignored. In these models, arguments are represented as abstract entities. Several dialogical models are presented by Dung [54], Bentahar *et. al* [20, 21], Atkinson *et. al* [13], and Hamblin [82] and Mackenzie [111].

Argumentation Frameworks

Dung's Framework is the most popular theory in modern computational argumentation. He argues that arguments are atomic and that there are attacking relations between arguments. The theory consists of a set of arguments AR with a binary relation R_{att}, described as

$$AF =< AR, R_{att} >$$

where

$$R_{att} \subseteq AR \times AR.$$

Given $(X, Y) \in R_{att}$, it can be said that argument X attacks argument Y.

To demonstrate the Dung's Argument Frameworks, let's take an example of US Election 2017 debate[1]:

X= `Mrs. Clinton poked at Mr. Trump by saying he believed that climate change was a hoax.`

Y= "`I do not say that, I do not say that`" Trump replied.

This example shows that the argument is presented as a dialog. Both parties argue by attacking the other. Sentence X states that Mrs. Hillary Clinton says that Donald Trump is claiming that climate change is just a hoax. Donald Trump replies by disagreeing with the argument made by Mrs. Clinton. This kind of attack relation can be represented through Dung's abstract framework. However, there are other examples of argument where a 'supportive' nature is also found between arguments; such arguments are not handled in Dung's framework. From the same selection.

"`I made a mistake using a private email`," Mrs. Clinton said.

"`That's for sure`," Mr. Trump said.

This example shows the supportive relation from one argument to another. To accommodate this

[1] https://www.nytimes.com/2016/09/27/us/politics/presidential-debate.html

kind of argument, Amgoud *et al.* [9] proposed a **Bipolar argument framework** where 'support' is also appended. In this case, the Argumentation Frameworks (AFs), consisting of a set of arguments AR with a binary relation R_{att} is described as

$$AF =< AR, R_{att}, R_{sup} >$$

where $R_{att} \subseteq AR \times AR$ and $R_{sup} \subseteq AR \times AR$.

such that $(A, B) \in R_{att}$ means that argument A attacks argument B and $(A, B) \in R_{sup}$ means that a argument A supports argument B.

Rhetorical Models

Rhetorical Structure Theory (RST) [34] is the theory describing the organization of natural language text. It characterizes its structure through the relation between parts of the text. 'Metamorphic' terms such as 'nucleus' and 'satellite' are used to assign figurative meanings in the text. The 'nucleus' represents the centric information while 'satellite' represents additional information needed by the nucleus. The relation between them is defined through four fields.

1. Constraints on the Nucleus,
2. Constraints on the Satellite,
3. Constraints on the combination of Nucleus and Satellite,
4. The Effect

Azar refers [14] "RST provides a set of the writer's intentions and the conditions which enable the reader or analyst to identify those intentions. The task of the analyst is to break the text down into text spans and to find a RST relation that connects each pair of spans until all pairs are accounted for. To determine whether or not a relation holds between two spans of text, the analyst examines whether the constraints on the nucleus and satellite hold and if it is plausible that the writer's point has the desired effect on the reader."

2.1.2 Argumentation Standardization

Arguments that appear in the text are often not structured. People who argue through speaking or writing may change the order of evidence and claim. Sometimes, they say the conclusion and then offer clarification, which could be a premise. The arguments are not in any agreed-upon standard format which makes processing them difficult. Logically, it is believed that the premise must always come before the conclusion, but there are exceptions. To address this issue, 'Argumentation Standardization' is proposed to bring the argument text into one standard form as shown in Table 2.1. It is set of premises and a conclusion in a simple standard format, that is:

Premise 1
Premise 2
Premise N
Therefore,
Conclusion

Table 2.1: Argument Standardization Format

Creating this structure is known as *Standardizing an argument*. The standardization helps to identify the conclusion, premise, and discourse marker words. Here is an example of creating a standardization, adapted from the book *A practical study of Argument (p 23)* [72].

> `It is a mistake to think that medical problems can be treated solely by medication. First, medication does not address psychological and lifestyle issues. Also, second, medication often has side effects.`

Logically, while it is considered that conclusion comes after the premises (and it is so, in most of the cases), this does not always happen, as shown in the above example, where a conclusion is stated before the premises. The standardization becomes:

1. `Medication doesn't address psychological and lifestyle issues`
2. `Medication often has side effects`

`Therefore,`

3. `Medical problems cannot be treated solely by medication.`

The order of the sentences has been changed and the phrases themselves also changed in both

premise and conclusion to make efficient writing and understanding. This technique helps to separate the premise and conclusion.

Besides increasing comprehensibility, the internal structure of the argument. The weakness or strength of an argument is measured on the basis of these components. Additionally, this technique helps to find gaps or problems in the argument and to identify the conclusion, premise and discourse marker words [197].

Argumentation Standardization is also called Argument reconstruction. Emeren and Grootendorst [186, 187, 188] proposed four steps to reconstructing an argument. These are deletion, addition, substitution, and permutation.

Deletion: Remove unnecessary information such as non-argumentative terminologies associated with the argumentative phrase. In the example above (medication standardization), it is necessary to remove `it is a mistake to think that....` from the sentence to make it standard. As a second example, during the conversion, or debate, one party may ask for water to drink, which is not relevant to the discussion and should be removed.

Addition: Information is present explicitly as well as implicitly. To extract the explicit information, some discourse techniques are applied, but in case of implicit information it may be necessary to add some background information by an addition of text.

Substitution: In linguistics, different words are used to express the same meaning. In a sentence, for example, nouns can be represented by pronouns but for machine, these two terminologies are different even after giving the same meaning. For example in ‘‘`John likes to study in Portugal because the Portuguese language is his favorite language.`" `In this example, the proper noun` “`John`" `and the pronoun` “`his`" refers to the same person, but technically, these two words are different, therefore, such occurrence of co-reference needs to be utilized to substitute the phrase as necessary [113].

Permutation: The discourse text plays an important role in the structure of the argument. Permutation rearranges these discourse texts to highlight their relevance to the resolution process.

2.1.3 Argumentation Diagram

Diagrams are important to understand the structure and the relation between the components of the argument. The representation of arguments in a diagram helps to demonstrate the concept of arguments visually. Arguments are represented through nodes and arrows point towards the conclusion.

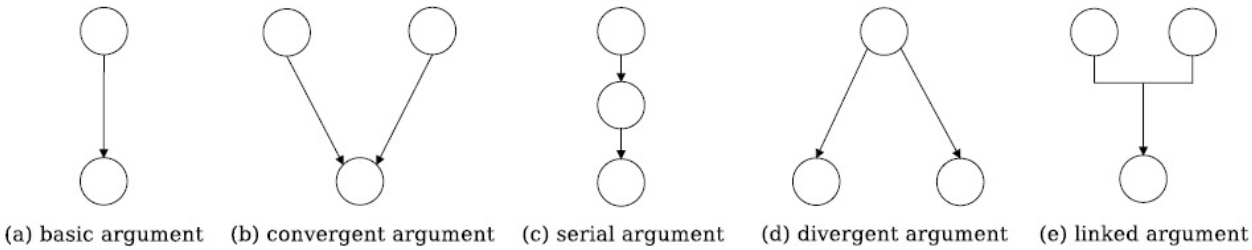

Figure 2.3: Example of argument diagram (adapted from [177])

There are altogether five types of diagrams as shown in Figure 2.3. The first is shown in Figure 2.3(a) the premise and conclusion relation which is very basic and general. The second Figure 2.3(b) is a convergent argument in which, more than one premise supports the single conclusion; in the case of a serial argument (Figure 2.3(c)), one premise support another premise which then supports the conclusion; Figure 2.3(d) is the divergent argument in which one single premise support more than one conclusion; finally in Figure 2.3(e) two premises together support the conclusion.

Let's consider the example presented section in 2.1.2 [72]. According to the structure, the argument is convergent. In the diagram premises A and B are represented as nodes connected to conclusion node C as shown in the Figure 2.4.

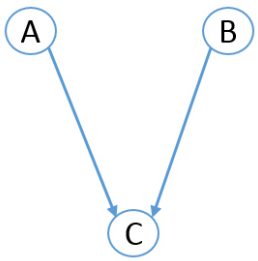

Figure 2.4: Convergent Argument [184]

2.1.4 Argumentation Quality

Quality of the arguments deals with the strength of arguments that depend upon the convincing power of one party by another. It is a matter of level of trust, developed from the argument presented. The arguments are generated from many sources (environment, presentation quality, social impact, trust, emotions) and whatever the criteria, the main goal is to approve the claim [90]. Logically, the quality of the argument can be divided into two distinct perspectives: formal logic and informal logic. Formal logic [49] creates the relation between the components of the arguments (i.e. premise and conclusion) whereas informal logic deals with evaluating arguments in everyday discourse [76, 177]. On the basis of the formal logical approach, [177] there are two ways of measuring the quality of the argument; these are Inductive Reasoning and Deductive Reasoning.

An inductive argument generalizes from the past cases to future ones and is an argument that consists of a strong reason to consider a conclusion to be true, although it is not a guaranteed outcome (i.e even though the premise is correct, the conclusion is not necessarily true). Therefore, an inductive argument's success is measured through a degree of probability or 'strength'.

For example, in the sentence

Every time I drive, I never have an accident. So, the next time I drive, I will not have an accident.

The argument about a road accident that has never happened and so can be considered as a 'strong' argument.

Deductive reasoning works out the general rule and procedure where a conclusion is guaranteed according to the evidence. In this category, arguments are assumed as either valid or invalid. There is no intermediate option between valid and invalid, such as a certain portion being valid while the rest is invalid. If premises are valid then the argument is said to be sound [203]. By this reasoning, if the evidence is true then the conclusion is also true. Take a mathematical example

If x=2 and y=3 then x+4y = 14

In this example, x+4y must equal to 14.

Similarly for,

X is Y Y is Z Therefore, X is Z

Let's consider another example on the basis of premise and conclusion.

Premise: *Red meat has cholesterol in it*

Premise: *Mutton is red meat*
Conclusion: *So, Mutton has cholesterol in it.*

The first premise declares that red meat has cholesterol; the second premise provides specific information that mutton (goat meat) is red meat. From these two premises, the third sentence is derived making it a conclusion. The first premise is a general statement, while the second premise refers to a specific case. The conclusion says that mutton has cholesterol in it, which is an inherent property of red meat for the purpose of the argument. This deductive argument is also valid, which means that the conclusion necessarily follows from the premises. Thus, in a valid deductive argument if the premise is true then the conclusion is also true.

2.1.5 Argumentation Schemes

According to Douglas [199] argumentation schemes are a stereotypical pattern of defeasible reasoning that occurs in every argument. He also defines *Argumentation Schemes are forms of argument (Structures of inference) that represent structures of common types of arguments used in everyday discourse, as well as in special contexts like those of legal argumentation and scientific argumentation.* Argumentation schemes are the forms of inference from premises to a conclusion. In 1963, Hastings started the concept of an argumentation scheme, being the first to develop a modern taxonomy of argumentation scheme [25]. He proposed raising critical questions to be asked, such that the corresponding argumentation scheme would answer them.

After 1969 [109] other like researchers Perelman and Wilkinson [201], Toulmin *et. al* [185],Eemeren and Kruiger [59], Kienpointner [94] and Grennan [74] followed the track of Hastings. In 1996, Douglas Walton presented 26 complex argumentation schemes [197]. The most common schemes are:

- Argumentation scheme for argument from position to know
- Argumentation scheme for appeal to expert opinion
- Argumentation scheme for appeal to popular opinion
- Argumentation scheme for argument from analogy
- Argumentation scheme for argument from correlation to cause
- Argumentation scheme for argument from positive consequences

- Argumentation scheme for argument from negative consequences
- Argumentation scheme for the slippery slope argument
- Argumentation scheme for argument from sign
- Argumentation scheme for argument from commitment
- Argumentation scheme for argument from inconsistent commitment
- Argumentation scheme for the direct *ad hominem* argument
- Argumentation scheme for the circumstantial *ad hominem* argument
- Argumentation scheme for argument from verbal classification

2.1.6 Argument Interchange Format

Carlos *et al.* [39] proposed the development of an Argument Interchange Format (AIF). Its main goal is to support the interchange of ideas and data between different projects and applications in the area of computational argumentation [80]. The idea of an AIF started after the development of theoretical concepts of argumentation logic and dialogic since there was no standard notation for argumentation and argument [150]. There are several tools available such as Compendium [87], Claimmaker [33], Argument Markup Language (AML) [153] and Araucaria system [153]. These tools were designed for the specific purpose of facilitating argument visualization rather than providing facilities of interoperability of arguments. Furthermore, these tools were not designed to process formal logical statement within a multi-agent system. To overcome this situation, the group of computational argument researchers set up a workshop called the 'AgentLink Technical Forum Group meeting' in Budapest, Hungary in September 2005 to draft the blueprint of the AIF [47]. The main goal of the AIF is to facilitate the development of a multi-agent system capable of argumentation-based reasoning and interaction; another goal is to facilitate the data interchange with the argument-based tools.

The Argument Interchange Format also documents the protocols for communication in a context, including how an interaction between multiple participants should proceed, and the effect of the context in which such an exchange takes place. The AIF uses directed graphs to represent arguments. Each node in the graph can be of one of two types: Information node (I-node) and Scheme node (S-node). Arguments that contain propositional information are represented by Information nodes (conclusion, premise, and data); Scheme nodes are the schema that depends upon the domain-independent patterns of reasoning. There are three kinds of scheme nodes: the rule of inference

application nodes (or RA-nodes), preference application nodes (or PA-nodes) and conflict application nodes (or CA-nodes).

2.1.7 Argument Visualization and Analysis Tools

Argument Visualization and Analysis Tools are developed for analyzing and visualizing the structure of arguments, like the inter-connectivity between arguments or within the arguments (i.e. connectivity between the components of argument). Being able to display the structure of an argument visually helps people to assimilate the substance of an argument quickly. The ways in which arguments are associated with each other can be highlighted. For example, nodes that are used to make a connection between the components of an argument can be annotated using different colors, e.g. Support argument are highlighted in green, while counter arguments are shown in red, which helps to grasp the structure very quickly and easily. There are many diagramming tools that can be used for processing arguments; next, the most cited ones are presented here:

OVA [24] is an acronym for 'Online Visualization of Argument' and is a mining tool developed in the ARG-tech, Centre for Argument Technology[2], University of Dundee, Scotland. OVA[3] is a native Argument Web tool dedicated to provide an institutional interface for the analysis of argumentation. This tool is a browser-based application that has a drag and drop interface for analyzing and visualizing arguments. The back-end engine of the system uses 'Inference Anchoring Theory' (IAT).

AIFdb [97] is a storage of argument data which is managed by the Arg-tech Centre[4], University of Dundee, Scotland. It consists of a wide range of web service interfaces. The developer divides the interface of AIFdb into two phases, a low level phase which provides the basic components of AIF argument such as nodes, edges, and schemes and a high level phase, which supports import and export features from other modules such as SVG, DOT, RDF-XML. The tool is also associated with other vendor products such as those of Carneades [70], Rationale [67] and Aracuria [153].

The Rationale [189] is an online argument mapping tool, developed for visualizing and representing the logical structure of an argument. It supports rapid building, modifying, viewing and sharing the diagrams. The software is developed at the University of Melbourne under the leadership of Professor Tim Van Gelber. Prior to Rationale, they also developed argument mapping software called Reason!able [67].

[2]http://http://www.arg-tech.org/
[3]http://www.arg.dundee.ac.uk/index.php/ova/
[4]http://http://www.arg-tech.org/

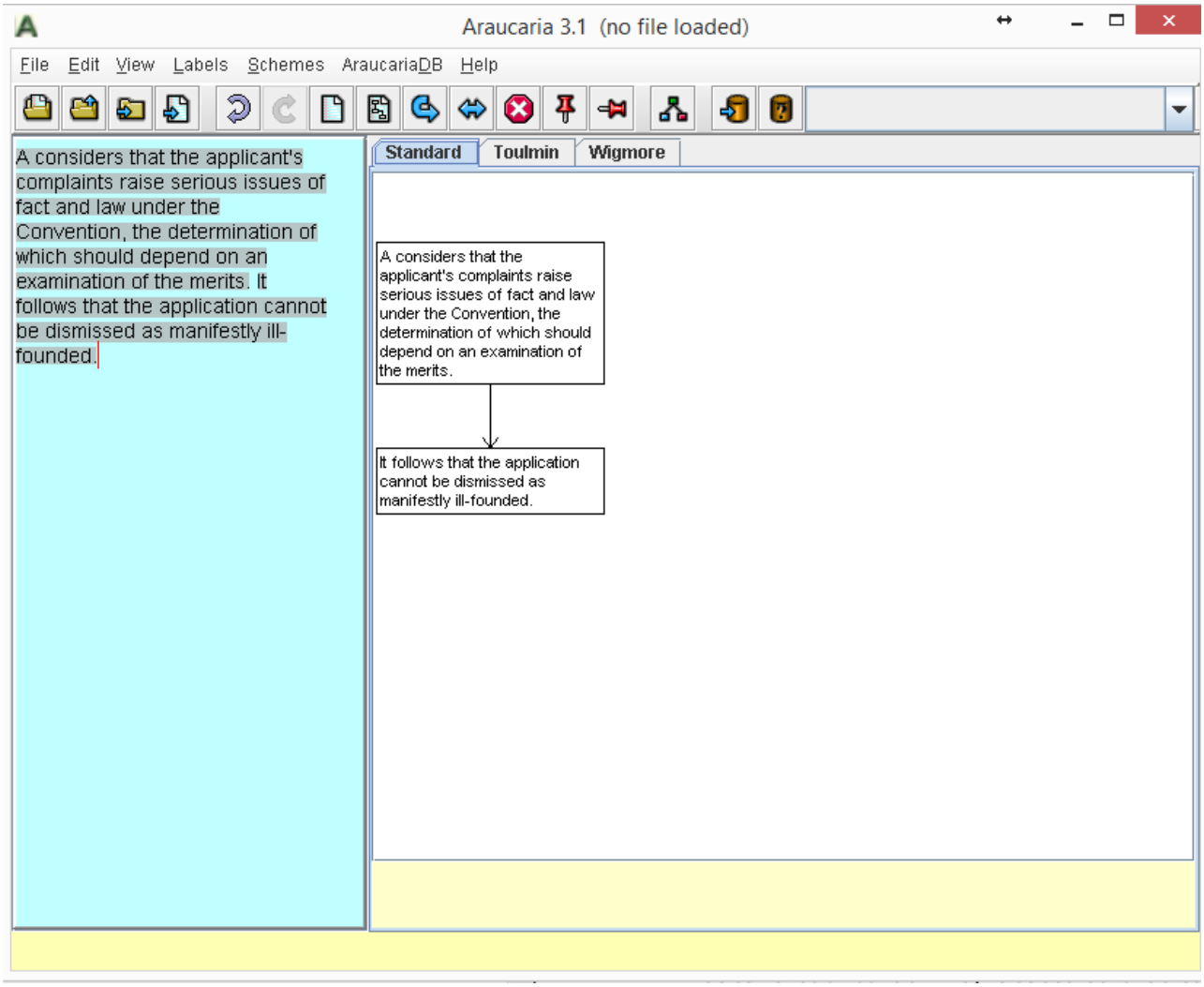

Figure 2.5: Diagram representation in ECHR argument in Araucaria [153]

Araucaria [153] is a tool for visualizing an argument which is developed on the basis of 'Argument Markup Language'. The tool supports convergent as well as linked arguments, enthymemes (missing premises) and refutation. Nodes are connected with arrows to represent inferences. Chris Reed and Glenn Rowe developed the application at the Arg-Tech, Centre in the University of Dundee. Scheuer *et al.* [165] recommended that the Araucaria tool be added to the syllabus so that students would have the ability to develop critical thinking, be familiar with the concepts of argument and also the relationships between arguments. Let's take an example of ECHR Argument and illustrate it in the Araucaria System:

> It considers that the applicant's complaints raise serious issues of fact and law under the Convention, the determination of which should depend on an examination of the merits. *It follows that the application cannot be dismissed as manifestly ill-founded.*

The argument has two sentences; the first is a premise and second is a conclusion. In Araucaria, the two nodes are connected with an arrow as shown in Figure 2.5. The working procedure of transferring the argument text to the diagram is made easy and user-friendly. The text is first uploaded to the left frame of the screen as shown in the figure. Then, a portion of the text (premise and conclusion) is selected and joined via assertions to another phrase of the sentence, which might be the premise or the conclusion.

2.2 Text Mining

Text Mining is about discovering knowledge from text. It is a process of identifying patterns to create a model that can be applied to new data to extract specific information. The general approaches for text mining are broadly categorized as rule-based approaches and statistical based approaches. In the rule-based approach, rules are applied to extract information; one of the most popular examples is an 'if then else' rule with regular expressions. In this case, the user must have in-depth knowledge of the domain so that he can write rules for a regular expression to capture entities from the document. A statistical approach (described in Section 2.3) aims to create a classifier, trained with data to generate a model. After that, the model is provided with real-world data to predict which class it belongs to. The statistical based approach does not need to be a classification problem. There other kinds of problems that can be tackled by the statistic based approach.

Next, the text processing techniques used in computational linguistics to help to structure texts are described.

2.2.1 Preprocessing

To treat a text computationally, the text should be structured. To transform the text from unstructured to structured, there are several tasks to fulfill: one of them is 'Tokenization' which is the splitting of texts into words, phrases or other meaningful units as necessary. Preprocessing is divided into two categories: 'low preprocessing' and 'high preprocessing'. Low preprocessing [42] deals with sentence boundary detection, location within the document, part of speech tagging and noun phrase chunking. High-level processing is semantic level processing such as name entity recognition, relation extraction, and temporal extraction. Several data mining tools help perform the standard way of preprocessing. Examples are WEKA [81] (Waikato Environment for Knowledge Analysis), GATE [50] (General Architecture for Text Engineering), Stanford NLP Toolkit [113].

Filtering There are various kinds of words available in a sentence, some of them have no significant role. This category or words includes common words (prepositions, conjunctions), section titles, numeric values that are represented as paragraphs or bullet numbers. These elements need to be removed, so that precise and accurate features can be extracted.

2.2.2 Sentence Representation

Bag of Words Vector

A Bag of Words is a representation of text consisting of unique word list represented by features. The occurrence of words in each sentence is tracked but word order is not presevered. These texts are not understandable by the classifier itself; therefore, it is necessary to convert these texts into specific patterns of numeric values (i.e. numeric vectors). This representation is called the *Vector Space Model* (VSM). Each vector is represented by a set of numeric values to be considered as weights (importance).

Allahyari *et al.* [8] assume a text documents $X = x_1, x_2, x_3...x_X$ and consider a bag of words $S = s_1, s_2, s_3...s_X$ to be the set of unique (distinct) terms in the collection. The frequency of the words $s \in S$ in text document $x \in X$ is shown by $g_x(S)$ and the number of documents having the word s is represented by $g_D(s)$. The term vector for document $\vec{V_d} = (g_d(s_1), g_d(s_2), g_d(s_3), g_d(s_u))$

The weight terms use a Boolean Model and TFIDF.

Boolean Model: In this model a weight $w_{ij} = 1$ is assigned to each term $w_i \in d_j$ and a term that does not appear in d_j, $w_{ij} = 0$ [8].

Term Frequency-Inverse Document Frequency (TF-IDF): Term Frequency-Inverse Document Frequency (TF-IDF) is the most common method of measuring the weight of the term in vector space model. The function measures the weight of word from the vocabulary (each vector component) on each document [15]. This measure weights word w_i in d as

$$tf - idf(w_i, d) = tf(w_i, d) ln \frac{N}{df(w_i)} \tag{2.1}$$

where $tf(w_i d)$ is the w_i word frequency in document d, $df(w_i)$ is the number of documents where w_i appears and N is the number of documents in the collection.

Consider two sentences that consist of 10 words (5 words each). Converting the text into the 'bag of words' form produces eight words (because two words are repeated).

(1) 'Mount Everest lies in Nepal'

(2) 'Buddha was born in Nepal'

These eight (8) words are

Numeric	Word
0	Buddha
1	Everest
2	Mount
3	Nepal
4	born
5	in
6	lies
7	was

Table 2.2: BOW Vector

These unique words will get a TF-IDF (numeric) value on the basis of the weight of features. In the example, the numeric value of each word means that the value is present in that line (sentence). This technology is being used in Weka to represent sentences by vectors [81]. This transformation leads to the following representation:

Sentence 1: (1 0.480453),(2 0.480453),(6 0.480453)
Sentence 2: (0 0.480453),(4 0.480453),(7 0.480453)

In the example, a 'StringToWordVector' function is used to convert String features into a set of features representing word occurrence. Moreover, the technique refines and optimizes the input by not mentioning the words which appear in all sentences. As it can be observed in the above example,

the two words 'in' and 'Nepal' are in both sentences which has no information associated.

N-gram Representation

The n-gram [43] is an extension of the 'Bag of Words'. It is a contiguous sequence of n items from a sentence. The specific size of the window defines the sequence of the words. For example, for the sentence 'Pedro lives in Setubal'. When $n = 1$ (known as unigram), and will have individual words in a sentence.

Pedro

lives

in

Setubal

If $n = 2$ (known as bigram), then the n-gram would be:

Pedro lives

lives in

in Setubal

If $n = 3$ (known as trigram), then the n-gram would be:

Pedro lives in

lives in Setubal.

N-grams are used mostly in the computational linguistics to generate a language model for machine learning algorithms to use.

Part of Speech tags

Part of Speech tagging is the process of marking words with the part of speech (grammatical notation) based on its definition and context. POS tagging is a prevalent and widespread methodology to generate the syntactical information of each word. Part of Speech tagging is a complex task as words may have more than one POS tag depending upon the context of the sentence. Therefore, the system needs to tag a word according to the meaning, structure, and context of the sentence. Let's take an example:

Read the sentence given **above**.

Read_VB the_DT sentence_NN given_VBN above_IN.

In this example, the word 'above' is a preposition or subordinating conjunction.

Our blessings come from **above**.

Our_PRP$ blessings_NNS come_VBP from_IN above_RB.

In this example, the word 'above' is an adverb.

Stanford NLP used Penn Treebank Project [114] to annotate POS tags. POS tags help to know the type of words present in a sentence. We can differentiate the presence of the grammatical structure according to the POS label feature and also can count the number of nouns, verbs, articles, etc. Thus, Part of Speech tags are essential in the computational linguistics field.

Syntactic Parse Tree

A parse tree is an ordered tree [91] and each document that is a sequence of sentences is represented as an ordered list of ordered trees. In this way, a document can be represented in a tree structure where each root's child is the parse tree of a sentence, and the leaves are its word's lemma. This representation is named 'syntactic tree representation'. To create a syntactic tree, a parser is run (representation is given below). Figure 2.6 (a) presents the tree structure and (b) is the horizontal flat parser tree represented using brackets.

The major difference with the description of the SVM-light-tk-1.2 [89] is that a tree can be specified with the following syntax:

(A (a)(b)(c))

As an example:

```
They were further charged with 'disrupting public peace' (diataraxi
koinis eirinis), an offence under Article 192 of the Penal Code, by
openly and indirectly inciting citizens to violence or by creating rifts
among the population by the use of the words 'Turk(s)' or 'Turkish' to
identify the Moslems of Western Thrace.
```

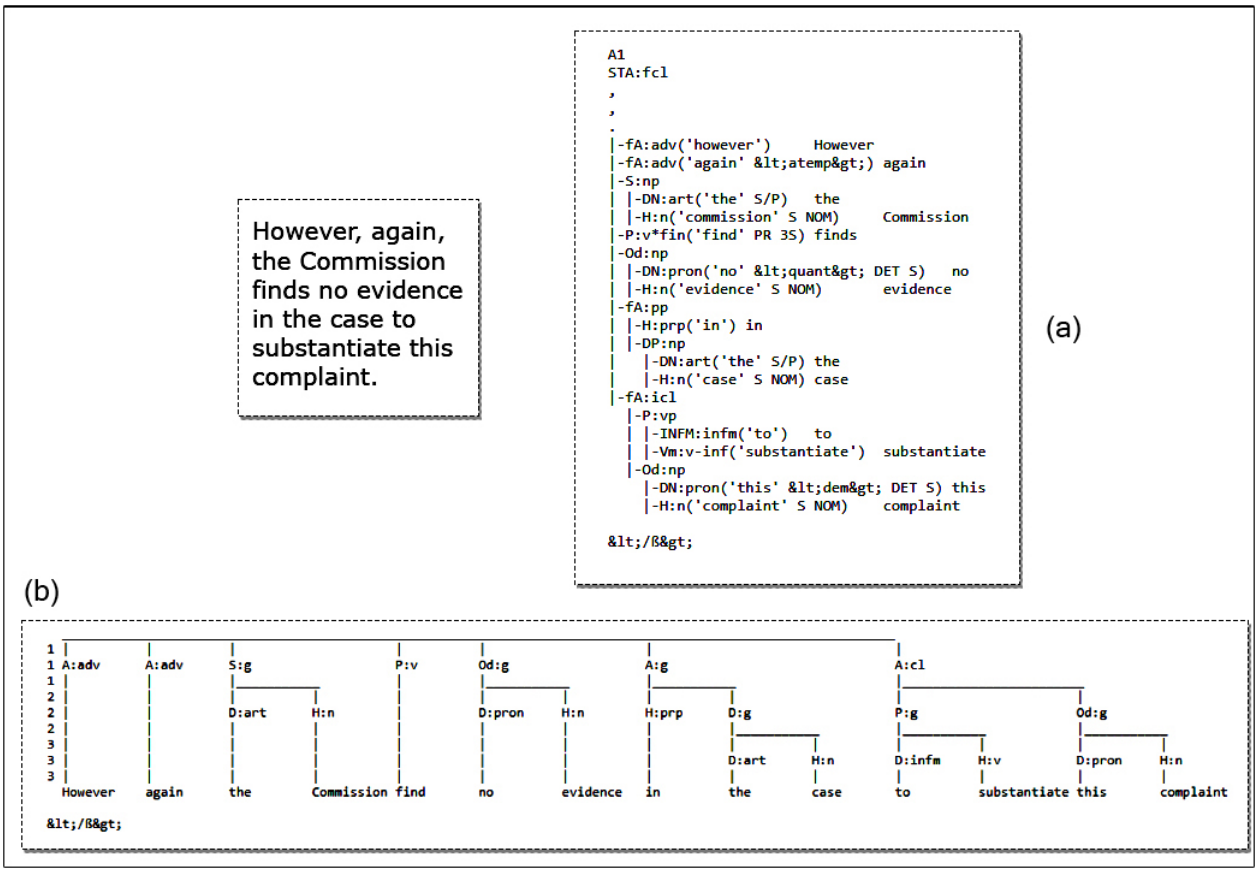

Figure 2.6: Syntactic Tree Representation

|BT| (ROOT (S (NP (PRP They)) (VP (VBD were) (RBR further) (VBN charged) (PP (IN with) (S (VP (VBG ‘disrupting) (NP (JJ public) (JJ peace’) (NN (diataraxi) (NNS koinis)) (S (VP (VBG eirinis),) (NP (DT an) (NN offence)) (PP (IN under) (NP (NNP Article) (NNP 192) (PP (IN of) (NP (DT the) (NNP Penal) (NNP Code,))))) (PP (IN by) (ADVP (RB openly))))) (CC and) (ADVP (RB indirectly)) (VBG inciting) (NP (NNS citizens)) (PP (TO to) (NP (NN violence)) (CC or) (IN by) (S (VP (VBG creating) (NP (NNS rifts) (PP (IN among) (NP (DT the) (NN population)))) (PP (IN by) (NP (DT the) (NN use) (PP (IN of) (NP (DT the) (NNS words) (NN ‘Turk(s)’)))))) (CC or) (VBG ‘Turkish’) (S (VP (TO to) (VB identify) (NP (DT the) (NNPS Moslems) (PP (IN of) (NP (NNP Western) (NNP Thrace.)))))))))))))) |ET|

‘|BT|’ represents the beginning of tree and ‘|ET|’ represents the ending of tree.

2.2.3 Feature Extraction:

Features are the symbols or numeric values representing the data that are understood by the classifier. By means of various techniques, these features are extracted from the text. For example, sentence statistics could be one kind of feature which includes counting words in a sentence, average words in a sentence, last words of a sentence, average word length of sentence, etc. In argumentation, features can be broadly classified into several classes: structural, contextual, syntactic and lexical.

Structural Features

Since argumentative sentences are found only in certain zones of the document, structural features are important for finding arguments on the basis of location. To illustrate, examples of case-law documents are considered. The arguments are rarely found in the beginning or even the middle part of the case-law document. The main reason is that these documents are written with certain rules and regulation which follows a certain format. At the beginning of the document, information about stakeholders of the court is provided, followed by information related to plaintiff and defendant. After this, most of the arguments are made in the 'The Law' section of the case-law document. Furthermore, most of the argumentative sentences are in sequential order, so 'sentence Closeness' which is a structural feature, would be effective for identifying the components of the arguments.

Contextual Features

The context of a text is important for understanding the argument presented by one party to another. To understand this, it is necessary to understand the notion of the 'context' of a text, which is especially important in the legal domain, when determining the premises and conclusion in an argument. Mochales-Palau and Moens [122] illustrate their example of an argument. If a sentence consists of 'this is because ....' this indicates that any predecessor of this clause must be a conclusion, and any successor is a premise.

Context features may be illustrated in greater depth by an example. In Table 2.3, 4 sentences are presented, followed by their argument notation in parentheses. The first sentence is the premise of argument number 21 and its conclusion is the second sentence; but at the same time, the second sentence is also the premise for argument number 22. Similarly, the third sentence is a conclusion for argument number 22 and simultaneously a premise for argument number 23. From this example, the assumption is that there is a certain relation between the sentences for which a sentence is a conclusion of one argument in one context and the premise of another argument in another context.

S. No	Sentence	AS ID
1.	The Commission notes that the proceedings instituted against the applicant have not yet been terminated.	$(21P)$
2.	He has not, therefore, been "held guilty of any criminal offence" as set out in Article 7 para. 1 (Art. 7-1) of the Convention.	$(21c, 22p)$
3.	The applicant can't, therefore, be regarded as a victim of a violation of Article 7 (Art.7) of the Convention.	$(22c, 23p)$
4.	This part of the application is therefore manifestly ill-founded within the meaning of Article 27 para.2 (Art.27-2) of the convention.	$(2c)$

Table 2.3: Examples of Context feature in the argumentative sentence

Therefore, the meaning of the words alone is not enough to classify the sentence as either premise or conclusion. So, Context features are quite important for identifying the particular component of an argument. It should be noted that the notion of context, while critical, is currently computationally troublesome.

Syntactic Features

Syntactic Features are concerned with parsing/grammatical behavior in a sentence. Two of the most popular syntactic features are POS tags and parse trees. POS tags are morpho-syntactic features which are important and effective. For example if a sentence has a modifier (e.g. an adverb or emphasis word), then it is highly probable that such a sentence is argumentative. Similarly, parsing plays an important role in dealing with a sentence's component parts by determining syntactic roles. Basically, there are two categories of parsing approaches: Constituency parsing and Dependency parsing. Constituency parsing deals with the phrasal structure of sentences by breaking a text into sub-phrases, wheres dependency parsing focuses on relations between words in sentences.

Lexical Features

Lexical features are unigram. bigram, verbs and adverbs, and word pairs. These features are the most commonly used onces for natural language processing. In argument mining, lexical features are discourse markers that have the property of detecting argumentative sentences: either premise or conclusion. In [72] 13 markers are said to be premise indicators; they are 'since', 'because', 'for', 'as

indicated by', 'follows from', 'may be inferred from', 'on the grounds that', 'for the reason that', 'as shown by', 'given by', 'may be deduced from' etc. Similarly, the following markers indicate a conclusion: 'therefore', 'thus', 'so', 'consequently', 'hence', 'then', 'it follows that', 'it can be inferred that', 'in conclusion', 'accordingly', 'for this reason (or for all these reasons) we can see that', 'on these grounds it is clear that', 'proves that', 'shows that', 'indicates that', 'we can conclude that', 'we can infer that', 'demonstrates that'. Let's take an example:

> Fear can cause accidents among older people. *Therefore*, doctors should used discretion when counseling older people about the risks of falling.

In this example, the word 'Therefore' is used indicating the sentence is a conclusion, or at least, a concluding sentence. These discourse markers are used to differentiate arguments from non-arguments and also determine the component arguments.

2.3 Machine Learning Algorithms

Machine learning algorithms can be divided into two types: supervised and unsupervised. Supervised algorithms need training data and learning is achieved by generalizing from it. The model obtained can then be applied to other unseen instances. The accuracy of the algorithm depends upon the quantity and quality training data. Similarly, in unsupervised approaches, patterns are learnt from non-annotated examples. Let's discuss some of the machine learning algorithms that are used in this work.

2.3.1 Support Vector Machine

Support Vector Machines [190] are linear classifiers that learn from training data and create a function to make predictions about novel data. Taking a set m of input vectors x_i $(i = 1, ..., m)$ for the training, where each of these input vectors has several component features. These input vectors are paired with the corresponding label y_i [37, 124].

The training data can be viewed as labeled data points in input space. In Figure 2.7 there are two classes of well-separated data; the learning task aims to find a directed hyperplane, that is, an oriented hyperplane such that the examples labeled y_i=+1 is separated from those labeled as y_i=-1.

Support Vector Machines [124, 190] are linear classifiers that construct a hyperplane with the

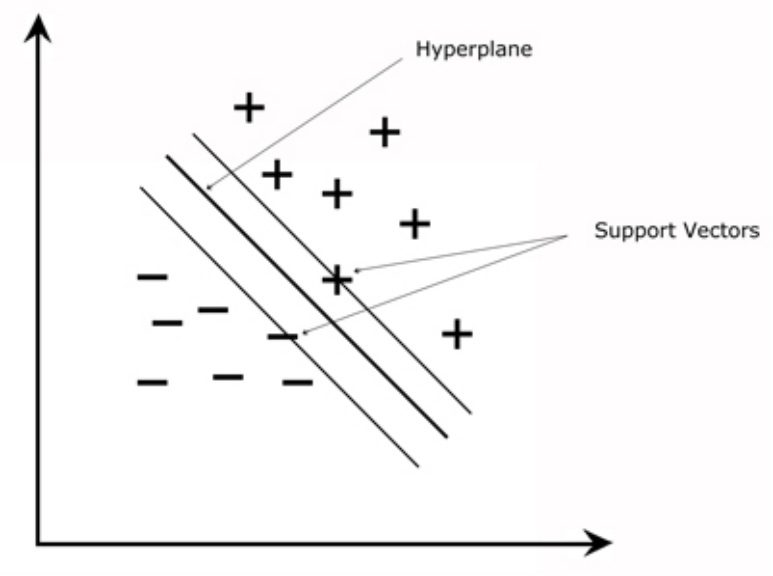

Figure 2.7: Support Vector Machine

largest margin between the positive and negative examples to reduce the error of the classifier. Let us suppose that the set S has n training examples:

$$S = (x_1, y_1), ...(x_n, y_n)$$

where $x_i \in \mathbb{R}^p$ ($p-$dimensional space) and $y_i \in \{+1, -1\}$, indicating that x_i is a positive or a negative example respectively. Then, the equation of the separating hyperplane can be represented by

$$< w.x_i > +b = 0$$

where w defines the direction perpendicular to the hyperplane. The value b helps to move the hyperplane itself. The term w and b are referred to *weight vector* and *bias*. SVM is based on ideas of VC (Vapnick-Chervonenkis) dimension and the Structural Risk Minimization (SRM) principle [191].

The **Complexity parameter**, C, is the coordinate position of a hyperplane in the vector space. Selecting an appropriate complexity parameter is done by placing a hyperplane in the appropriate position so that misclassified points are within the classification zone. If the value of C is large, then there will be a smaller margin between the support vectors. In such cases, if the classifier is trained with such training points, then the generated model can classify the testing data set more accurately. Inversely, for a very small value of C, the optimizer needs to look at a larger margin-separating hyperplane (even if the hyperplane misclassified the points) [171].

2.3.2 Random Forest

Tin Kan Ho is the first developer of the Random decision forest using the random subspace method [83]. Leo Brieman extended the algorithm and gave it the name Random Forest to a collection of decision trees [32]. The extension includes the 'bagging' idea of Breiman and also the random selection of features. The algorithm of Random Forest [132] is presented in algorithm 1.

Algorithm 1: Algorithm Random Forest for Classification

1. **for** *b=1 to B:* **do**
   (a) Draw a bootstrap sample Z* of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.

end

2. Output the ensemble of tree $\{T_b\}_1^B$

To make a prediction at a new point x:

Classification: Let $C_b(x)$ be the class prediction of the b^{th} random-forest tree. Then $C_{rf}^B(x) =$ majority vote $\{C_b(x)\}_1^B$

The Random Forest algorithm [132] has relatively high accuracy among algorithms performing classification. It can handle large data sets, and it has features to balance and unbalanced the data.

2.3.3 Fuzzy Clustering

A clustering technique is used to derive natural data groups from a suitably sized data set. It helps to get a brief picture of system behavior obtained from the data groups. There are various techniques to do the clustering; among them *Fuzzy c-means* (FCM) is a soft clustering technique that handles data points that can belong to more than one cluster [202]. This technique was developed by J.C. Dunn in 1973 and later on improved by J.C. Bezdek in 1981 [26]. FCM generates membership values ranging from 0 to 1 and clusters a dataset into n clusters where every data-point in the dataset belongs to every cluster with a certain probability/degree. This implies that a data-point which lies close to the center of a cluster has a higher probability of membership and that a data-point which lies towards the center of another cluster that is far from the data-point has lower certainty. FCM initially starts from random guesses for the cluster center, which is marked as the mean location for that cluster. Through an iterative process, the membership value for each data-point is updated.

This way, the cluster center are correctly located. The iteration is performed based on minimizing the objective function that estimates/reduces the distance between the cluster center and a data-point [116]. When clustering, every algorithm follows different rules for partitioning. FCM uses a value called the 'partition matrix component', m, that controls the degree of fuzzy overlap, and the value m should be greater than 1. The overlap indicates the boundaries between clusters. It also describes data-points that may have a significant membership in more than one cluster [116].

The equation for the FCM is described below. The following notations are used. Let

1. N be the number of data points with m-dimension
2. $x_i = (x_{i1}, \ldots, x_{im})$ be a m-dimensional data point, $\forall\, i \in \{1, \ldots, N\}$
3. k be the number of clusters, where $k \in \{2, ..., N\}$
4. l be the fixed level of cluster fuzziness with $l > 1$
5. $c_j = (c_{j1}, \ldots, c_{jm})$ be the m-dimension center of the cluster, $\forall\, j \in \{1, \ldots, k\}$
6. $\epsilon \in [0, 1]$ be a termination criterion.
7. $U = [u_{ij}]_{N \times k}$ be a matrix, where $u_{ij} \in [0, 1]$ is the degree of membership of x_i in the cluster j, $\forall\, i \in \{1, \ldots, N\}\, and\, j \in \{1, \ldots, k\}$.

The main idea of the algorithm is to minimize the objective function

$$J_l = \sum_{i=1}^{N} \sum_{j=1}^{k} u_{ij}^{l} \left|\left|x_i - c_j\right|\right|^2,$$

The algorithm can be represented by the following steps.
Step 1. Initialize matrix $U^{(0)} = U$
Step 2. Assume that $U^{(k)}$ is known. The center's vectors $C^{(k)} = [c_j^{(k)}]$ are determined by the following formula

$$c_j^{(k)} = \frac{\sum_{i=1}^{N} u_{ij}^{l} x_i}{\sum_{i=1}^{N} u_{ij}^{l}} \qquad \forall\, j \in \{1, \ldots, k\}$$

Step 3 : Calculate $U^{(k+1)} = [u_{ij}^{(k+1)}]$ by the formula

$$u_{ij}^{(k+1)} = \frac{1}{\sum_{r=1}^{k} \left(\frac{\|x_i - c_j\|}{\|x_i - c_r\|} \right)^{\frac{2}{l-1}}} \quad \forall\, i \in \{1, \ldots, N\}\, and\, j \in \{1, \ldots, k\}$$

Step 4 : Verify the stop condition if $\left\| U^{(k+1)} - U^{(k)} \right\| < \epsilon$ then stop. Otherwise, repeat step 2.

2.4 Computational Tools

There is a handful of tools we use to accelerate the speed of research and to perform the core lingusitic analytic tasks. These tools are used for the preprocessing, classification, clustering and development of features, and also for measuring the performance of the proposed features by examining how well they achieve classification. There are several tools available for computational activities. Some of the tools that were used in our experiments are described below.

2.4.1 Weka

Weka [81] (Waikato Environment for Knowledge Analysis) is an open source data mining software developed in the Java programming language under the General Public License. Initially, the tool was designed in TCL/TK, c, and makefile in 1993. Later in 1997, the software was rewritten from scratch but in Java, to create platform independent and user-friendly application. It consists of 49 data preprocessing tools, 76 classification/regression algorithms, eight clustering algorithms, 15 subset evaluators, ten search algorithms for feature selection, and three algorithms for finding association rules. The software provides a graphical user interface as shown in Figure 2.8 and a command set for access to the functions. The software is developed and maintained at the University of Waikato in New Zealand.

2.4.2 Tree Kernels

Structural ambiguity is a characteristic of natural language sentences. To mitigate such ambiguity, it is necessary to work with syntactic features, using a Tree Kernel. The sentences Parse Trees were generated by SVM-LIGHT-TK 1.5 [128] and presented to SVM-Light [89]. The advantage of a tree kernel is the ability to generate a large number of number syntactic parser trees and let the classifier select the most suitable/relevant onces according to its specific application. Further, tree

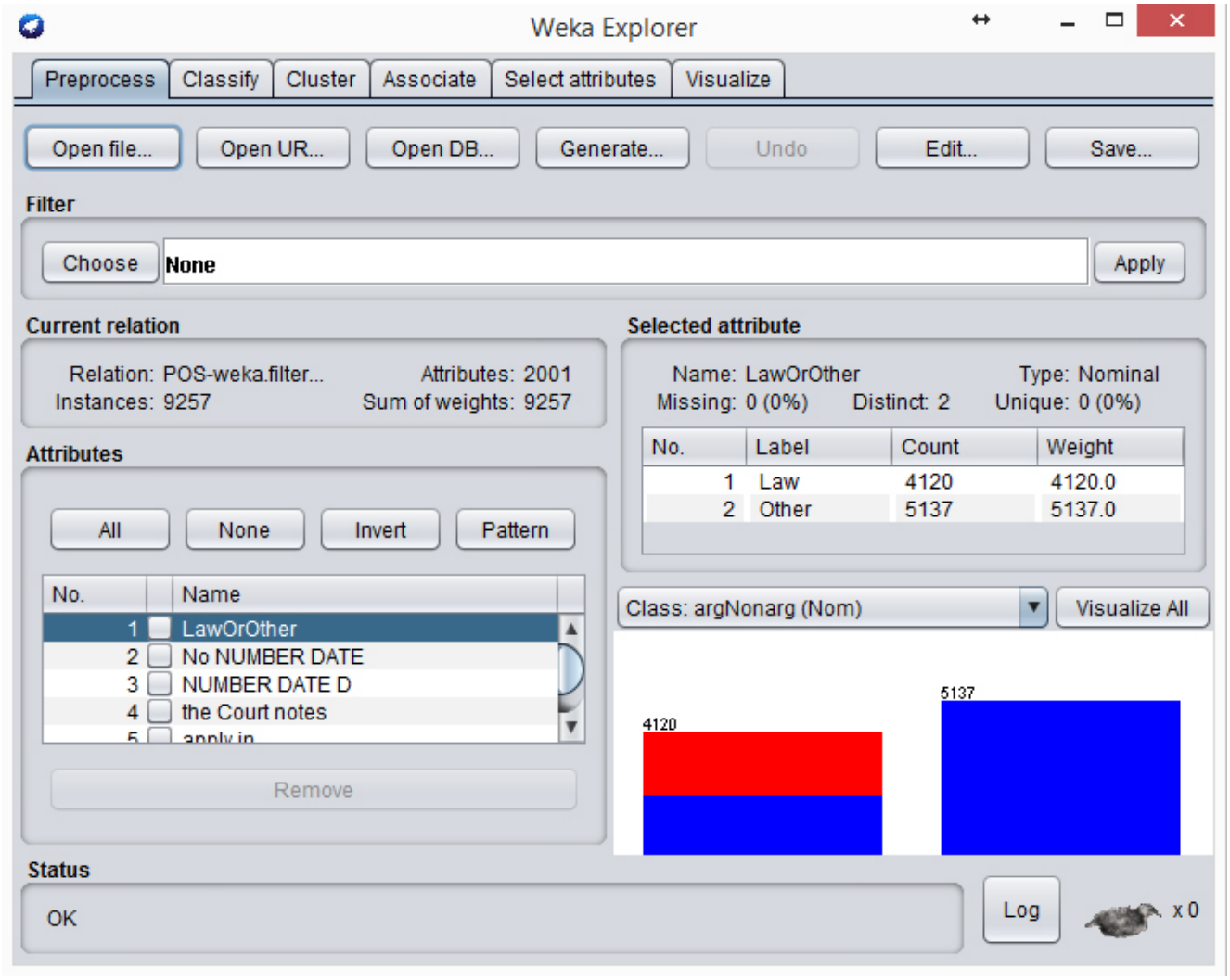

Figure 2.8: Interface of the Weka [81]

kernels are used to measure the similarity between parse trees. This concept of Tree Kernel is used in the Argument Element Identifier module described in section 5.1 and section 6.1.3. Three kinds of features are developed : Syntactic Parse features in the form of Bracket notation tree, TF-IDF features, and a combination of both.

2.4.3 Word2vec

Word2vec is one of the prominent techniques to represent words as vectors. This technique addresses the semantic and contextual information present in the text. Take the following sentences: *Kids like ice cream* and *Children like ice cream*. In this example, the word *Kids* and *Children* refers to the same meaning or concept, so although these two words are different, they will have similar word vectors due to the similarity of their semantics.

This notion was proposed by by Mikolov *et. al* [118] and can be implemented in two different ways. As a 'Continuous Bag of Words' (CBOW) or as a 'Skip gram'. With CBOW, word vectors are predicted from the context of adjacent words. CBOW is suitable for use if the corpus is relatively small in scale. Its computing capabilities are faster than with Skip-grams. In Skip-grams, the vectors of context words are predicted from a vector of given words. Skip-gram is suitable for use on a dataset that consists of a large corpus of high dimensionality, but its computational capacity is slow in comparison to CBOW.

We applied this word2vec concept in the argument builder module to cluster argumentative sentences into an argument. The details of this experiment are described in Sections 5.2 and 6.2.

2.5 Performance Measuring parameters

Precision and recall [15, 157] are measures used for evaluating the performance of information retrieval systems. Precision [163] is defined as the number of relevant documents retrieved divided by the total number of documents retrieved. The recall is defined as the number of relevant documents retrieved, divided by the total number of elements that belong to the positive class. F-measure is the harmonic mean of precision and recall, and belongs to a class of functions used in information retrieval. F_β can be written as

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.2}$$

when $\beta = 1$, precision and recall have the same weight and it's referred to as F_1

These evaluation methods can be applied to the analysis of clustering techniques as well. Suppose cluster α of gold-standard data sets consists of N_α number of the sentences. Similarly, cluster β of the predicted system consists of N_β number of sentences. Precision ($P_{\alpha\beta}$), recall ($P_{\alpha\beta}$). The equations are :

$$P_{\alpha\beta} = \frac{n_{\alpha\beta}}{N_\beta} \tag{2.3}$$

$$R_{\alpha\beta} = \frac{n_{\alpha\beta}}{N_\alpha} \tag{2.4}$$

$$P = \frac{\sum_{\beta=1}^{c} N_\beta P_{\alpha\beta}}{\sum_{\beta=1}^{c} N_\beta} \tag{2.5}$$

$$R = \frac{\sum_{\beta=1}^{c} N_\beta R_{\alpha\beta}}{\sum_{\beta=1}^{c} N_\beta} \tag{2.6}$$

Here, $n_{\alpha\beta}$ is the total number of sentences matched in between cluster α and cluster β. Overall performance is measured by calculating the weighted average of the individual precision (P) and recall (R) which are expressed in formula 2.5 and 2.6. F-measure (f_1) is shown in 2.2.

These evaluation measure are used to measure the performance of proposed features for use by the Argument Builder module in Section 6.2 which clusters argumentative sentences to form an argument. The cluster α is an argument that is present in the ECHR case-law and cluster β is the argument predicted by the proposed system from it. The equations 2.2, 2.5 and 2.6 are used to calculate the precision, recall and f-measure to determine the performance of each classification feature described in Section 5.2.

Cluster purity is used to evaluate the cluster accuracy. It can be computed by counting the number of correctly assigned entities and dividing the total number of N [167]. Formally

$$ClusterPurity\,(\varphi, C) = \frac{1}{N} \sum\nolimits_{d=1..k} max_{e=1..k} |w_d \cap c_e| \tag{2.7}$$

where N is the summation of the total number of elements in all clusters, $\varphi = \{w_1, w_1, \cdots w_k\}$ is the set of clusters and $c = \{c_1, c_1, \cdots c_k\}$ is the set of classes. We interpret w_d as the set of sentences in w_d and c_e as the set of sentences in c_e in Equation 2.7.

2.5.1 Stratified Cross-Validation

The Cross-Validation (CV) called rotation estimation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is a model evaluation method where the original dataset is divided randomly into k subsets (in our experiments, k=10). Then, one of the k subsets is used as the test set, and the other k-1 subsets are put together to form as a training set; a model is built from the training set and then applied to the test sets. This procedure is repeated k times (one for each subset). All the data appear in a test set exactly once only and appear in a training set k-1 times.

2.6 Statistical Tests

2.6.1 Paired T-Test

The paired t-test is a statistical hypothesis test between two means of random samples from a population. These means are normally distributed. The samples are collected under the same type of conditions, or from the same population. This test is mainly to see if the difference between two observations is zero. If two paired sets are X_i and Y_i where $i = 1, 2, 3.....n$. The paired differences are normally distributed, identical and independent. The test determines if any difference achieves significance [168].

The paired t-test is given by

$$t = \frac{(\sum D)/N}{\sqrt{\frac{\sum D^2 - \left(\frac{\left(\sum D\right)^2}{N}\right)}{(N-1)(N)}}} \tag{2.8}$$

where $\sum D =$ sum of differences and $N = 1, 2, 3, ..., n$ (n is any positive integer number)

As an example consider the following: A group of N students is given a test before they take a course, and their scores form the set X. After students finish the course, they are given another test to evaluate what they have learned from the course. These new scores form the set Y [168, 179]. The 'paired t-test' is used to test the validity of the results. The critical assumptions of a paired t-test are as follows:

- Both group's standard deviations should be approximately equal.
- The population that the data comes from should be normally distributed.
- The comparison data pair have to be identical.
- The pair should not be dependent on each other.

2.7 Summary

This chapter has presented an overview of the theoretical concepts of argumentation models, natural language processing, and machine learning. Since the main background of the work is 'argument

mining in the legal domain', it's important to show the relevance of the natural language processing, machine learning and argumentation models for this domain. Principally, five topics have been discussed, starting with a systematic review of the theoretical concepts of argumentation models. Then, natural language processing tools are described. Third, Machine Learning Algorithms that are used in the thesis are shown. Fourth, Performance measure parameters are explained. Finally, statistical tests are briefly described.

3

State of the Art

This chapter presents an overview of the state of the art of research in the domain of this thesis. Starting with a historical perspective of argument since the time of Aristotle in Section 3.1 and moving on to explore the connection between artificial intelligence, argument, and the law in Section 3.2.; In Section 3.3, argument mining related proposals are introduced. This section is divided into three subsections: Argumentative Sentence Detection, Argument Boundaries Detection, and Argument Structure. The approaches, features and algorithms along with the results obtained are also described in detail. Next, the corpora used by the several researchers are analyzed in section 3.4. Finally, a summary section with conclusions is presented.

3.1 Argument - a Brief History

The 'history of the argument theory' begins with ancient human civilization. The first notable action starts in Ancient Greece [125] where a central part of Western education aimed at training public speakers and writers moved audiences to action. The context of arguments begins with the predecessors of the Greek philosopher Aristotle (384 – 322 BC) who were interested in constructing persuasive arguments [86]. Conversely, Aristotle favored a systematic critical observation approach for analyzing and evaluating arguments. He proposed a logic called *Syllogistic* which was used to combine statements to deduce a conclusion. Furthermore, he introduced *modal logic* that was associated with the concept of possibility, necessity, belief, and doubt. The contribution of Aristotle is considered as the breakthrough for the philosophical analysis in the Philosophical world. After the death of Aristotle, the Greek Philosopher Chrysippus (280-206 B.C.) considered propositions either to be true or false [86]. From this theory he developed the rules for identifying truth or falsity from compound propositions. These two Greek philosophers were pioneers in the contribution of logic and philosophy.

Thirteen hundred years after these two philosophers' death, investigation in philosophy had a re-awakening. The physician Galan, who lived from 129 AD to 199 AD developed the theory of compound categorical Syllogism [86]. After that, Abelard (1079 - 1142), who was the first major logician, reconstructed and refined the logic instigated by Aristotle and Chrysippus. The book *Summulae Logicales* written by Peter of Spain (ca. 1205, 1272) became the standard logic textbook for three hundred years. At the end of the 11^{th} century William of Ockham (ca. 1285 -1347) originally contributed to philosophy and logic with the extension of the theory of modal logic which studied the forms of valid and invalid syllogisms [86].

In the middle the 19^{th} century logic was in the limelight due to its extremely rapid development. Many philosophers and mathematicians such as Augustus De Morgan (1806-1871), George Boole (1815-1864), William Stanley Jevons (1835-1882), and John Venn (1834-1923) work on Symbolic logic [86].

During the twentieth century, much of the work in logic was focused on the formalization of logical systems. Until the middle of the 20^{th} century, the approach to argumentation was based on logic, rhetoric [125]. The recognition of the importance of argument began after Toulmin's model [184]. In 1980s, Birnbaum *et al.* [29], presented an AI model of argument. The authors used a human interaction (Arabic and Israeli conversion) corpus to represent argument in the computer program by dealing with the rules and structures in the corpus. In the 21^{st} century, the Logic, Philosophy, and technology converge to create a coact relationship. With the passage of time [86] and

the rapid development of technology, its importance has grown rather exponentially in fields such as Literature, Mass-media, Communication, Logic, Law and Artificial Intelligence. Lippi and Torroni [106] mentioned that after the appearance of Pollock [144], Simari and Loui [169] and Dung's models [54], argument models began to appear in the area of Intellectual technology zone which creates connectivity between Philosophy and Artificial Intelligence, giving rise to a new field named as Computational Argumentation. The last two decades have witnessed, in several fields, the rapid development of argument mining. Now, argumentation is found in diverse areas of knowledge such as Linguistics, Logic, Social Science, Political Science and Artificial Intelligence.

3.2 Artificial Intelligence and Law (Legal Argumentation)

The law is a discipline supposed to be followed by every law-abiding citizen of the nation. It is a set of rules with exceptions, reasons, guidelines, and qualifications. Commonly, legal bodies are used to regulate the behavior of the citizen and society. The Law is not exact; unlike a mathematical formula, it cannot be applied mechanically, depending only on the crime committed. Each country has its constitutional law which is either in written or unwritten format and existing as constitutional conventions. Besides this, there are some courts with jurisdictions external to the nation, such as the European Union Court of Justice, which is considered to be supranational. These courts produce an extensive number of case-law documents which are difficult to analyze, and a burden to those trying to research precedents. Therefore there is a need for building an automated system which can analyze such documents and provide any necessary information. Analysis of precedent cases is an important factor when trying to predict the outcome of current cases, and several attempts to use AI for the analysis of case-law documents have been made since the 1980s, resulting in the creation of the 'International Conference on AI and Law' (ICAIL) in 1987. This conference is dedicated to the publication of research on AI and Law and holds a meeting every two years. The conference is organized under the foundation of the 'International Association for Artificial Intelligence and Law' (IAAIL). Bench-Capon *et al.* [17] mentions that the existence of the ICAIL is considered to be the birth for the AI and Law Community. Legal reasoning is seen as a path towards a bilateral relationship between Artificial Intelligence and Law, with one of its principal goals being to provide support for the analysis of precedents and existing justifications. Prediction of outcomes before a trial is also a matter of great interest. A year after the establishment of ICIAL, the JURIX conference, now conducted annually, was begun by the JURIX Foundation for Legal Knowledge Based Systems. Since 2007, Japan has run 'Juris-informatics' (JURISIN) workshops under the Japanese Society of Artificial Intelligence.

The TAXMAN system is a legal analysis system developed in 1977, whose objective is to identify majority and minority opinions computationally from landmark Supreme Court Cases trailed in the United States concerning tax law [117]. TAXMAN showed the importance of being able to construct *theories* derived from a knowledge base [4]. Similarly, Gardener [193] proposed a system that displayed the questions raised in a given case by providing the case-law document number, enabling the user to quickly make an estimate of the difficulty of an upcoming case. In 1990, the HYPO project [155] was a starting point for dialogical analysis of legal reasoning; HYPO is a case-based reasoning system that compares and contrasts legal problems by using a Dimension (generalization scheme). Ashley and Vincent Aleven developed another tool called CATO [4, 7] which is both a simplified version of the HYPO but also an extension of it. The purpose of CATO is not to predict or recommend for decisions. It helps to form better case-based arguments and also helps to improve the ability to distinguish cases. Carneades [70] is an argumentation system tool that is designed for constructing arguments via rule-based argumentation schemes. It supports several argumentation tasks that include reconstruction, evaluation, and visualization, and also enables expansive primitive sets that allow users to construct arguments in different domains [70, 165]. The first version, which was released in 2011, developed by the 'European ESTRELLA project' (IST-2004 - 027655) from 2006 to 2008. The latest version, *Carneades-4*, was released in July 2017. The software is open source and is freely available.

3.3 Argument Mining

Professor Marie-Francine Moens[1] defined argumentation mining as follows:

"Argumentation mining can be defined as the detection of the argumentative discourse structure in text or speech and the recognition or functional classification of the components of the argumentation" [125].

Similarly, Lippi and Torroni [106] said that main goal of argumentation (or argument) mining is to extract the arguments from a corpus to provide structural arguments data to the computational models. They mention that most researchers use three subtasks in their argument mining systems. These are Argumentative Sentence Detection, Argument Component Boundary Detection, and Argument Structure Prediction.

[1] Director of the Language Intelligence and Information Retrieval (LIIR), KU Leuven

3.3.1 Argumentative Sentence Detection

The first step in processing is to detect a sentence that contains an argument or part of it. The task can be seen as selecting an appropriate classifier and features in order to distinguish argumentative sentences from non-argumentative. Some classifiers used to classify argumentative sentence are the Support Vector Machine [58, 71, 105, 121, 159, 164, 176], Naive Bayes [29, 58, 121, 176], Maximum Entropy classifiers [121], Decision Trees [58, 172] and Random Forest [58, 176].

From the literature, it seems that SVM is the most favored machine learning algorithm for sentence detection. Regarding the selection of features, n-gram, POS, bag of words, textual features, semantic features, syntactic features, and lexical features are used. In the following section, techniques, features and classifiers that are being used to identify argumentative sentences are discussed in depth.

Moens *et al.* [126] used n-gram, verb nodes, word couples and punctuation features to identify argumentative sentences and obtained an average of 74% accuracy in corpora of various types, but this dropped slightly to 68% when applied to a legal corpus. The authors extended this work [122] by adding more features: modal auxiliary, keywords, negative/positive words, text statistics, punctuation keywords, same word in previous, current and next sentences and 'first and last words in next sentences'. The results are reported to be better than the previous experiment, with an accuracy of 90%. Similarly, using a context-free grammar [139] the authors obtained around 60% accuracy in detecting argumentation structures and around 70% f_1 when identifying components of arguments. Likewise, the authors [120] studied ten legal documents (from the ECHR) and generalized the structure of the arguments present in these judicial documents using top-down grammar (LL) and bottom-up grammar (LR) schemes; they used the LR version of the argumentative grammar for the analysis and obtained a precision of 59% with recall at 59% for the premises. However, Lawrence *et al.* [99] proposed a different approach using two Naive Bayes algorithms to identify the proposition a text word. The first Naive Bayes is used to find the starting word of the propagation and the second Naive Bayes is used to find the end of the proposition. Nonetheless, instead of identifying argumentative text and non-argumentative text like Moen, the author identified the segments connected to the proposition (identified in the first phase), considering a 'connected' phrase to be argumentative and a 'non-connected' text to be non-argumentative.

3.3.2 Argument Component Boundary Detection/Clustering

The main goal of argument boundary detection is to detect the beginning and end of arguments. Moens *et al.* [121] performed experiments aiming to find the boundaries of an argument. Since components of an argument are dispersed throughout the text, the authors proposed to use 'semantic distance' to handle the problem. Context-free grammars (CFG) were used to detect the argument structure in a very limited portion of case-law documents and obtained 60% accuracy. Sardianos *et al.* [164] instead detected boundaries by classifying words (tokens) of a sentence as *boundary tokens* i.e. the ones that start or end an argumentative segment. Cabrio and Villata [36] used a combination of textual entailment framework and bipolar abstract argumentation to evaluate argument texts and to find a relation between arguments. Lawrence *et. al* [99] performed a manual analysis in addition to an automated analysis to find the boundaries of an argument, and to build the train and test sets. The authors relied on help from experts to manually analyze the text. For the automatic analysis they used two Naive Bayes classifiers; one to identify the first word of the proposition and another to identify the last word. Likewise, Levy *et al.* [101] proposed a pipeline method called 'Context Dependent Claim Detection' (CDCD) with three consecutive steps for identifying the boundaries of context-dependent claims in Wikipedia Articles; the first *Sentence Component* determines whether a candidate sentence contains a 'Context-Dependent Claim' (CDC) or not; the second detects the exact CDC boundaries within a CDC sentence; the final step is to select the most relevant claim using a logistic regression classifier.

Stab and Gurevych [172] proposed a scheme that includes the annotation of claims and premises, as well as support and attack relations for capturing the structure of argumentative discourse. They used structural, lexical, syntactic and contextual features to determine argumentative discourse structures from Persuasive essays. The experiment obtained an f_1 of 0.726 when identifying argument components. Florou *et al.* [63] identified arguments (in the Greek Language) that support or oppose an opinion; they developed a Java Annotation Patterns Engine (JAPE) grammar that extracts the tense and mood of each verb chunk. The experiment was performed on 677 text segments with an average of 60 words using a J48 classifier. The results show that verb tense with discourse markers appear to be significant features which obtained an f_1 of 0.764.

Boltuzic and Snajder [31] conducted an investigation into argument-based opinion mining from online discussions. As a source of data, user comments containing arguments on the topic Under God in Pledge (UGIP) and Gay Marriage (GM) were selected from two websites: *procon.org* (containing user comments) and *Idebate.org* (containing the arguments) and assembled into the COMARG Corpus. Their investigation was a multiclass classification problem, requiring the classifier to predict the correct label from the set of five possible labels; these labels were: the capital letter 'A' for

explicitly attacks, lowercase letter 'a' *for implicitly attacks*, uppercase letter 'N' for *make no use of the argument*, lowercase letter 's' for *implicitly supporting the argument*, and capital letter 'S' *for explicitly supports the argument*. Three kinds of features were used: Textual Entailment (TE), Semantic Text Similarity (STS) and Stance Alignment (SA) features. Three other experiments using more multiple features were also done by combining them e.g. STS+SA, TE+SA and TE+STS+SA. The result shows that the STS+TE+SA model slightly outperforms the TE+SA model on the A-a-N-s-S (classification of a comment-argument into one of the five labels) problem, while on Aa-N-sS (two labels of equal 'polarity') and A-N-S (comment-argument pairs where arguments are either not used or used explicitly), the TE+SA model performs best.

Persing and Ng [143] present a model for finding the strength of an argument in student essays. The author annotated 1,000 essays from the International Corpus of Learning English (ICLE) [57]. The dataset was divided into three parts: 60% model training, 20% parameter training and feature selection, and the final 20% for testing. Since there is a standard platform for measuring strength, two baseline systems are predicted: Baseline 1 develops an argument strength score based on the argument's frequency of occurrence. Baseline 2 is a learning-based version of Ong *et al.* [137] system. Their system significantly outperformed the baseline system that relied solely on features built from heuristically applied sentence argument function labels by up to 16.1%. Furthermore, Habernal and Gurevych [79] proposed an approach to predict how convincing an argument would prove to be. They conducted two tasks: (1) identifying which pair of arguments was most convincing and (2) ranking arguments based on their convincingness. Two algorithms namely SVM and bidirectional LSTMs, were used to obtain accuracies of 0.78 and 0.74 respectively, and a Spearman's correlation coefficient of 0.35-0.40 in a cross topic scenario.

In the argument mining field, there is not much research work using clustering techniques to identify and group argumentative sentences into arguments. Clustering techniques were not considered as an appropriate technique for information retrieval in the 1980s, due largely to computational complexity and a disappointing lack of accuracy. To approach this problem, in 1988 Cutting *et al.* [51] proposed a new approach to cluster documents called Scatter/Gather. At first, from the collected documents the system distributes them in small clusters or groups with short descriptions of them for the user. Depending on the description, the user chooses one or more from the groups for advanced study. From these selected groups, a new sub-collection is formed. Then the system again applies the clustering technique to distribute this sub-collection into smaller groups and process for the users [51]. Huang [85] compared and analyzed the effectiveness of the distance function and similarity measures in partial clustering of text documents. The author evaluated five measures empirically: Euclidean distance, Cosine Similarity, Jaccard coefficient, Pearson correlation coefficient, and averaged Kullback-Leibler divergence. He used two measures, purity and entropy, to evaluate

the overall quality of clustering solutions. Li *et al.* [103] introduced a combination of semantic information and word order of sentences to compute the similarity between short texts. Li *et al.* [102] proposed two new text clustering algorithms: 'Clustering based on Frequent Word Sequence' (CFWS) and 'Clustering based on Frequent Word Meaning Sequence' (CFWMS). The key feature of these two algorithms is that both consider the text document as an ordered sequence of words rather than simply a 'bag of words'. The difference between these algorithms is that CFWS uses the frequent word sequence to reduce the high dimensionality of the documents. In CFWMS, words are converted into word meanings by using a Wordnet ontological database, and the authors found that this is more accurate than CFWS. Similarly, Contractor *et al.* [48] developed a system that summarizes scientific articles by using the Argumentative Zone (i.e. features and final sentence selection process). Their method contains of two stages: classification and sentence clustering. The classifier distinguishes the sentence that needs to be summarized. The selected sentences are clustered into groups by using argument zone labels to reduce redundancy in the summaries. The result shows that Argumentative Zone improves f_1 score by approximately 7% in full document summarization and by 54-76% in customized summarization.

Another significant area of the research being developed is to implement an evaluation procedure. There are several evaluation validation criteria for the Fuzzy c-means Clustering Algorithm such as 'Xie-Beni', 'Purity', 'Entropy', and 'Partition Entropy' [26]. Achananuparp *et al.* [2] evaluated sentence similarity measures by considering word overlap, TFIDF, and linguistic measures. Their results show that with their low-complexity data set, the linguistic measure is much better at identifying paraphrase than the word overlap and TF-IDF measures. Further, Hotho *et al.* [84] used the Wordnet [119] to improve the results of text clustering.

3.3.3 Argument Structure

The Argument Structure is concerned with the identification of the internal structure of arguments; (i.e. identification of components of arguments as either premise or conclusion). In most of the previous investigations, researchers identified the links between the arguments or argument components instead [106]. Teufel proposed an 'Argumentative Zone' to identify the components of arguments from scientific articles [182]. Similarly, Park and Blake [140] identified claims within scientific publications. Palau and Moens [139] used Maximum Entropy and Support Vector Machine classifiers and obtained f_1 scores of 0.68 and 0.74 for identifying premises and conclusions respectively. Biran and Rambow [28] proposed an approach to identify the justification for subjective claims in interactive written dialogs using a corpus from 309 blog threads at LiveJournal [107]. They used a Naïve Bayes classifier, and the results were found to be statistically significant using paired permutation

tests on key system combinations. Rosenthal and Mckeown [161] investigate claims that express an opinionated belief from 285 LiveJournal blog spots [107] and 51 Wikipedia discussion forums. The authors used lexical and social media features, committed belief (e.g. *I know...*), non-committed (*I may*) and not applicable (*I wish*). The experiment was conducted using their Logistic Regression algorithm. They show that lexical and social media features differ in cross-domain classification. The performance of POS and n-gram has a strong influence on the accuracy of the results. The POS tags were found to be the most useful features for the LiveJournal corpus, while n-gram was better for Wikipedia and according to the authors, 'committed belief' was useful in both. Roitman *et.al* [158] presented a novel approach to retrieve claim-oriented documents. They applied two steps: the first was topic-based to retrieve similar articles; the second was a claim-oriented re-ranking that ranked on the basis of potential claims proposed from features. The results improve the document and claim recall by 10.8% and 10.3% respectively.

Kwon *et. al* [96] intended to identify the precise sentence that is the focus of the main agenda (the claim). Claims were classified on the basis of 'polarity' (positive and negative) rather than classifying whole documents. Two consecutive steps were followed, first, two supervised machine learning algorithms SVM and BoosTexter, were used to identify claims by using lexical and structured features obtaining an f_1 score of 0.52 using SVM and 0.55 using BoosTexter. Since Boostexter obtained the highest f_1 score in the first step, this was used in the second step for the identification of claims (polarity classification) and obtaining f_1 score of 0.67.

Similarly, Guggilla *et. al* [77] described a supervised approach, based on a deep neural network for classifying claims in an online argument. Two claims data sets, 'Factor/Feeling Debate Forum Posts' from proposed by Walker *et. al* [195], and 'Verifiable and Unverifiable User Comments' suggested by Park and Cardie [141] were used. A binary classification was performed by Walker *et al.* on a factual/feelings dataset, and a multi-class classification on a 'verifiability' data set. The experiment was conducted using 'Convolutional Neural Networks' (CNNs) and 'Long Short-Term Memory Networks' (LSTMs) for claim classification. On the verifiability data set, they obtained a 70.47% f_1 score and a 70.34% f_1 score using the CNN and LSTM methods, respectively. On the other hand, Park and Cardie [141], Park *et al.* [142] performed claim classification on the same dataset using SVM and CRF classifiers and obtained f_1 scores of 68.99% and 63.63% respectively, significantly less than those obtained by Gugilla *et. al*. Furthermore, Gugilla *et. al* obtained f_1 scores of 79.56% and 75.10% on Factual vs. Feeling Claims Data Set by using CNN and LSTM-based methods respectively, with distributional embedding. The performance of LSTM is lower than that of CNN, but better than the SVM baseline (obtained f_1 of 70.24%) and Naive Bayes (obtained f_1 of 65%). Sardianos *et al.* [164] used the CRF algorithm to segment argument components from news and social webs texts in the Greek language. Apart from the function that is generated from

the words and POS tag, the authors emphasized cue words which signal the presence of a premise segment. Moreover, these components play an efficient role in the argument extraction process. Their evaluation was performed with various words as context (0, $\pm$2, and $\pm$5 words before and after the word under concern). The overall performance of the model was improved when two or five words were used as context. The best results were obtained with a two word context, yielding an f_1 score of 0.353.

Habernal and Gurevych [80] proposed an approach to determine the components of arguments (premise, claim, backing, rebuttals, and refutations) in user-generated web discourse. They used 11 classes with 'BIO encoding': O (not a part of any argument component), Backing-B, Backing-I, Claim-B, Claim-I, Premise-B, Premise-I, Rebuttal-B, Rebuttal-I, Refutation-B, Refutation-I. A sequenced labeling approach was used to identify argument components in the discourse and it significantly outperformed the baseline (0.156) with an overall $macro - f_1$ score of 0.251. Along with this, a feature set based on word embedding in a cross-domain scenario was applied and obtained a $macrof_1$ score of 0.209. Llewellyn *et. al* [108] proposed an approach for classifying social media texts (tweets) into argument types (claims and counter-claims). They annotated eight argument classes tweets (London Riots in 2011) which were used to train SVM Classifier. After that, the developed model was used for classifying tweets into argument structures. This process of investigation was followed by Rosé *et al.* [160] who used the online discussion forum to train the classifier. The task of both Llewellyn *et. al* and Rosé *et. al* was to identify the appropriate features that are useful in predicting different argumentation classes. The results show that SVM performance is much better with tweets (Llewellyn's work) than it was with Rosé's online discussion. The main reason for this was that twitter data contains lots of repetition which causes the machine learning algorithm to overfit to the data. The authors claim that punctuation is the best and most useful feature when adapting a given model to another dataset.

Kang and Saint-Dizier [92] developed a linguistic model for the analysis and portrayal of argument compounds (arguments that are closely related to each other and in a context that is expressed by discourse relations). The chunks of an argument are related to each other through conjunctions, connectors, and various other forms of references and punctuation. To identify argument compounds, discourse grammar was developed. They show that discourse relations can be conceptually characterized so that inferences may be drawn within and between argument compounds.

3.4 Text Corpus Analysis and Statistics

A Text Corpus is a representation of text, language and subject that is annotated with certain signs for a specific purpose. It is essential for carrying out computational research. The performance of predictive system also depends upon the usually it is quality and quantity of the corpus. These corpora are collected from several domains such as Newspaper, Legal, Political, Scientific and Persuasive essays and also in various languages: German, English, Greek, Nepali, etc.

Annotating corpora is complex, expensive and requires experts who are well versed in the corresponding field to ensure that annotations are correct. Palau and Ieven [138] dealt with the theoretical aspect of structure present in legal corpora. They highlighted the different critical points humans need to encounter when applying theory to real argumentation and also emphasized the association between real arguments and the theories that describe those arguments. Mochales and Moens [120] noted that annotation requires expertise in case-law and also in arguments, and finding such expertise is very difficult. Likewise, Stab *et al.* [176] also mention the difficulty of annotating claims and premises in persuasive essays; the structure of a corpus varies depending on the subject matter, and the structure of arguments available in the case-law documents and persuasive essays is rather variable. Even so, several research centers in the world are devoted to developing corpora. One is of it is Arg-tech Centre[2]. This Centre plays an important role in many aspects of argumentation, from theoretical to practical. The goal of the Centre is to develop freely available software tools to aid the researcher in the Argumentation field. Along with these, there are more than 50 corpora from different sectors offered in AIFdb [24]. These corpora are available in different formats, such as SVG, PNG, DOT, JSON, LKIF, RTNL, RDF, PL. There are corpora available on different subjects, and also available in various languages. The annotator can login to the AIFdb system (a web portal) upload the datasets and annotate the corpus. Figure 3.1 shows the interface to AIFdb where arguments (in the Nepali language) are presented. In the figure, it can be seen that components are connected via an 'assert'. The selected components are represented graphically as nodes and arrows in a form of visual programming. A popular corpus named Araucaria [152] was developed in the Arg-tech centre[3]. The granularity of the corpus was 'claim' and 'premise'. The datasets were collected from 19 newspapers (from the UK, US, India, Australia, South Africa, Germany, China, Russia and Israel), 4 parliamentary records (in the UK, US and India), 5 court reports (from the UK, US and Canada), 6 magazines (UK, US and India), and 14 other online discussion boards and ''cause'' sources such as Human Rights Watch (HuRW) and GlobalWarming.org.

[2]http://www.arg-tech.org/
[3]http://www.arg-tech.org/

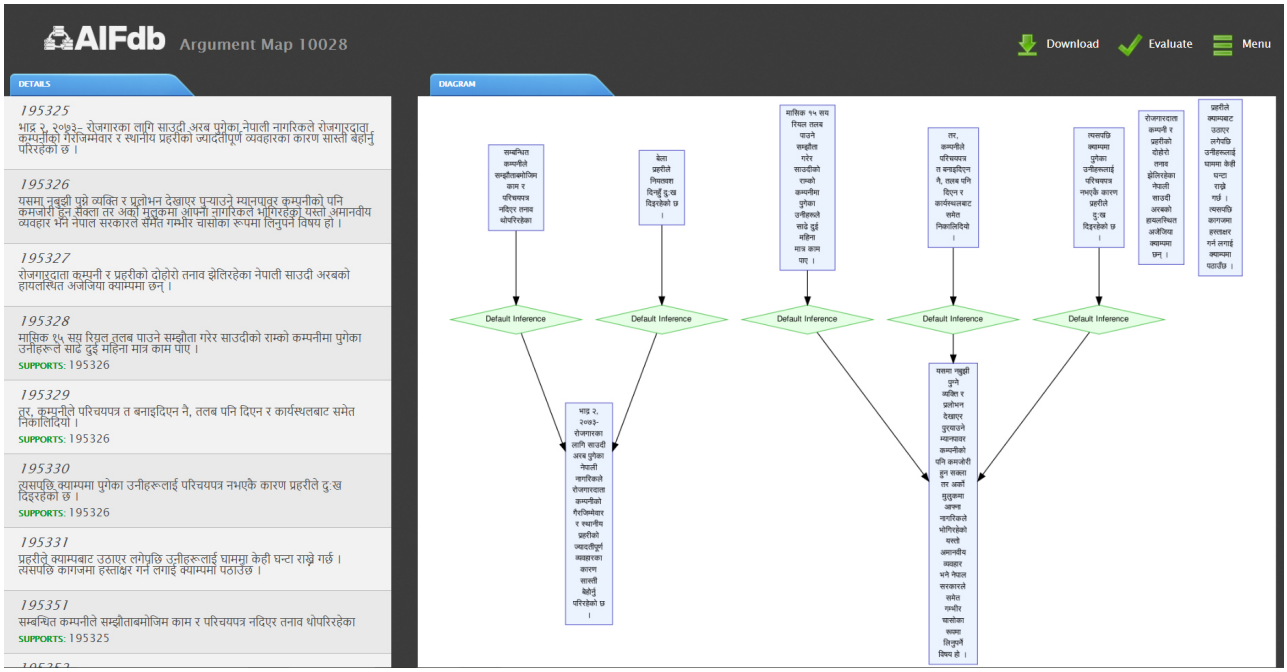

Figure 3.1: Screenshot of an argument from AIFdb [97]

Another, research lab Ubiquitous Knowledge Processing (UKP) Lab [4], TU Darmstadt, Germany is dedicated to developing natural language processing and machine learning tools and techniques. One of their activities is to develop corpora for argument mining in English and German. Habernal and Gurevych [80] created a user-generated web discourse named the *Argument Annotated User-Generated Web Discourse* consisting of 90,000 tokens from 340 documents. Datasets were prepared on the basis of Toulmin's argument classification (Backing, Claim, Premise, Rebuttal, Refutation). The authors compared the results of human and machine detection in the domain of educational controversies (homeschooling, single-sex education, mainstreaming). Due to high amounts of noise in the data sets, the corpus was studied in two ways. First, 990 documents were annotated for further deep analysis of argumentation. Second, various argumentation models were studied and the Toulmin Model was considered appropriate to annotate the corpus. Three persons annotated the documents as either argumentative or non-argumentative. 340 argumentative documents were annotated as claims, premise, backing, rebuttals, and refutation in multiple-sentences.

Other work from Habernal and Gurevych [79] created a user-generated Web content corpus named the *UKPConvArg1 Corpus* from ACL 2016 that consists of 11,650 argument pairs. The same authors [78] created a new crowd-sourced benchmark data-set that contains 9,111 argument pairs labeled with 17 categories. The corpus is tested against in several computational models, both traditional and neural network-based, and their performances, were evaluated quantitatively and

[4] https://www.ukp.tu-darmstadt.de/ukp-home/

qualitatively. Echkle-Kohler *et al.* [58] annotated 88 news documents (German Language) using a claim-premise model. Kirschner *et al.* [95] annotated 24 scientific articles written in German. The authors introduced an annotation scheme and annotation tool named DiGAT which they claim outperforms WebAnno [206].

Corpora are typically divided into two main categories: Macro Arguments which are automatic representations of sentences/phrases or clauses and Micro Arguments which are arguments that have internal structure.

3.4.1 Macro-level Corpora

A macro-level corpus includes the properties of arguments or relations between arguments. These type of corpora are considered as a full argument which are produced on the basis of dialogical communications. These corpora are not annotated with the components of arguments as is done in micro-level corpora.

Next, some web portals that provide Macro-level Corpora are discussed.

Debatepedia: Debatepedia is a Wiki encyclopedia of argument, debates and support quotas. Brooks Lindsay launched Debatepedia at Georgetown University in 2006 [104]. A year later, he merged Debatepedia with the 'International Debate Education Association' (IDEA). The working principle of Debatepedia is similar to that of Wikipedia in that anyone can contribute. The documents are 'pros' and 'cons' of daily life activities, arguments, national politics, vision statements, development, claim, provision, states, business, academic documents, professionalism documents, etc. Some of the debates concern virtual worlds, matters of expression, ethical values, and norms, etc. Articles available in Debatepedia are not final versions, as articles are user-generated based on changes, and updated to improve the content. However, it is a resource for 'pros' and 'cons' of diverse fields. Cabrio and Villata [35] exploited support and attack relations between 200 arguments pairs extracted from Debatepedia.

iDebate.org : This web portal provides storage for online debates [31]. The information is moderated and edited to maintain the highest standards of quality. Arguments are labeled as either 'pro' or 'con' on a topic that contains a set of prominent arguments in a debate. The web portal is maintained by the International Debate Education Association (IDEA) which is a global network that helps youth to establish their voice. The main motto of the organization is to produce a resource for critical thinking and also to help foster cultural exchange among young people. Khalid Al-Khatib [5] used the corpus from this portal to identify argumentative text. The data was collected from 14

cross-domain data sets that contain 28,689 argumentative texts. The advantage of working with cross-domain data is the ability to retrieve arguments from diverse fields and also to be able to measure the performance of a system dealing with multi-dimensional data sets.

proCon.org: procon.org is an online portal for discussion and debate. The web portal consists of more than 50 controversial issues ranging from gun control and the death penalty to illegal immigration and alternative energy. The goal of this web portal is to make people aware of the facts and also to encourage them to think critically about the issues. The forum is run by a nonprofit, nonpartisan public charity organization. The mission of the organization is '*promote critical thinking, education, and informed citizenship by presenting controversial issues in a straightforward, nonpartisan, and primarily pro-con format'* [147]. Its portal has become the USA's leading source for information and civic education. The organization serves more than 25 million people each year including students and teachers in more than 9,000 schools all over the United States and 90 foreign countries. The web portal became a pioneer resource for journalists to get information regarding the people's voice. This forum became a pioneer resource for journalists to get information regarding "the people's voice", and from 2008 onwards, it has been a major contributor to the presidential election of United States. The organization, which was founded on July 12, 2004, runs entirely through donor support.

4Forums.com, CreateDebate.com and ConvinceMe.net: Walker *et al.* [195] developed a larger scale dialogical corpus called the *Internet Argument Corpus* or IAC. They collected 390,704 posts in 11,800 discussions extracted from the online debate site 4Forums.com, which is concerned with Economic and Tax Debates. After four years, they released IAC 2.0 [1]. This time they included debates from CreateDebate.com and ConvinceMe.net and structured them into a novel data schema in SQL. The dataset consists of 65,368 posts in 5,413 debates by 5783 authors from CreateDebate.com; 2958 posts, 16,671 sentences, and 275,472 tokens from createdebate.com and 41,4000 posts in 11,000 debates by 3,500 authors from 4Forums.com. The corpus is related to controversial issues such as gun control, abortion, the existence of God, and gay marriage. These documents are self-annotated with user tags which users themselves can vote 'for' or 'against'.

ProCon.org and iDebate.org: Similarly, Boltuzic and Snajder [31] presented the COMARG (comments with argument) corpus that is freely available for research purposes. The corpus was developed for the purpose of training and evaluating argument recognition models. The data source was from ProCon.org (which contains user comments) and iDebate.org (which contains the arguments). To maintain a large number of comments and also to maintain a good balance between 'pro' and 'con' stances, two topics were selected: Under God in the Pledge - ProCon.org (UGIP) which has 175 comments and six arguments, and Gay Marriage (GM) which has 198 comments and seven arguments.

3.4.2 Micro-level Argument Corpus

A micro-level argument corpus deals with the internal structure of arguments. It is common practice to divide an argument into two components: premise and conclusion, but in the scientific arena of argument mining, there several more divisions appear. Toulmin [184] uses six categories (Facts, Warrant, Backing, Qualifier, Rebuttal) and Stab *et al.* [176], divide components into 4 categories: major premise, minor premise, claim and none. Some of the corpora that are available are described below.

Kwon *et al.* [96] developed the public's comments about the Environmental Protection Agency (EPA)'s standards rules on hazardous pollutants. Two annotations were involved to categorize 119 documents, and they achieved an agreement of Cohen's Kappa coefficient =0.62. Biran and Rambow [28] used three corpora (RST Treebank [40], English Wikipedia and LiveJournal [107]) in different stages of the development of their system. They used 309 blogs threads from LiveJournal to annotate 1,377 multi-sentence argument components using a claim-premise model. A RST Treebank was used to find the relation between indicators or discourse markers of arguments. Lastly, 'English Wikipedia' was used for unsupervised word pair extraction.

Rosenthal and MaKeown [161] created a corpus from two datasets: 285 LiveJournal blog[5] spots and 51 Wikipedia discussion forums. The datasets are annotated for the purpose of identifying claim only. The ratio of claims vs. not claims is 60:40 in LiveJournal and 64:36 in Wikipedia. Aharoni *et. al* [3] present an argumentative structure dataset consisting of 33 controversial topics. The corpus is derived from 586 Wikipedia articles. The author insists that the corpus was constructed (manual annotation) with great attention to detail.

Goudas *et al.* [71] annotated 204 documents in Greek related to renewable energy. The documents contain 16,000 sentences and were collected from social media, news, blogs, and microblogs. 760 sentences were annotated with premise and claim at the clause-level. Similarly, Sardianos *et al.* [164] choose 300 news items (sports, politics, economics, and culture, etc.) in the Greek language to annotate. Two post-graduate students were assigned to annotate the corpus and since they were only moderately experienced, guidelines were provided to describe the identification of arguments in which annotators were focused on discourse markers such as because, in order to, and but. Each annotator was assigned to annotate 150 documents with argument components, and the final version of the corpus contained 1191 argument components.

Persing and Ng [143] present a different strategy in which they propose a model to find the strength of arguments in student essays. The authors annotate the 1000 essays from the 'International Corpus

[5] https://www.livejournal.com/

of Learning English' (ICLE) [73]. Their dataset was divided into three parts: 60% for model training, 20% for parameter training, feature selection and spotting, and the remaining 20% for testing.

Mochales and Moens [121] developed the *European Court of Human Rights* (ECHR) corpus (see Chapter 4 for more details). Similarly, Reed *et al.* [152] used newspaper, and court cases as source material. Stab and Gurevych [172] used persuasive essays and scientific articles. Likewise, Feng and Hirst [61] used newspapers and court cases as a corpus to investigate argumentation schemes. Llewellyn *et. al* [108] extracted arguments from social media.

3.5 Summary

In this chapter, the state of the art regarding extraction of the arguments from various domains is presented. It is clear from the literature that research on this subject has grown in a very rapid manner, and that this is an area that is being actively researched in the legal domain. Mochales and Moens [121] work in this field is noted. The history of argumentation up to the modern day is briefly recapped, and then the state of the art approach using three subtasks for argument mining is introduced. The clustering approach is examined in more detail. At the end of the chapter, the corpora used by several argument mining investigations are discussed.

4

ECHR Corpus

Texts can be categorized into subjects such as law, philosophy, computing, science, etc., and annotated with information appropriate for the purpose of the research (e.g. argument mining, name entity recognition, etc.). This information is collectively called a corpus and forms a critical component of this research. The corpora are a source of knowledge for creating certain rules & regulations (i.e. a model) which are used in statistical and hypothetical tests. Although influenced by many other factors, performance of statistical approaches still depends primarily on the quality and quantity of the corpus. Therefore, during the process of creating a corpus, annotators need to prioritize their ability to maintain quality and quantity via inter-annotator agreement [10]. While creating corpora, it is desirable to ensure that a system will achieve the highest possible accuracy. Nevertheless, creating and constructing corpora is labor-intensive, as it is complex, time-consuming and requires experts who need to be well versed in the corresponding corpus field. For example, in the case of legal corpora, the annotating experts must be lawyers and should also be familiar with

the legal arguments. Within these constraints, there are a limited number of corpora available in the respective fields. The Language Intelligence and Information Retrieval (LIIR) research lab, KU Leven, Belgium kindly provided us with a copy of their ECHR corpus, but only in printed format. This chapter deals with transferring the ECHR corpus [122] into electronic format. In addition, the structure and the complexity faced during the process of annotation is discussed.

4.1 Historical Background of ECHR Court

After the Second World War the European countries decided to establish the civil and political rights Common Court [44]. Twelve states signed the contract to establish the court in 1959 in Strasbourg, France. The court was named The European Court of Human Rights. In later years the number of states gradually grew to reach 50 states. The judges are elected from the parliamentary assembly of the Council of Europe where a single judge represents each state. The court receives hundreds of applications every day but these applications need to be validated because there are many cases that does not fulfill the requirements of the court. There are four categories of the judiciary: Single (1 Judge), Committee (3 judges), Chamber (7 Judges), and Grand Chamber (17 Judges) [1].

The comprehensiveness of the court has expanded to non-European people with issues and it now deals with a large variety of cases such as migration, asylum requests, violations within European countries. Cases are related to the different propositions, and some of them have waited for more than 20 years to be trialed in the national courts. Cases should be dealt with at the national level if the fundamental rights of the individual concerned are not recognized, as should cases that are related to discrimination civil partnerships, environmental issues and also unauthorized access to technology or politically sensitive issues. These listed violations are samples of thousands of the cases in ECHR. The member governments must take action to ensure that the ECHR's Convention is respected at a national level. After the court makes a decision, the states (foreign ministry of each state council) need to abide by it and apply the necessary remedial actions [44].

4.2 Statistics of the ECHR Corpus

Case-law documents are written using detailed information from the stakeholders of the court, factual information from the defendant, allegations made by the plaintiff, arguments from both parties, and a decision made by the judge. After collecting all information, it needs to be structured to be

[1] http://www.ijrcenter.org/european-court-of-human-rights/

useful. To structure the data and also to know the location of the components of the arguments, it is necessary to analyze and determine the content of each section available via case-law. From the website[2], it is known that case-law is divided into seven categories: judgments, decisions, communicated cases, legal summaries, advisory opinions, reports and resolutions. Of these seven categories, two categories (Judgment and Decision) are found in the ECHR Corpus.

There are 20 decision categories (released before 20 October 1999 by the European Commission on Human Rights), and 22 judgments issued by a chamber of seven judges ruling on the admissibility and merits of the cases available in the ECHR Corpus. Sample files of judgment and decision categories are presented in the appendixes A and B respectively. Both categories represent similar information, however, the 'Decision category' presents the information briefly (the average word length is 3500 words) in the corpus whereas, in the case of Judgments, more detailed information is available (an average word length of 10000 words). The Decision case-law documents are divided into six sections: i. Introduction, ii. The Facts, iii. Complaints, iv. Proceedings before the Commission, v. The Laws and vi. For the Reason. Judgment case-law documents are divided in eight sections: i. Introduction, ii. Procedure, iii. As the Facts, iv. The Circumstances of the Case, v. Proceedings before the Commission, vi. Final Submissions to the Court, vii. As to the Law, and viii. For the Reason.

The case-law documents begin with introducing the stakeholders of the courts (President, Judge, Registrar and Deputy Registrar, Lawyers, Plaintiff, Defendants, and their agents), with their designations, plaintiff, defendant and other members involved in the case. After this, procedure and facts regarding the plaintiff are described. The facts describe an overview of the case that includes information from previous cases, the reason for making an allegation and the chronological sequence of events. The structure of the case-law varies depending upon the exact laws. This information is included as necessary, meaning that the case will vary in length depending upon the case-law and any other essential information that is included. After providing facts of the plaintiff and defendant, the case-law include the discussions held in the court based upon the allegation made by the plaintiff are presented. Likewise, the defendant provides the reason and claim from their perspective and returns a response to the claim made by the plaintiff. After several discussion and arguments presented by both parties, the Judge renders his decision.

PROCEDURE

1. The case was referred to the Court on 4 December 1995 by the Government of Turkey ("the Government") and on 12 December 1995 by the European Commission of Human Rights ("the Commission"), within the three-month period laid down by Article 32 para. 1 and Article 47 of the Convention (art. 32-1, art. 47). It originated in an application (no. 21987/93) against the Republic of Turkey lodged with the Commission under Article 25 (art. 25) on 20 May 1993 by Mr Zeki Aksoy, a Turkish citizen.

The Government's application referred to Article 48 (art. 48); the Commission's request referred to Articles 44 and 48 (art. 44, art. 48) and to the declaration whereby Turkey recognised the compulsory jurisdiction of the Court (Article 46) (art. 46). The object of the request and of the application was to obtain a decision as to whether the facts of the case disclosed a breach by the respondent State of its obligations under Articles 3, 5 para. 3, 6 para. 1 and 13 of the Convention (art. 3, art. 5-3, art. 6-1, art. 13).

2. On 16 April 1994 the applicant was shot and killed. On 20 April 1994 his representatives informed the Commission that his father wished to continue with the case.

3. In response to the enquiry made in accordance with Rule 33 para. 3 (d) of Rules of Court A, the applicant's father (who shall, henceforward, also be referred to as "the applicant") stated that he wished to take part in the proceedings and designated the lawyers who would represent him.

On 26 March 1996 the President granted leave, pursuant to Rule 30 para. 1, to Ms Françoise Hampson, a Reader in Law at the University of Essex, to act as the applicant's representative.

Figure 4.1: screen shot of case-law

4.3 Preprocessing

Case-law documents are divided into several sections (as described in section 4.2) however overall, the case-law documents are unstructured and several types of preprocessing are performed on the corpus to increase its quality. The main way to do this is to remove irrelevant information, of which there are three kinds: The first is the initial section of the documents that consists of the name and designation of the stakeholders; second, are the index numbers within section titles (as shown in Figure 4.1). The figure illustrates the section titled PROCEDURE and contains the index numbers 1, 2 and 3; third is the section title itself.

Only the remaining sentences from each case-law are used for experimental purposes. These sentences were separated by using Stanford NLP tool kit [113]. Furthermore, we replace all types of date with keyword DATE and the numeric values of rules and regulations are replaced with the keyword NUMBER. The advantage of this is to increase consistency and to create unique identifiers for training or notifying the classifier about a date or rule and regulation.

[2]https://hudoc.echr.coe.int/eng

4.4 Annotation

The ECHR Corpus was developed by Mochales and Moens [120] and the annotation procedure is described in Section 4.4.1. However, since, an electronic version of the corpus was unavailable, the process of transferring the hard copy electronic form is detailed in Section 4.4.2.

4.4.1 The First Version

Mochales and Moens [120] hired two lawyers to annotate the ECHR case-law documents. The annotators were given an argumentation scheme formalism and guidelines that describe the arguments. Once annotation was completed, they were compared and found to score an inter-rater agreement tally of 58% according to the Kappa measure. A third lawyer was selected to analyze the annotations and found that the main reason for the discrepancies was due to a different demarcation of argument boundaries or, put another way to the ambiguity that is found in argumentative structure. Subsequently, a fourth annotator was selected and was given new guidelines, new sets of comments and recommendations. His annotation achieved 80% agreement, which was quite a significant gain. This ECHR corpus was used in following [152, 126, 122, 120, 139, 138, 121, 204] publication.

4.4.2 The Second Version

As mentioned at the beginning of the chapter, the ECHR corpus was received from the LIIR research lab. Out of 43 case-law documents (in hard copy), one case-law was in French (which was omitted). The received document is a list of arguments separating the premises and conclusions of each case-law. The documents did not include the non-argumentative sentences and also no indication of the relations between the components (premise and conclusion) or between the arguments. A screenshot of the corpus is shown in the Figure 4.2. A sample of the annotation of a case-law is shown in Figure 4.3: text highlighted in orange highlight is a premise and the sentences highlighted in light green are a conclusion. The annotation $6p_1$ means that it is the first premise of the sixth argument and $6c$ means that it is the conclusion of the same argument.

There is a dual nature to components, such that the premise of one argument can be a conclusion of, or a premise to another argument. As observed in the Figure 4.3, the sentence *In these circumstances, the Commission finds that the applicant's complaint to the Constitutional Court about ill-treatment does not constitute an effective and sufficient remedy for the purposed of exhaustion of domestic*

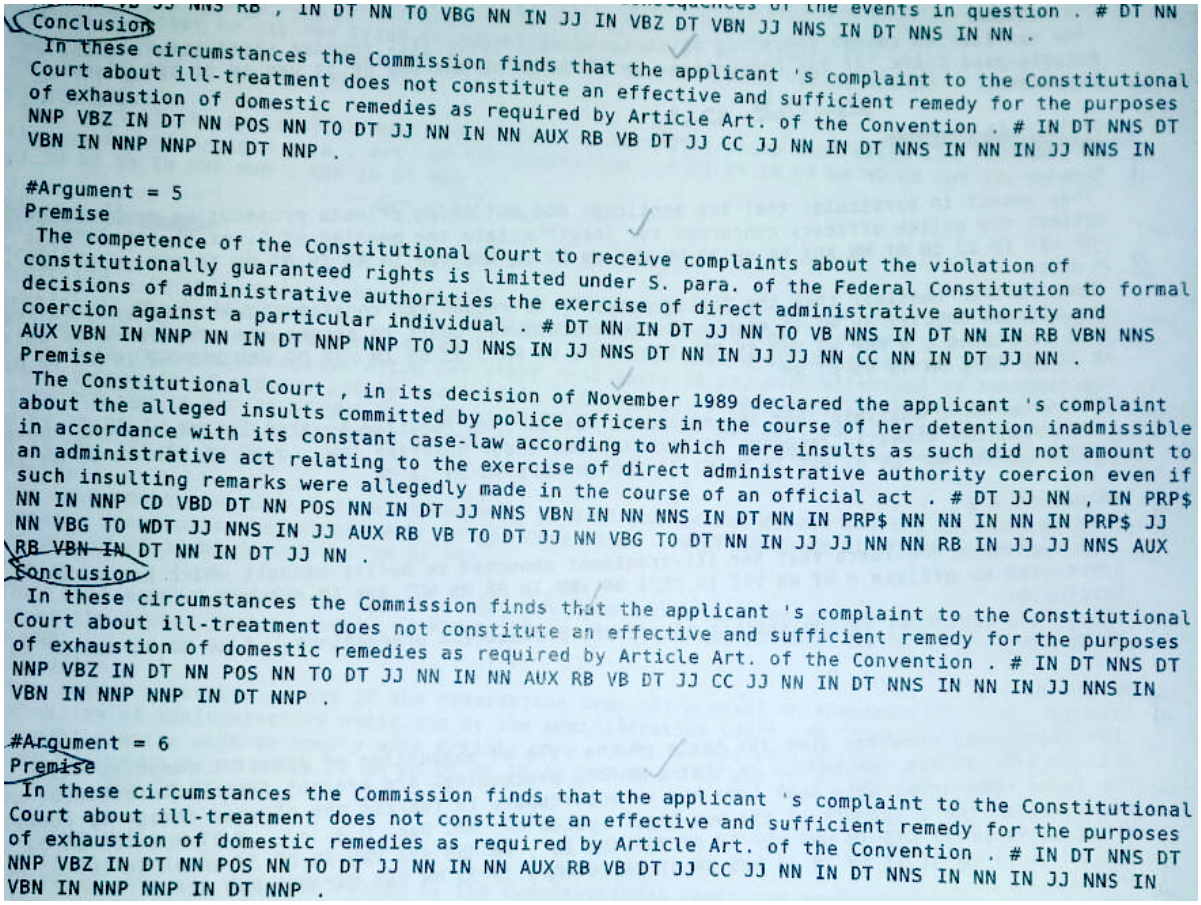

Figure 4.2: screen shot from sample case-law received from the LIIR lab [127]

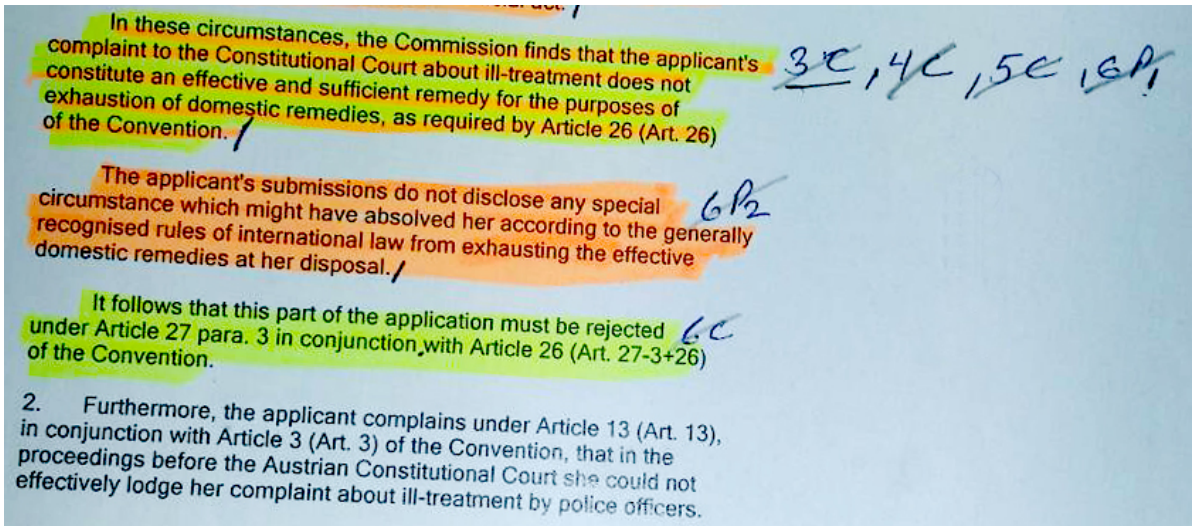

Figure 4.3: Sample case-law with annotation

remedies, as required by Article 26 (Art. 26) of the Convention is the conclusion of the third, fourth and fifth arguments, and also the first premise of the sixth argument.

After eliminating unnecessary information as described in Section 4.3, the corpus is composed of 9257 sentences, of which 7097 (77%) are non-argumentative and 2160 (23%) are argumentative sentences. These argumentative sentences were further tagged as premises and conclusion leading to a set of 1828 premises and 657 conclusions (i.e. some sentences are premises/conclusions for more than one argument, and some sentences are both premises of one argument and conclusion for another). The average word length and the average number of words in each sentence of the corpus are 4.98 and 27.79, respectively. There are 28,6341 words in the corpus of which 21,0355 words are from non-argument sentences and 75,986 are from arguments.

The Commission concludes that this application cannot be rejected for non-exhaustion of domestic remedies under Articles 26 and 27 para. 3 (Art. 26, 27-3) of the Convention.
Six months' time-limit
The Commission has examined whether the applicant has complied with the requirement imposed by Article 26 (Art. 26) of the Convention that an application must be introduced within six months of the final decision taken in respect of the complaints.

Table 4.1: Example of Complex Argument from ECHR Corpus

Further, there are 47 sentences that are combinations of non-argumentative phrases and argumentative phrases. For instance, non-argumentative phrases emerged in between sentences. Such phrases are neither the section titles nor expressions of argumentative or non-argumentative sentences. Such activities can be observed in the Table 4.1. The phrases *Six months' time-limit* is a 'pop up' in between the sentences. The phrase has neither a full stop nor any notification symbol to separate it from the second sentence.

The Stanford NLP toolkit [113] was used to split sentences. After that, annotation of each sentence was undertaken. The annotation procedure is shown in the Figure 4.4.

Case Law File number: There are altogether 42 case-law files. Each sentence of a case-law file is labeled from 00 to 42 (except case-law file number 36, which was rejected due to being in the French Language).

Case-law Type: Since there are two types of Case: Judgment and Decision, information was included to distinguish between Judgment and Decision corpora. Therefore, sentences are labeled capital letter 'D' for Decision Corpus and capital letter 'J' for Judgment.

Section Type (Other and Law): The case-law file document is divided into two categories: 'Other'

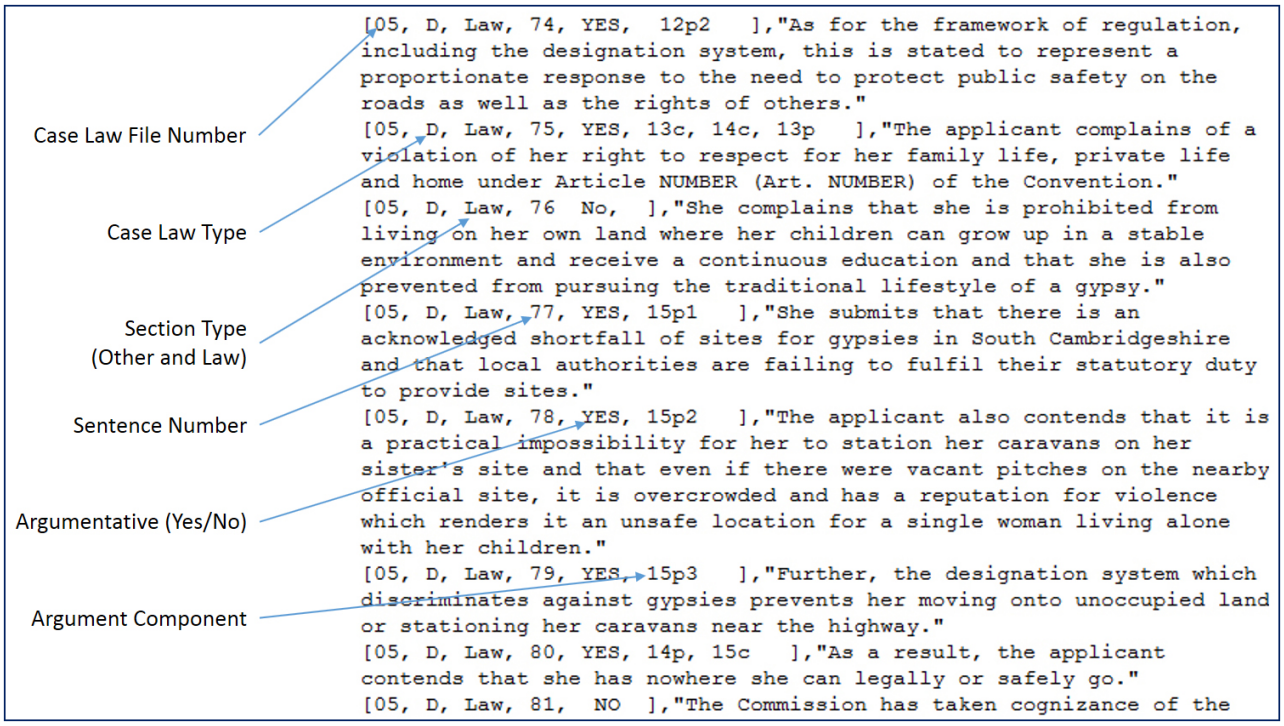

Figure 4.4: Annotation Procedure of the ECHR Corpus

and 'Law'. Other refers to sentences that belong to all the sections except The Law section. Law refers to sentences that belong the 'The Law' (Decision category) and 'As to the Law' (Judgment category).

Sentence number: Sentences are numbered for each case-law file.

Argumentative (YES/NO): Tags a sentence as being argumentative or not.

Argument Component: The sentence is tagged with the argument number, then later, if the sentence is a premise, annotated with a lowercase letter p followed by the premise number. If the sentence is a conclusion, then the sentence is tagged with a lower case letter c. For example, looking at Figure 4.4, $15p_3$ means that the sentence belongs to the argument 15 and is its third premise.

4.5 Dataset Structure

Premises and conclusion of arguments can be in sequence and also scattered in case-law documents. Corpora of this nature are complex and yield low accuracy results when attempts are made to extract their arguments. The position of an argumentative sentence that comprises an argument also has an impact on feature and argument identification. Information almost always flows in sequential order. However, knowledge is obtained through references to different sections of the corpus, and

not necessarily in a sequential manner. Background information is necessary for such cases but automatically gathering background information is very difficult, making identifying argumentative sentences that are scattered around a text quite a challenging task. Despite this, it is found that argumentative sentences can be categorized as sequential sentence argument, single sentence argument, scattered sentence argument and duality of sentence structure. Each of these situations is discussed below.

In the ECHR Corpus, most of the argumentative sentences of a particular argument are in sequential order, which means that premises follow one after another, finally ending with a conclusion. In Table 4.2 there are three columns: the first column is the sentence number; the second column lists the actual sentences; in the third column, the argumentative sentence ids are listed.

Sentence Number	**Sentence**	AS ID
96	Mr. Pfarrmeier contended that none of the bodies that had dealt with his case in the proceedings at issue could be regarded as a 'tribunal' within the meaning of Article 6 para. 1 (art. 6-1).	$13c$
97	This was true not only of the administrative authorities, but also of the Constitutional Court, whose review was confined to constitutional issues, and above all of the Administrative Court.	$12p1$
98	The latter was bound by the administrative authorities' findings of fact, except where there was a procedural defect within the meaning of section 42(2), sub-paragraph 3, of the Administrative Court Act (see paragraph 21 above).	$12p_2$
99	It was therefore not empowered to take evidence itself, or to establish the facts, or to take cognizance of new matters.	$12c, 13p_1$
100	Moreover, in the event of its quashing an administrative measure, it was not entitled to substitute its own decision for that of the authority concerned but had always to remit the case to that authority.	$13p_3$

Table 4.2: Example of sequential sentence argument

Similarly, some of the sentences are themselves arguments. In the Table 4.3, the sentence is divided into three parts, the underlined text is the first premise, bold text is a conclusion, and italic text is the second premise.

Furthermore, there are some arguments which have their component dispersed throughout the document. Such arguments are known as 'Scattered Sentence Argument'. Table 4.4 presents the argumentative sentence (premise or conclusion) of an argument (i.e. argument number 24 of case-law

Sentence
The applicant next complains under Article NUMBER para. NUMBER (Art. NUMBER) of the Convention of **an unfair hearing in the determination of the criminal charges against him,** *in that , allegedly, insufficient reasons were given by the courts to justify his conviction.*

Table 4.3: Example of single sentence as argument

documents). The first premise is the 69^{th} sentence of case-law, the second of the 81^{th} sentence, the third of the 86^{th} sentence and the fourth the 94^{th} sentence, while the conclusion is the 99^{th} sentence.

Sentence No.	**Sentence**	AS ID
69	The Commission notes that the applicant's conviction involved his writings.	$24p_1$
81	The Commission considers that this indicates an issue falling within the scope of freedom of expression.	$24p_2$
86	The Commission concludes that the applicant has complied with the requirements of Article NUMBER (Art. NUMBER) of the Convention.	$24p_3$
94	These complaints cannot therefore be regarded as manifestly ill-founded within the meaning of Article NUMBER para. NUMBER (Art. NUMBER) of the Convention, and no other ground for declaring this part of the case inadmissible has been established.	$24p_4$
99	For these reasons, the Commission, by a majority, DECLARES ADMISSIBLE, without prejudging the merits of the case, the applicant's complaint that his conviction for having disrupted public peace amounts to a violation of his rights set forth in the Convention; DECLARES INADMISSIBLE the remainder of the application.	$24c$

Table 4.4: Example of scattered sentence argument

As for the sentences showing duality, i.e. sentences that are premises or conclusions of one argument while simultaneously being the premise or conclusion of another, different argument are annotated as *sentence dual nature*. In Table 4.2, it can be seen that the sentence *It was therefore not empowered to take evidence itself, or to establish the facts, or to take cognizance of new matters* is the conclusion for argument number 12 but also the first premise for argument number 13. The complexity of this kind of sentence is reflected in the difficulty of determining exactly which components belong to which arguments and results in generally lower accuracy scores.

4.6 Premise and Conclusion Structure

The components of arguments are premises and conclusions. There is a symbiotic relation in between them to form an argument. Three kinds of a structure were found in the case-law: Single Annotation, Overlap Annotation, and Sentence Partition. Table 4.5 shows that type of annotation with its frequency of appearance. Each of the categories is described below.

	Type of annotation Sentence	**Number of annotations**
Single Annotation		
	Single Conclusion	281
	Single Premise	1399
Overlap Annotation		
	Conclusion Overlap	35
	Premise Overlap	27
	Premise and Conclusion	164
Sentence Partition		
	Premise and Premise	55
	Premise and Conclusion	151
	Conclusion and Conclusion	1
	Non-argument and Conclusion/Premise	47
	Total number of Sentences	**2160**

Table 4.5: Annotation types showing frequency of occurrence

4.6.1 Single Annotation

An argumentative sentence that is annotated with only one component of an argument is designated as a Single Annotation. There are 1680 single annotation sentences indicated by the manual annotation process performed on the corpus.

In Table 4.6, the first sentence is a conclusion of the argument number eight (i.e. 8c), the second sentence is the premise of the argument number eight ($8p_1$) and also a conclusion for argument number nine, ten and eleven ($9c$, $10c$ and $11c$), the third sentence is the second premise of argument eight ($8p_2$). Single Annotation in this case means that the first and third sentences are annotations of only one component of an argument.

Sentence	AS ID
The applicant maintained that there was no requirement that he pursue domestic remedies further than he did by telling the public prosecutor that he had been tortured by his custodians.	$8c$
The applicant submitted that any purported remedy is illusory, inadequate and ineffective.	$8p_1$, $9c$, $10c$, $11c$,
He did not deny that the procedures identified by the Government are formally part of the Turkish legal structure, but he contended that the Government have not shown how such procedures could conceivably be effective for the specific circumstances of the present case.	$8p_2$

Table 4.6: Example of Single Annotation Sentence

4.6.2 Overlap Annotation

An argumentative sentence that is annotated as the premise or conclusion of one argument and annotated again as premise or conclusion of another argument is termed an Overlap Annotation. There are altogether 226 such sentences. The second sentence of Table 4.6 is an example of an Overlap Annotation sentence. The sentence is a premise of argument eight, and the conclusion of arguments 9, 10 and 11 as shown by a notation $8p_1$, $9c$, $10c$, $11c$.

4.6.3 Sentence Partition

Out of 2160 argumentative sentence, 254 of them are found to be partitioned into components of the argument (i.e. premise and conclusion within the argument), or from a different argument. Hence, such sentences are designated as Partitioned. The argumentative sentence is formed of independent clauses, either premises or conclusions. In between these clauses, certain specific words punctuation marks are found that differentiate their components. These are listed in table 4.7. Numerical references, alphabetized and bulleted lists can also be perceived as separators. Some of them are discussed with the examples, to elucidate the partition structure within a sentence.

In the legal document, there are several references as a premise for the claim. The Table 4.8 shows

'that', 'because', 'and', 'since', 'as', 'therefore', 'was', 'if', 'of', 'is', 'did', 'find', 'thereby', 'which', 'judgment', 'were', 'restrictive', 'reports', 'contracting', 'not', 'such', 'issue', 'an', 'on', 'excessive', 'become','have', '(see', '(cf.', 'e.g.', 'Eur.', 'para.', 'art.','comma', 'colon', 'semicolon', 'quotation marks'

Table 4.7: List of words that differentiate the partition categories sentences

the example of *argumentation by citation*, in which the reason for the claim is shown via a case-law link.

Sentence	AS ID
The notion of security of person has not been given an independent interpretation *(see in this respect Selçuk and Asker v. Turkey, nos. NUMBER, Commission's report of DATE, §§ 185-187).*	**6c**|*6p*

Table 4.8: Example of Partition Categories (Partition by '(see')

As can be observed in the example, the bold text **The notion of security of person has not been given an independent interpretation** is not a complete sentence (phrase) but it is a conclusion of the sixth argument, whilst the remaining part of the sentence, in italics *(see in this respect Selçuk and Asker v. Turkey, nos. NUMBER, Commission's report of DATE, §§ 185-187)* is a premise of the sixth argument. The reason/premise for the claim/conclusion $6c$ is a referral to a case-law link/citation which is annotated as $6p$. During the process of annotation, the annotator marks the clause of the sentence as the conclusion of the argument, after finding the premise that is referred to.

'therefore': The word *therefore* alters the clause within the sentence, which is mostly used to give the consequences and transitions or to connect ideas/consequences. Table 4.9 shows an example of using 'therefore' to separate the components of a premise (in boldface) and a conclusion (in italics).

However, the application was lodged with the Commission on DATE, five months after the cassation decision and,therefore *within the six month time-limit provided for by Article NUMBER (Art. NUMBER) of the Convention.*	$14p, 14c$

Table 4.9: Example of Partition Categories (Partition by Therefore)

Punctuation Marks: Punctuation marks (comma, semicolon, colon, and quotations) are used to separate components. A colon is used to separate clauses and also often to separate a title from a list of information. The information is within the quotes, and while separating sentences, other non-arguments are associated with each other. As can be seen in table 4.10, each premise is separated by a semicolon and comma.

Sentence	AS ID
According to the applicant, there was no 'pressing social need' to ban a video work on the uncertain assumption that it would breach the law of blasphemy; indeed, the overriding social need was to allow it to be distributed.	$11c, (;)11p1$
To demonstrate that the available remedies were not ineffective, the Government have referred to some judgments by the administrative and criminal courts.	$14c, (,)14p1$
It is moreover undisputed that this interference was 'prescribed by law', the applicant's conviction being based on Articles NUMBER (b) and 23 (1) of the Penal Code.	$1c, (,)1p1$

Table 4.10: Example of Partition Categories (Partition by Punctuation)

Alphabetized list: In the gold-standard data, there are several paragraphs consisting of phrases/sentences in list form. Each element of the list is a premise or a conclusion. During the sentence partitioning, since each item is within the list, it is not separated. Therefore, all items of the lists appear within the sentence. Table 4.11 has three items (a),(b),(c); each of them is premise and (a) is the conclusion.

Sentence	AS ID
According to the Government, the applicants failed to exhaust their domestic remedies (a) by not having applied for judicial review either of the Inspectors' conduct of the inquiry or of their decision to submit their report to the Secretary of State; (b) by not having applied for judicial review of the Secretary of State's decision to publish the Inspectors' report; and (c) by not having pursued the libel proceedings commenced against The Observer newspaper.	$2c, 2p1, 2p2, 2p3$

Table 4.11: Example of Partition Categories (Partition by Alphabetized list)

Furthermore, there are 75 sentences where one of the phrases belongs to the 'Sentence Partition' category while another phrase is 'Overlap Annotation'. As can be seen in the Figure 4.5, half of the sentence (in orange and green) is a conclusion of argument three and four and the rest of the sentence (in green) is a premise of argument four.

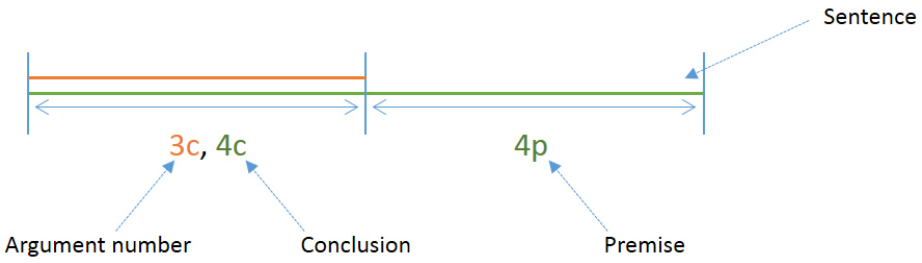

Figure 4.5: Overlap and Partition Sentence Structure

4.7 Summary

In this chapter, the ECHR Corpus is described in detail. As mentioned earlier, converting the provided in the chapter, the passage from hard copy to an electronic version created a second version of this corpus. The procedure undertaken for annotating the components of the arguments in the case-law is described, and an analysis of the quality of the corpus, including its statistical nature and structure, is offered.

5

Proposed Architecture

In this chapter, the proposed architecture of the system is described. The goal of the system is to identify arguments in a legal corpus. At first, argumentative sentences are identified, then these identified sentences are grouped to form arguments. Later, each argumentative sentence is classified as either a premise or a conclusion. To accomplish this task, independent modules are created and are then interlinked with each other. There are altogether three modules, as shown in Figure 5.1, working sequentially to accomplish the requirements of the system. The modules are the Argument Element Identifier (AEI) discussed in Section 5.1, the Argument Builder (AB) discussed in Section 5.2, and the Argument Structurer (AS) discussed in Section 5.3.

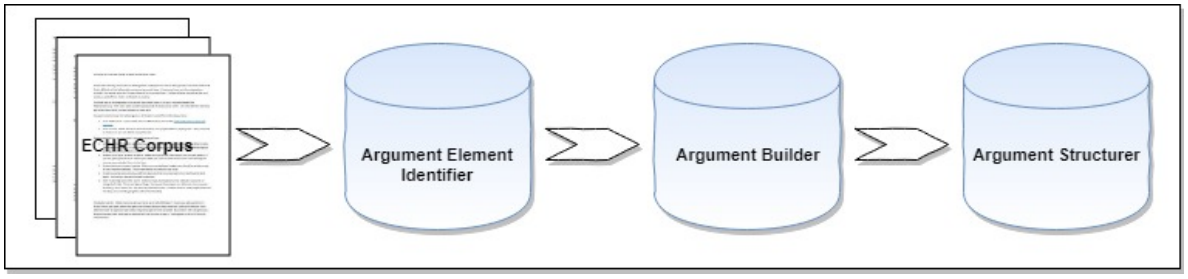

Figure 5.1: Working Principle of the Proposed Architecture

5.1 Argument Element Identifier

This is the first module of the proposed system and its main goal is to find an optimal ML Model, with appropriate features, to distinguish argumentative and non-argumentative sentences. Figure 5.2 presents an overview of the module; each sentence of the case-law documents is labeled as either argumentative or non-argumentative. To achieve this goal several procedures are conducted; the details of the proposed architecture of the module are shown in the Figure 5.3. First, the text is refined by applying several low and high-level preprocess techniques (explained in Section 2.2); second, the Stanford NLP toolkit [113] is used to separate sentences from the narrative text; third, several types of features are extracted, and on the basis of these features, several classifiers are built to obtain the most accurate results.

On the basis of the type of features extracted and the classification algorithm, the research was divided into three parts: Basic, Multi-feature and Tree Kernel. The features of Basic and Multi-feature are used to construct a vector space representation and Tree Kernel uses as input to create a syntactic representation of each sentence.

5.1.1 Basic Experiments

The basic experiments aim to identify an efficient machine learning algorithm that perform the optimally detecting argumentative sentences within legal documents. A 'bag of words' approach with TF-IDF measures normalized to the unit length was used. The top informative features were selected and their performance is measured. The following ML algorithms were used: a polynomial kernel SVM algorithm with various values for the complexity (C) parameter (0.001, 0.01, 0.1, 1, 10, and 100) and a Random Forest algorithm with 7, 11, 17, 50, 100 numbers of trees. The goal of this investigation is to create a baseline.

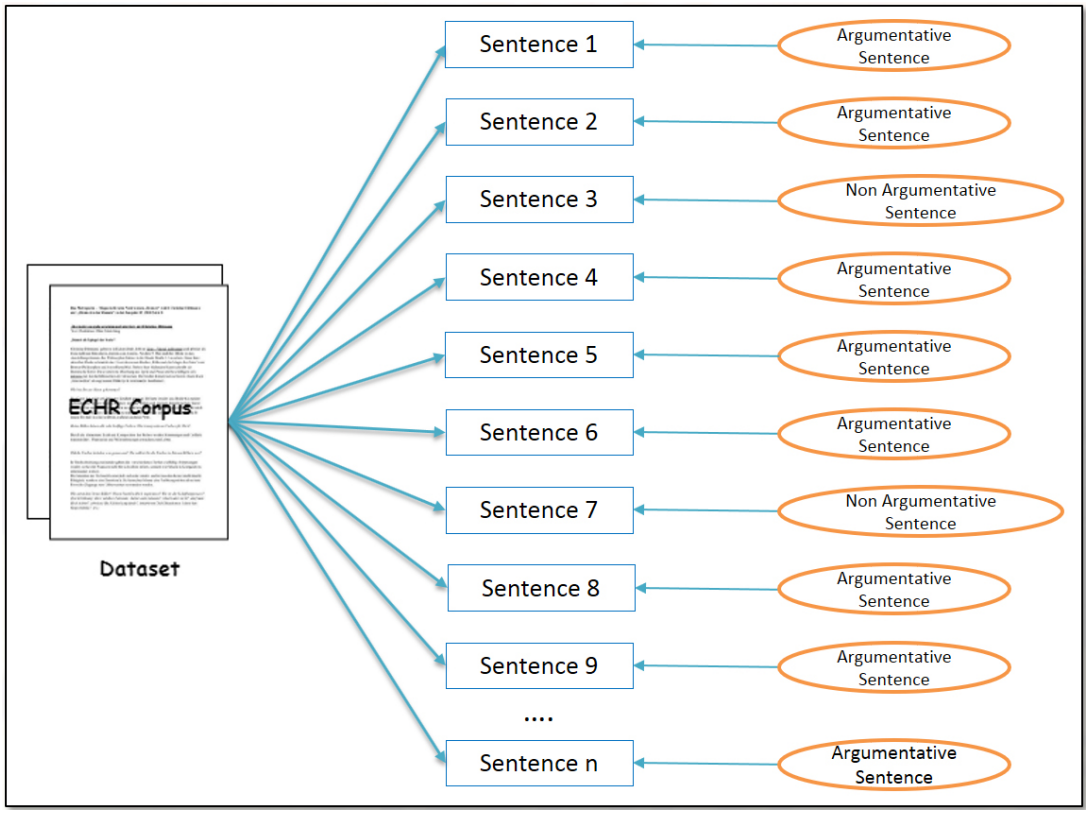

Figure 5.2: Overview of the Argument Element Identifier

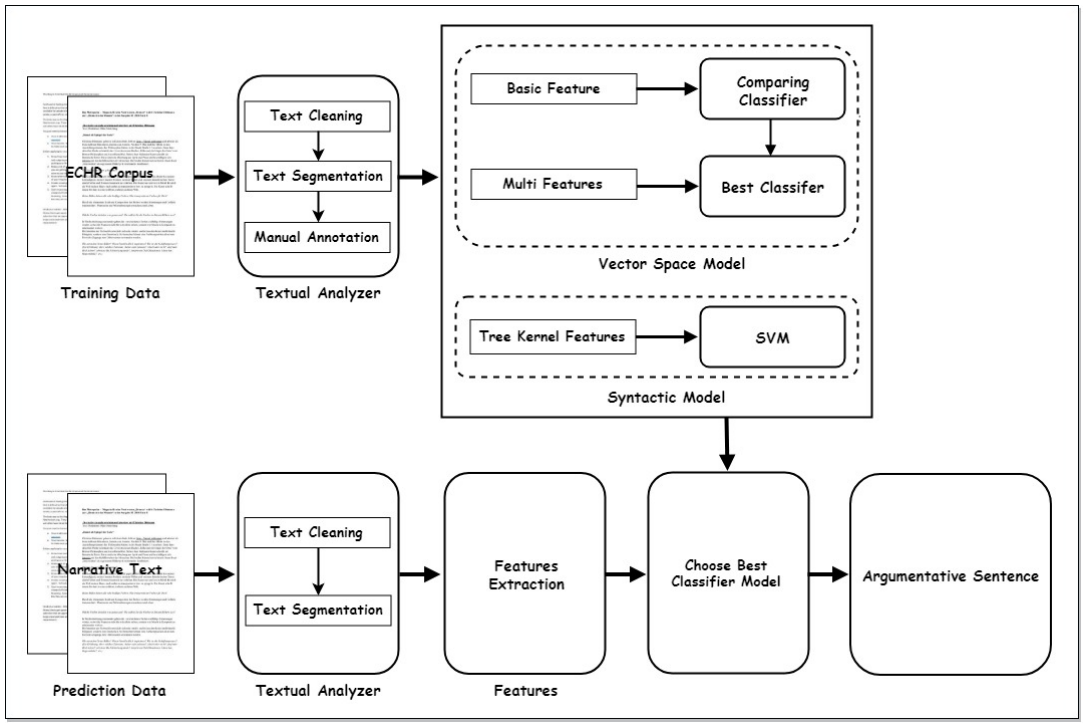

Figure 5.3: Proposed Architecture of Argument Element Identifier

5.1.2 Multi-Feature Experiments

After identifying the most suitable machine learning algorithm for the corpus, a set of experiments were defined based on lexical features, syntactic features, and structural features. Features previously proposed by Moens *et al.* [126] such as n-gram, Part of Speech (POS), Sentence Length, Average Word Length and Punctuation were also used. Along with these, Universal Dependencies, Syntactic Parser features and some features specific to the legal corpus such as 'Other Law', 'Decision and Judgment' were also evaluated. The features used are described below.

N-gram is one of the most efficient, straightforward and useful features. It mainly works to identify an adjoining sequence of 1 to n tokens of any given sentence. Further details concerning this feature are discussed in Section 2.2.2.

Part of Speech (POS) tagging is the process of annotating the words with the part of speech (grammatical notation) based on its definition and context. POS tagging is a prevalent methodology to generate the grammatical knowledge of each word. On the other hand, it is quite a difficult task as words may belong to more than one part of speech depending upon the context of the sentence. Thus, the system needs to tag the word according to the meaning, structure, and context of the sentences. The Stanford NLP toolkit [113] was used to annotate the words of the case-law. These features' details are discussed in Section 2.2.2.

Sentence Length is another important feature that depends on the amount of information present in a sentence, a feature used in most textual analysis scientific research [121, 122, 139, 129, 194, 204]. In general, descriptive information (non-argumentative sentences) are longer in comparison to sentences devoted to making claims or providing reasons (argumentative sentences). The sentence length is determined by calculating the total number of characters present in the sentence. The unit of this feature is an integer.

Average Word Length is a relevant feature used in several publications in argument mining research [121, 126, 130, 139, 156]. To find the average word length, the total number of characters is divided by the number of words present in the sentence. The unit of this feature is a real number.

Punctuation: Presenting an argument through text is different from arguing verbally. The verbal argument may include sentiments, aggression, and politeness. Punctuation helps to translate these feelings into a written format. Question marks, exclamation marks, and commas have a significant impact on the written argument. The units of this feature are the punctuation marks themselves.

Universal Dependencies [133] are descriptions of logical relationships in a sentence. It shows textual relations between the words in a sentence. A set of dependencies from the English Language

were used. The 50 grammatical relations listed by Marneffe and Manning [115] are used to identify the structure of the sentences present in the case-law.

Syntactic Parser: The structure of the Parse Tree is also relevant and effective. Details of such features were described in Section 2.2.2. The Stanford NLP toolkit [113] was used to calculate the parser length, depth, and size. The unit of these features is an integer.

Other Law: This is a binary feature based on location. Case-law documents are partitioned into several sections, as stated in Section 4.2. In the experiment, two categories were employed. The sentence that lies in the 'As the law' section of case-law of the Judgment Category and 'The Law' section of case-law of Decision Category of the corpora are label as 'Law'. The 'Other' tag is applied to the remainder of the sections (Introduction, The Facts, Complaints, Proceedings before the Commission) of the corpora. An advantage of such location-based feature is that it helps to identify the content of the section, where argumentative sentences are located.

Decision Judgment: This is another binary feature (Decision and Judgment) developed on the basis of the kind of the case-law. There are two types of case-law in the data set (as mentioned in Section 4.2). The main difference between these corpora is size. The Decision has a word average of 3500 while Judgment has, on average, 10000 words as detailed in Chapter 4. Sentences were categorized as capital letter 'D' for Decision categories and capital letter 'J' for Judgment categories and annotated accordingly.

After defining features, the analysis was divided into three categories: a Collective-based approach, a Categorical-based approach and a Merge-based approach. In the Collective-based approach, all available features were used and performance was measured by selecting the top features; in the Categorical-based approach, the analysis was made measuring performance with a specific type of feature, and in the Merge-based approach, the best features were chosen from the previous two approaches: Collective and Categorical. The performance of the classifier based upon these features is described in Section 6.1.

5.1.3 Tree Kernel Experiments

A Tree Kernel generates large number of constituency parse trees based on the bracket notation (Lisp S-Structure). It is worthwhile to compare the tree structure generated by argumentative sentences vs. non-argumentative sentence. These generated features are used to measure the performance of SVM. The main advantage of using a Tree Kernel is that it can use as input the Syntactic Parse Tree of each sentence to create a model. Three kinds of features are proposed: Syntactic Parse

Tree (generated through Core-NLP tool [113]), the TF-IDF features and a combination of both.

5.2 Argument Builder

After identifying the argumentative sentences, it is necessary to organize these sentences to form an argument (i.e. a set of related argumentative sentences). Hence, the second stage of the work was to identify the boundaries of the arguments (from in Chapter 3). Identifying boundaries of an argument is quite a difficult task for those corpora where argument's sentences are dispersed around the document; which is the case with the ECHR Corpus. Mochales and Moens [121] found boundaries by using 'semantic distance', but here a new clustering technique is proposed for joining argumentative sentences into a cluster representing a potential argument. It is important to recall that as previously stated, arguments are composed of argumentative sentences, which can be premises or conclusions.

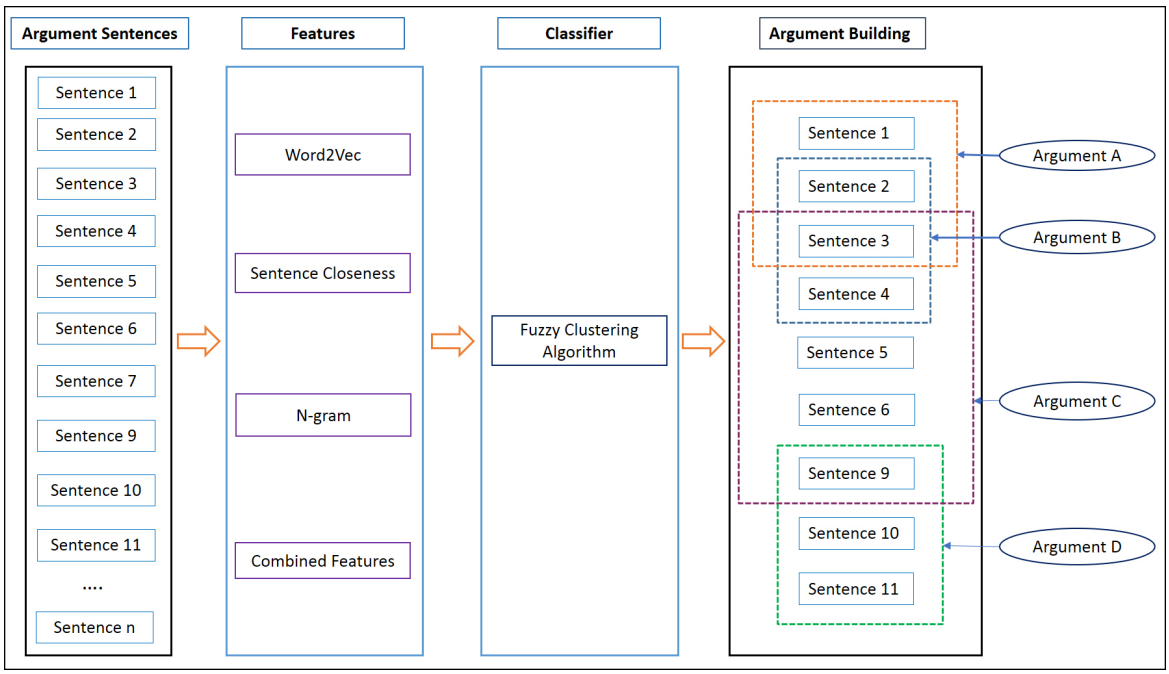

Figure 5.4: Overview of the Argument Builder Module

The overview of the proposed approach is shown in Figure 5.4. The task is challenging because the components of one argument (premise or conclusion) can also be involved in another argument. As depicted in the figure, there all altogether four arguments: for instance, sentence #2 belongs to argument A and also to argument B. To cluster such sentences, hard clustering algorithms are not an appropriate choice; instead, soft clustering i.e. a Fuzzy Clustering Algorithm (henceforth referred to as FCA) is needed, as FCAs allow a sentence to be in multiple clusters (as a requirement

of the proposed system). The membership values are the key assets of the FCA, allowing us to associate each sentence to more than one cluster/argument. The performance of the algorithm depends heavily on the kind of features being used. For this experiment, four kinds of features were tested: n-gram, Word2vec, Sentence Closeness and Combined Features. The goal is to identify the best features and techniques to cluster components to form an argument.

After extracting the features associated with each sentence, the FCA is used to get a membership value for every sentence. To obtain the composition of each cluster, the 'Distribution of Sentence to the Cluster Algorithm' DSCA was used as discussed in Section 5.2.3. To evaluate the system the 'Appropriate Cluster Identification Algorithm' ACIA was developed which helps to map each cluster to the closest argument on the gold-standard. Details of the ACIA are discussed in Section 5.2.4.

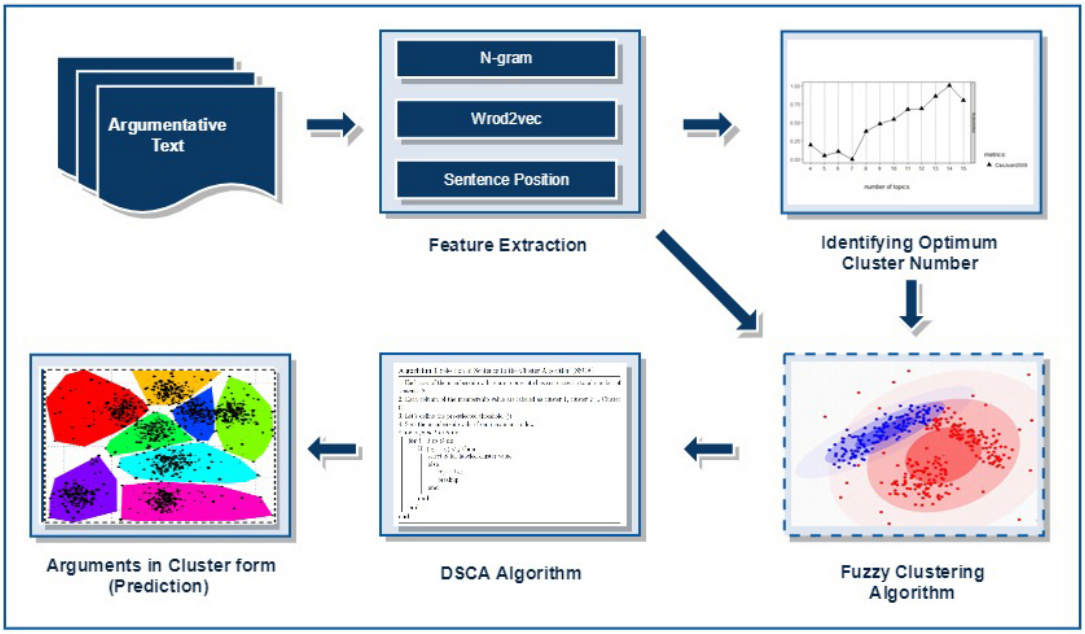

Figure 5.5: Proposed Architecture of the Argument Builder Module

Figure 5.5 presents the system architecture. As can be seen several phases need to be accomplished: feature extraction; identification of the optimum number of clusters, application of algorithms and system evaluation. Each of these phases will be discussed in the following pages.

Feature Extraction

Features can be numerical values as well as non-numerical values that represent each sentence, and are suited for a machine learning algorithm to handle. It is essential to select the most appropriate and precise features to train the machine learning algorithm so that the model can successfully be

applied to new data. Therefore, features are needed that can correlate the similarities between the sentences and also address the sequential order of the sentences (i.e. the majority of the components of the arguments are in order). To address this requirement, the following features were used: n-gram (explained in Section 2.2.2) and Word2vec and Sentence Closeness (discussed below). A further set of features can also be obtained by combining these three types of features; and this was termed *Combined Feature*. Each of them is discussed here.

Word2vec: To cluster sentences that are dispersed around the document is very difficult. It is necessary to understand the context of the sentence in the document. To address such issues one possibility is Word2vec [118] (described in Section 2.4.3). Word2vec is one of the prominent and emerging methods that help to represent the semantic and contextual information present in a text.

We used a Skip-gram model with a window size of 5. A Wikipedia dump from 05-02-2016 was used as input to the Word2vec implementation of Gensim [154], where 100 dimension-vectors were generated for each word. From the training set, each word of the sentence is looked up and its corresponding vector found among these generated word vectors. Then the average of all vectors of the words present in the sentence is taken and considered to be the 'sentence vector'. Code for producing a sentence representation using Word2vec is presented in appendix D.

Sentence Closeness: Sentence distance is the reciprocal of the inter-sentence distance (i.e. the distance between sentences) counted in units of whole sentence. To capture the sequential nature of sentences, distance is a useful feature that helps to determine which sentences belong to which argument. The highest scoring sentence is considered to be the origin sentence (with a score of 1) from which distances are measured. With the exception of the origin sentence, 'closeness' scores should decrease monotonically as they move away from the origin. Furthermore, meaning and concepts flow from one sentence to another, implying that sentences whose 'closeness' is high are good candidates for being clustered together. Equation 5.1 was used to calculate the 'closeness' for each sentence.

$$Closeness_n = \frac{1}{n} \tag{5.1}$$

where n is the distance between sentences, measured in integer units of sentences.

Combined Features: The previously presented features (n-gram + Sentence Closeness + Word2vec) were combined into a new set of features in an attempt to improve the performance of the clustering algorithms.

5.2.1 Identification of the optimum number of clusters

To cluster sentences, the number of clusters needs to be determined. Until recently, there was no approach to defining the exact number of clusters. Several techniques that claim to be able to define the optimum number of clusters in the Fuzzy c-means (FCM) clustering algorithm proposed by Bezdek *et al.* [27] have been proposed by Xie and Beni [205] and Latent Dirichlet Allocation (LDA) model [30]. Experiments were reviewed with cluster numbers ranging from three to 15 for each case-law file (i.e. case-law files that have less than ten arguments).

In the Xie and Beni approach [205], an FCM was used with a fuzziness value of m set to 1.3, with the features Word2vec, n-gram and Sentence Closeness. The cluster that scored the minimum was then selected because according to Xie and Beni, an appropriate cluster number can be predicted on the basis of the minimum value of the index.

The LDA technique estimates the number of topics existing within a text, which means estimating the probability of groupings within the text, and also to estimate the number of topics. Inspired by the concept, it was decided to look for such groups within our corpora, and to use the estimated number of topics as a proxy for the number of clusters. The LDA technique does not accept the Word2Vec, TF-IDF, and Sentence Closeness values as features. Therefore, word frequency of the document was used. We selected the 'CaoJuan2009' method as a metric which is the best LDA model based on density [131]. 'CaoJuan2009' was tested and agreed the appropriate number of topics (number of clusters) can be predicted by the minimum index value [38].

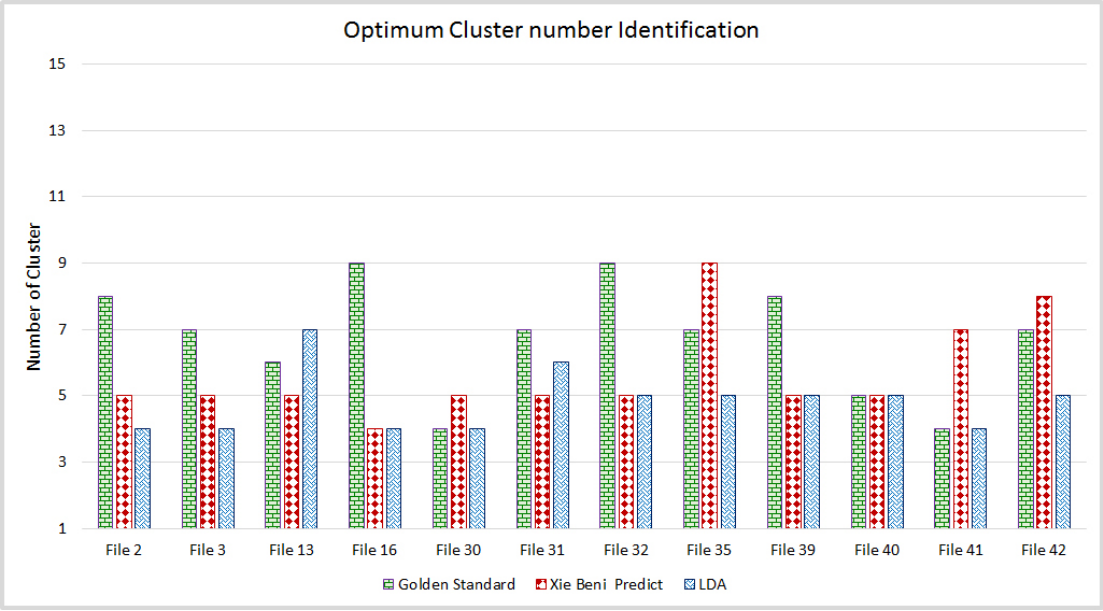

Figure 5.6: Argument counts of gold-standard vs. System Prediction (proposed by Xie and Beni and Cao *et al.*)

Figure 5.6, illustrates the results for each experiment. The first set is the gold-standard, the second is Xie and Beni's proposal, and the third is from LDA (Cao *et al.* [38]). In the case of Xie and Beni, it can be observed that case-law files 02, 31, 32, 39, 42 find the closest number of clusters (although still slightly different) to the gold-standard, whereas for other case-law files the differences are greater.

In cases of LDA (Cao *et al.*'s) prediction: Case-law files 40 and 41 finds the correct data required for identification, whereas other case-law files present a slight difference, but not as big as that observed by Xie and Beni. The exact accuracy score achieved 25% for LDA and 8% for Xie and Beni respectively. Furthermore, it is preferable to find a cluster value from equation 5.2 that differs by at most 2. As a result, closeness accuracy from equation shows an increase in value of up to 58% for LDA and 42% for Xie and Beni, respectively.

$$|C_s - C_g| \leq 2 \tag{5.2}$$

where C_s is the cluster number given by the prediction system, and C_g is the cluster number given by the gold-standard.

From the analysis, it's possible to conclude that LDA (Cao *et al.* [38]) technique achieves the more accurate result and one that is much closer to the gold-standard. Improvements in the results are expected to be achieved in the future finding suitable features.

5.2.2 Clustering Algorithm

After extracting the features, FCM was used to generate membership values for each cluster. The number of clusters is provided according to the number of arguments identified in the previous section. Fuzziness values $(m) \in [1.1, 1.3, 2.0]$ were tested.

5.2.3 Distribution of Sentence to the Cluster Algorithm

After receiving the membership value for each sentence from the FCM, the DSCA was applied (transforming the membership value into a clustering). DSCA is presented as Algorithm #2.

As defined, membership values are represented by a matrix where each row represents a sentence and each column is labeled as cluster numbers beginning from 1 to C. To convert a soft clustering to a hard clustering (i.e. distributing the sentence to the respective cluster), a *threshold value* t needs to be set to help create boundaries between the clusters. The value is defined, only if the difference

Algorithm 2: Distribution of Sentence to the Cluster Algorithm (DSCA)

1. Denote the matrix of the sentences x cluster by $(a_{ij}) \in [0,1]$, $i = 1,2,3 \cdots S$ and $j = 1,2,3 \cdots C$ such that i stands for sentence and j stands for cluster.
2. Pre-selected threshold (t) is defined
3. **for** *each* i *do* **do**
   - $i_{max} = max(a_{ij}) \quad \forall i$
   - **for** *each* j *do* **do**
     - **if** $(i_{max} - a_{ij}) < t$ **then**
       - select sentence i for cluster j
       - **else**
         - reject i;
       - **end**
     - **end**
   - **end**

end

between maximum membership value to the corresponding i^{th} position membership value is less than the *threshold value* of that cluster (C_i). Otherwise, that sentence is rejected. The algorithm ends after conducting an iterative process going through all positions in the matrix. The concept of *threshold value* is discussed by Al-Zoubi *et al.* [6] as well as Jain *et al.* [88]. Al-Zoubi *et al.* [6] used a *threshold value* to eliminate data points that were smaller than the *threshold value*. The authors also claim that the definition of the appropriate *threshold value* is based on experimentation.

After applying the DSCA, a clustering of arguments is obtained, which is the main goal of this work. However, the performance of this module has to be evaluated through the application of the ACIA, a process which is explained below.

5.2.4 Appropriate Cluster Identification Algorithm

Let A, B be the system's clustering set and the gold-standard clustering, respectively, having a cardinality of n: $A = \{a_1, \cdots, a_n\}$ and $B = \{b_1, \cdots, b_n\}$. We define the matrix $F = \{f_{ij}\}$ where $f_{ij} = a_i b_j$ with $a_i \in A$ and $b_j \in B$. Here, $F = \{f_{ij}\}$ is the f-measure value calculated taking cluster i in A and cluster j in B.
We denote by $(F)_{ij}$ the matrix formed from the F by removing the j^{th} column and i^{th} row

State 1 : Initialize

$$F^o = (f_{ij})_{n \times n}$$

$$R^o(-1,-1) = \emptyset$$

i.e. Nodes are connected with the cost value C=0 to form a tree structure.

State 2 : From $k = 0$ to n iterate (at each k step, we have $F^{(k)}(i,j)$ and $R^{(k)}(i,j)$)

Find all maximum elements of $F^{(k)}(i,j)$

Let

$$M_k = \left\{(i,j)|f_{ij}^{(k)} is\ the\ maximum\ element\ of\ F^{(k)}(i,j)\right\}$$

i.e. Maximum f-measure value is selected and placed in tree structure;

State 3: For each element $(i,j) \in M_k$, update route

$$R^{(k+1)}(i,j) = R^{(k)}(i,j) \cup \{(i,j)\}$$

and matrix

$$F^{(k+1)}(i,j) = \left(F^{(k)}\right)_{ij}$$

Do it for all elements (i,j) of M_k

Stop $k = n$.

i.e. Procedure repeat again for other remaining values;

State 4 : {For each route, calculate total cost}

$$TC_{R^{(k)}(i,j)} = \sum_{(i,j)\in R^{(k)}(i,j)} f_{ij}$$

i.e. The total cost of each route is calculated.

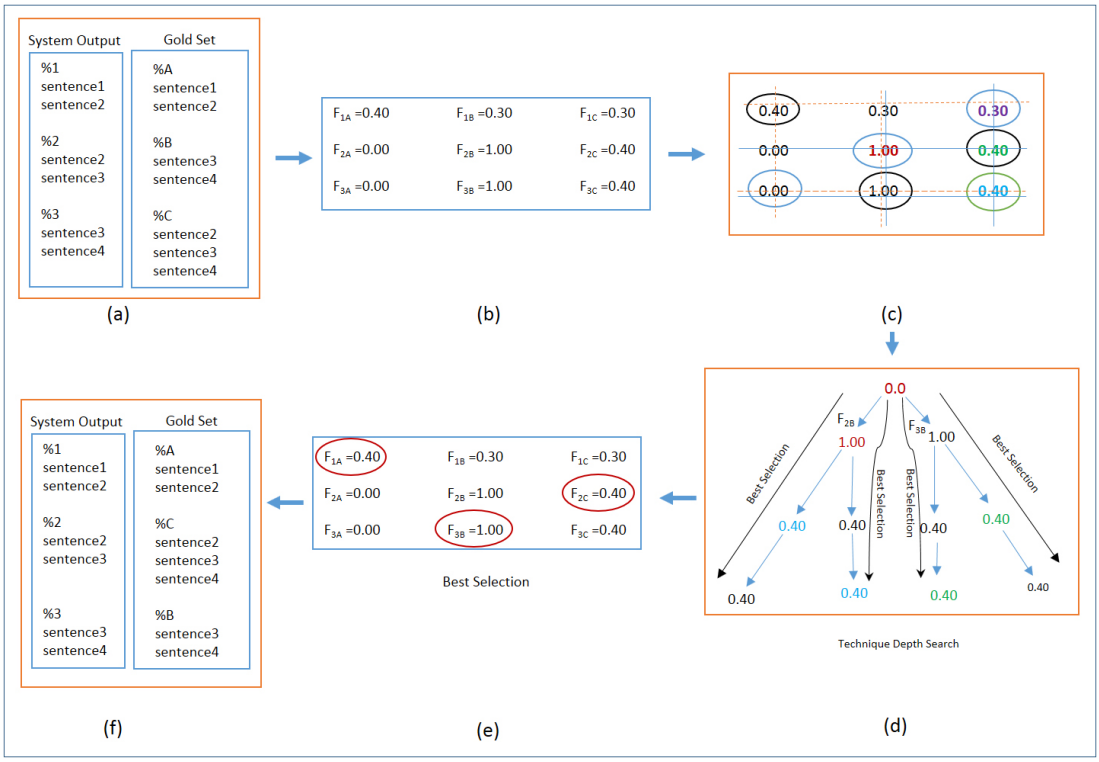

Figure 5.7: Demonstration of ACIA

State 5: Select one of the maximum value

$$TC_{R_o(i,j)},$$

and its route

$$R_o(i,j) = \{(i_1, j_1), \cdots, (i_n, j_n)\}$$

i.e. The route that scores maximum value is selected.

After identifying the appropriate cluster (argument) with respect to the gold-standard; an f-measure is calculated between the i^{th} cluster of the system as recommended by the ACIA and the j^{th} cluster of the gold-standard. After that, the average f-measure value is calculated.

A short description of the mathematical notation of the above algorithm is explained through the example presented in the Figure 5.7.

- Consider a 3×3 matrix of f-measures (i.e. 3 clusters in gold-standard datasets, and 3 clusters in system prediction data sets) as shown in Figure 5.7 (a);

- An f-measure value is calculated between the i^{th} cluster of the system and the j^{th} cluster of the gold-standard data sets and the value is presented in matrix forms (as shown in Figure 5.7(b))
- The maximum value from matrix is selected. The procedure is repeated for other remaining values as shown in Figure 5.7 (c);
- Nodes are connected with a cost value $C = 0$ forming a tree structure. The total cost of each route is calculated. The route that scores maximum value is selected as shown in Figure 5.7 (d);
- On the basis of selected route, a matrix position/entry along with the obtained f-measure is selected as shown in Figure 5.7 (e);
- In Figure 5.7 (f) the closest argument is selected as defined in (e).

The f-measure is calculated for the closest cluster/argument as selected; After that, the average f-measure value is calculated. Code for producing a ACIA algorithm is presented in appendix E

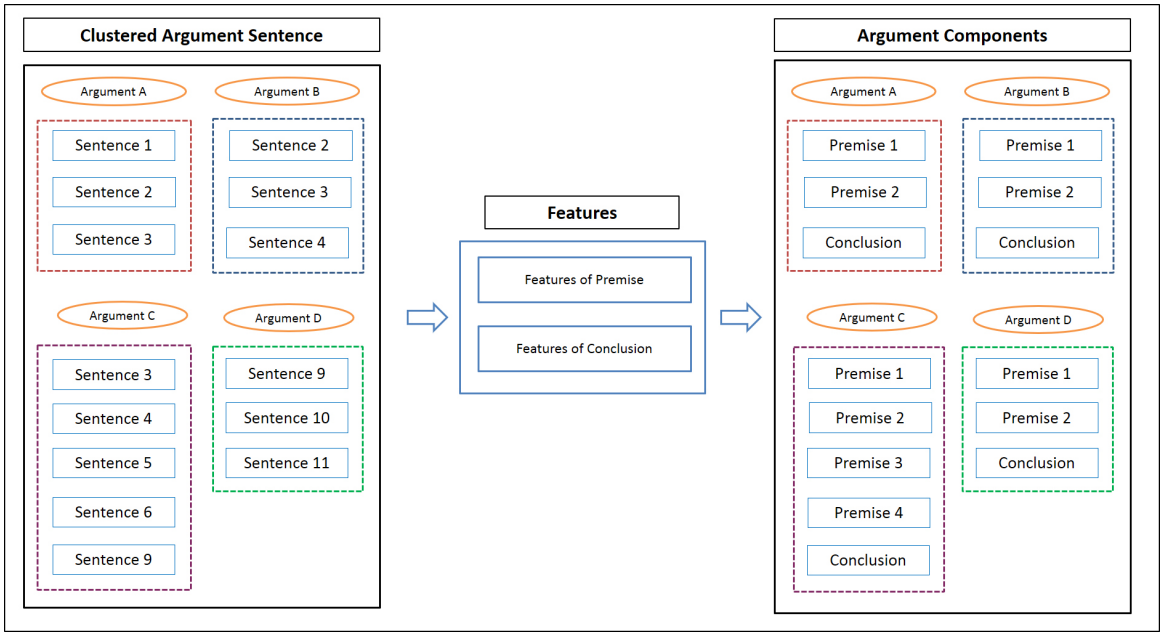

Figure 5.8: Overview of the Argument Structurer

5.3 Argument Structurer

The Argument Structurer (AS) module deals with the identification of the internal structure of the arguments i.e. identification of components of arguments as either premise or conclusion. An

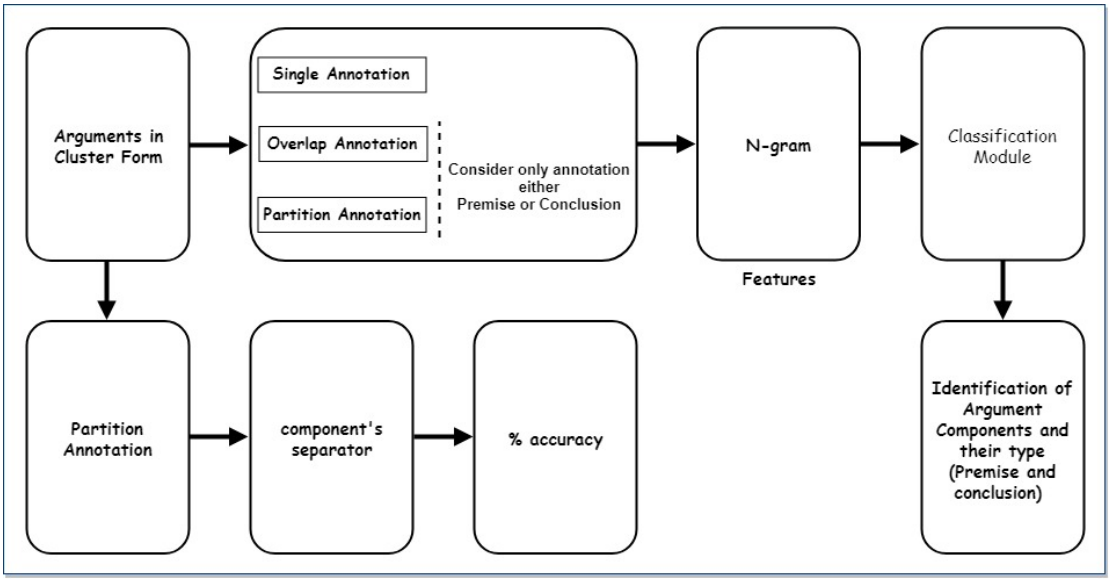

Figure 5.9: Proposed Architecture of the Argument Structurer

overview of the argument structurer model is presented in Figure 5.8. The task is to detect premises and a conclusion for the arguments that are found.

The proposed architecture for the system is shown in Figure 5.9. Several phases need to be accomplished in this stage of the experiment: features are extracted, the classifier is built and its performance is evaluated.

In Chapter 4 the structural complexity of the argumentative sentences is explained. In the experiment, the annotation of the sentences in the corpora is divided into three categories: Single Annotation, Overlap Annotation, and Sentence Partition. The goal is to find distinctive features to classify the argumentative sentence either as premise or conclusion. To classify a Single Annotation sentence, the same technique described in Section 5.1 is used. To classify a Overlap Annotation sentence is more difficult as explained in Section 4.6.2. Sentence Partition classified is based on the structural components of the argumentative (as detailed in 4.6.3). To address such complicated argument structure, the experimental procedure was divided into two phases: a classification-based approach and a rule-based approach.

In the classification based approach, two sets of experiments were tested: Premise Basis and Conclusion Basis. Similarly, in the rule-based approach, the accuracy of an indicator that separates the components of the argument in the Sentence Partition category is measured. Experiment results are explained in Section 6.3.

5.4 Summary

This chapter describes the proposed architecture for a system that identifies arguments from text in the legal domain. Three modules were designed and run sequentially. Investigations were carried out on how the argument structure of legal text can be identified. First, an approach to identify the best machine learning algorithm with appropriate features to determine the argumentative sentences from the legal documents was proposed. Experiments were divided into three parts: Basic, Multi-feature and Tree Kernel.

Second, a novel approach using a clustering technique is applied to the argument mining field. The primary purpose of this technique is to group the argumentative sentence to form an argument. The method is quite challenging due to the characteristics of legal argument and drove us to use a soft clustering algorithm with a new evaluation setup. The evaluation provides an accuracy score comparable to the gold-standard dataset.

Third, an approach aiming to deal with argument structure is investigated. In this module, a machine learning approach was applied to classify the argumentative sentence as premises or conclusions. A rule-based approach was also applied to identify the token that separates the components of arguments (i.e. premise and conclusion) within the argument.

6

Experiments and Results

This chapter describes the experimental work and evaluates the system proposed in Chapter 5. The primary goal to ensure the most efficient process of extracting arguments from legal texts. The task has been addressed by dividing it into several modules as explained in the previous chapter. The experimental work implemented in each module is presented, with the evaluation of their individual qualities and drawbacks to determine if the proposed approach is adequate. The results obtained in each module are analyzed and discussed.

6.1 Argument Element Identifier

A system is proposed to identify automatically argumentative sentences from unstructured text. Several phases need to be accomplished, such as refinement of the corpus (as discussed in section

4.3), extraction of features (as described in section 5.1), the selection of a classifier and evaluation of its performance. As mentioned before the experiments are divided into three categories: Basic, Multi-feature and Tree-Kernel. The results obtained in each of the categories are compared.

6.1.1 Basic Experiment

The primary goal of the Basic experiment is to identify an optimum machine learning algorithm with appropriate features to distinguish argumentative from non-argumentative sentences in the legal domain. To achieve this goal, the top n informative features where $n \in \{100, 200, 500, 1000, 2000, 5000, 11374\}$ were selected using the gain-ratio measure [148], a polynomial kernel with various values of complexity parameter ($C \in \{0.001, 0.01, 0.1, 1, 10, \ and \ 100\}$). The results, for precision, recall and f_1 are given in Tables 6.1, 6.2 and 6.3, respectively.

Features \C	0.001	0.01	0.1	1	10	100
100	0.000	0.981	0.981	0.991	0.992	0.992
500	1.000	0.683	0.726	0.696	0.685	0.685
1000	0.712	0.732	0.715	0.639	0.583	0.577
2000	0.727	0.703	0.650	0.615	0.610	0.610
5000	0.723	0.685	0.653	0.646	0.646	0.646
11374	0.717	0.672	0.654	0.654	0.654	0.654

Table 6.1: Precision for SVM (Basic Experiment)

Features \C	0.001	0.01	0.1	1	10	100
100	0.000	0.025	0.047	0.101	0.111	0.111
500	0.002	0.175	0.304	0.389	0.413	0.413
1000	0.139	0.341	0.477	0.520	0.520	0.519
2000	0.237	0.478	0.545	0.550	0.551	0.551
5000	0.250	0.489	0.528	0.527	0.527	0.527
11374	0.310	0.501	0.513	0.513	0.513	0.513

Table 6.2: Recall for SVM (Basic Experiment)

Table 6.1 and Table 6.2 display the results obtained for precision and recall. As the number of the features and the complexity value C increases, the precision decreases, in contrast recall value increases. Table 6.3 reports how with fewer features and low values of C, then the value of f_1 is also low, but as the value of C increases, the f_1 value increases but only up to $C = 0.1$; after that,

as the value of C increases, the f_1 value decreases, albeit slightly. In consequence, the highest f_1 value of 0.595 is achieved with $C = 0.1$ with 2000 features. Applying statistical tests, it was found that there is no significant difference between the results obtained with $C = \{0.1, 1\}$. Therefore it was considered taking $C = 1$ which scores f_1 0.581 with 2000 features.

Features \C	**0.001**	**0.01**	**0.1**	**1**	**10**	**100**
100	0	0.048	0.090	0.183	0.200	0.200
500	0.004	0.279	0.428	0.499	0.515	0.515
1000	0.233	0.465	0.572	0.573	0.550	0.546
2000	0.357	0.569	0.593	0.581	0.579	0.579
5000	0.372	0.571	0.584	0.58	0.580	0.580
11374	0.433	0.574	0.575	0.575	0.575	0.575

Table 6.3: F_1 of SVM (Basic Experiment)

Similarly, experiments were performed using the Random Forest (RF) algorithm with a number of trees, 'nt', where $nt \in \{7, 11, 17, 50, 100\}$. Tables 6.4, 6.5, 6.6 present the precision, recall and f_1 values. For precision, there was no difference between the values at the lower end of top-features even though the number of trees varies. As the number of features increases, precision value declines. This is true only up to 1000 features, beyond this point, the value starts to rise as the feature number increases. Notably, beyond 2000 features, as the number of trees increases, the precision value also rises. Recall and f_1 values both follow a similar trend, as shown in Tables 6.5 and 6.6. For f_1, one can conclude that with fewer features and numbers of trees, the algorithm scores poorly. However, for the same number of features, the value increases as the number of trees increases. At a certain level, after achieving a peak score, values start to decline. To be more specific, a peak of 0.524 is achieved with 500 features and seven trees.

Features \No. of Tree	7	11	17	50	100
100	0.992	0.992	0.992	0.992	0.992
500	0.687	0.684	0.687	0.683	0.681
1000	0.609	0.616	0.632	0.650	0.661
2000	0.646	0.690	0.702	0.758	0.761
5000	0.632	0.684	0.696	0.750	0.756
11374	0.629	0.665	0.682	0.745	0.759

Table 6.4: Precision for RF Algorithm (Basic Experiment)

Features \No. of Tree	7	11	17	50	100
100	0.11	0.111	0.112	0.113	0.113
500	0.424	0.415	0.422	0.415	0.412
1000	0.435	0.419	0.422	0.419	0.429
2000	0.353	0.336	0.310	0.276	0.272
5000	0.323	0.300	0.280	0.241	0.234
11374	0.314	0.285	0.259	0.228	0.214

Table 6.5: Recall for RF Algorithm (Basic Experiment)

Features \No. of Tree	7	11	17	50	100
100	0.198	0.2	0.201	0.203	0.203
500	0.524	0.517	0.523	0.517	0.513
1000	0.507	0.499	0.506	0.51	0.52
2000	0.457	0.452	0.43	0.405	0.401
5000	0.427	0.417	0.399	0.365	0.358
11374	0.419	0.399	0.375	0.349	0.334

Table 6.6: F_1 for RF Algorithm (Basic Experiment)

Analysis

From the above results, it can be concluded that, for this type of corpus, the SVM algorithm with a polynomial kernel and a complexity C parameter of 1 performs better than the RF algorithm. Therefore, in the following experiments, a SVM polynomial kernel with $C = 1$ is used.

6.1.2 Multi-feature Experiment

This section describes the experiments based on different categories of features. This experiment was divided into three approaches as shown in the Figure 6.1: Collective-based (merging all the features), Category-based (dividing the feature by kind: Word n-gram, POS n-gram, and Doc- Info) and Merge-based (merging the best features of each category). For the reasons explained in the previous paragraph, the performance is measured using a SVM polynomial kernel with C = 1.

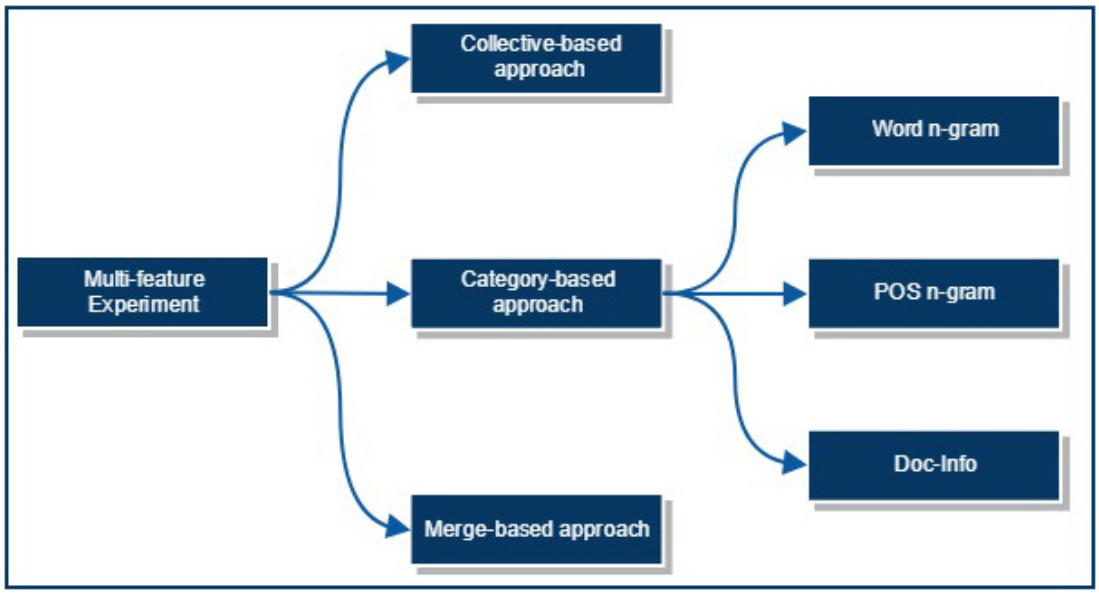

Figure 6.1: Classified structure of different kinds of Multi-feature experiment

Collective-based approach

In this approach all features were merged together. After that, the top-n informative features were selected using gain-ratio [148]. Models were built with $n \in \{2000, 5000, 10000, 20000,$ 50000, $100000, 200000$ and 238853 $(allfeatures)\}$. Table 6.7 shows precision, recall, and f_1 measures. The results indicate that precision value decreases gradually as the number of features increases; for recall, the result is the opposite. The f_1 value starts to decline from 2000 to 5000 features and

increases again from 5000 features onwards. The best f_1 value is obtained with 20,000 features with a value of 0.705. However, when statistical analysis was performed, there was no significant difference between the result of 10,000 features vs. 20,000 features, hence, the f_1 value 0.686 (obtained using 10,000 features) is considered as the best value achievable by the 'Collective Based Approach'.

Features	Precision	Recall	f_1
2000	0.991	0.438	0.607
5000	0.993	0.416	0.586
10000	0.718	0.656	0.686
20000	0.719	0.691	0.705
50000	0.708	0.684	0.696
100000	0.698	0.686	0.692
200000	0.698	0.7	0.699
238853	0.697	0.701	0.699

Table 6.7: Precision, Recall and f_1 Results (Collective-based approach)

Category-based approach

In this approach, each kind of feature was tested separately. Three experiments were conducted: Word n-gram, POS n-gram, and Doc-Info.

Word n-gram: Unigram, bigram and trigrams of words were used as features. Initially, the 2000 most informative features were selected to measure the performance of the classifier by observing precision, recall, and f_1. Next, the same process was repeated for other numbers of features 5000, 10000, 50000, 100000, 200000 as well as 229746 (all features). The results are presented in Table 6.8. It was found that precision gradually decreases as the number of features increases. Recall and f_1 are more consistent, displaying only small fluctuations. In the end, the feature number that gives the highest f_1 value (a value of 0.599 with 2000 features) is selected. Also, when statistical analysis was performed, there was no significant difference between the result using 2,000 features vs. 5,000 features, hence, the f_1 value 0.599 (obtained using 2,000 features) is considered the best that can be achieved by the 'Word n-gram'.

Feature	Precision	Recall	f_1
2000	0.99	0.43	0.599
5000	0.993	0.401	0.572
10000	0.689	0.516	0.59
20000	0.7	0.467	0.56
50000	0.737	0.359	0.483
100000	0.701	0.388	0.5
200000	0.676	0.404	0.506
229746	0.66	0.417	0.511

Table 6.8: Precision, Recall and f_1 Results (Category-based approach using of Word n-gram)

POS n-gram: In this experiment, unigram, bigram, and trigram of POS tags were used as features. The 100 most informative features were selected using gain-ratio measure [148], and performance of the classifier was measured. The same process was repeated for other numbers of features: 100, 200, 500, 1000, 2000, 5000, and 9052 (all features). Table 6.9 presents the results for the precision, recall, and f_1 of this classifier. F_1 increases gradually as the number of features increases. However, this condition is only true up to 2000 features. After that f_1 slightly decreases as the features increases. Thus, the highest f_1 value achieved was 0.516 with 2000 features. When a statistical analysis was performed, there was no significant difference between the result with 500 features and 2000 features, hence, the F_1 value 0.475 (obtained using 500 features) is considered as the best value achievable by the POS n-gram.

Feature	Precision	Recall	f_1
100	1	0.078	0.144
200	0.723	0.169	0.274
500	0.621	0.385	0.475
1000	0.522	0.504	0.513
2000	0.528	0.505	0.516
5000	0.535	0.494	0.514
9052	0.547	0.486	0.515

Table 6.9: Precision, Recall and f_1 Results (Category-based approach using POS n-gram)

Doc-Info: There are only 57 features in this approach, therefore, finding the top performing informative features was not necessary. Features such as location-based (Other and Law), corpus type

(Judgment and Decision); sentence word length, word count, punctuation, parser length, depth and size, as well as universal parser dependencies and adverb and verb count were used. The performance of SVM is shown in Table 6.10. Even with only a small number of features, the results obtained are among the best of any of the experiments. This result appears to be a consequence of using the Other Law feature.

Feature	Precision	Recall	f_1
57	0.606	0.606	0.613

Table 6.10: Precision, Recall and f_1 Results (Category-based approach using Doc-Info)

Merge-based approach

The features that produced the best results from Word n-gram, POS n-gram, and Doc-Info were merged and the SVM achieved a score shown in Table 6.11. It can be seen that an f_1 value of 0.661 was obtained with 2557 features.

Feature	Precision	Recall	f_1
2557	0.781	0.572	0.661

Table 6.11: Precision, Recall and f_1 of merging approach

Overall Assessment

Table 6.12, presents the best f_1 results. The Collective-based approach deals with a variety of features and creates models by selecting the top features obtaining a score of 0.686. The results obtained using the Category-based approach were less impressive. Out of the three (Word n-gram, POS n-gram, and Doc-Info), the most reliable outcome was produced by Doc-Info, which scored a value of 0.613. The Merge-based approach was designed to create a model from the best features overall. However, the results turned out to be less significant than initially expected despite being closer to the highest value achieved using the Collective-based approach.

A relevant result was obtained with Doc-Info especially considering that good accuracy was reached with the lowest number of features available. The most effective feature was 'Other and Law' since the most of the arguments came from the Law section. After achieved high performance also it was not able to surpass the Collective-based approach.

Type of Approach		**Highest Performance**	f_1
Collective-based		10,000 features	**0.686**
Category-based	Word n-gram	2,000 features	0.599
	POS n-gram	500 features	0.475
	Doc-Info	57 features	0.613
Merge-based		2557 features	0.661

Table 6.12: Highest performance and f_1 value according to type of approach used

6.1.3 Tree Kernel Experiment

As detailed is in Section 2.2.2, Syntactic Parser features in the form of Bracket notation tree (Lisp S-structure) were used for this experiment. Sentence Parse Trees are generated and used as input to a Tree Kernel [128]. Three kinds of features were employed: Syntactic Parse Tree (generated through the Core-NLP Tool [113]), the 2000 top most informative TF-IDF features; and a combination of both. The experiment was conducted in SVM-Light [89].

Table 6.13 presents the results for the Tree Kernel. The highest f_1 value (0.548) was obtained with the combination of features.

Experiment	Precision	Recall	f_1
Syntactic	0.717	0.428	0.504
TF-IDF	0.837	0.439	0.543
Combination	0.785	0.444	**0.548**

Table 6.13: Results for Tree Kernel

6.1.4 Evaluation of Argument Element Identifier

Approach	F_1
Basic	0.587
Multi-features	**0.686**
Tree Kernel	0.548

Table 6.14: Overall results for f_1 according to type of approach used

Table 6.14 summarizes the results obtained for the experiments conducted thus far. The Multi-feature experiment using 10,000 features is considered the most promising as it scored the highest

f_1 value (0.686). Although the f_1 value obtained through the Basic experiment was not particularly high, it nevertheless appears to have the potential to select the most appropriate algorithm. The results achieved by the Tree Kernel approach were the least promising.

This Argument Element Identifier module work is similar to the one developed by Mochales and Moens [121] for detecting argumentative information on the ECHR corpus in which the Naïve Bayes classifier was used and an 80% accuracy was obtained. Their results are not comparable to the ones presented here, because different evaluation measurements were used. Another relevant difference between the two studies is the annotations added to the corpus (explained in Section 4.4).

6.2 Argument Builder

After identifying the argumentative sentences from case-laws, it is necessary to group these sentences to form an argument. A set of possible features and algorithms were proposed to achieve this goal. Given that the same sentence of one argument can also be part of other arguments, an Fuzzy c-means (FCM) clustering algorithm proposed by Bezdek *et. al* [27] was chosen. However, membership value (ranging from 0 to 1) provided by FCM cannot be compared directly to gold-standard datasets. Thereafter, the DSCA was developed to transfer membership values to the cluster form (transforming a soft cluster into a hard cluster). Details of the approach and technique were explained in Section 5.2.

To perform an evaluation procedure, it is necessary to find the best mapping between the obtained clustering and the gold-standard clusters. The ACIA (described in Section 5.2.4) was proposed and developed to deal with this problem.

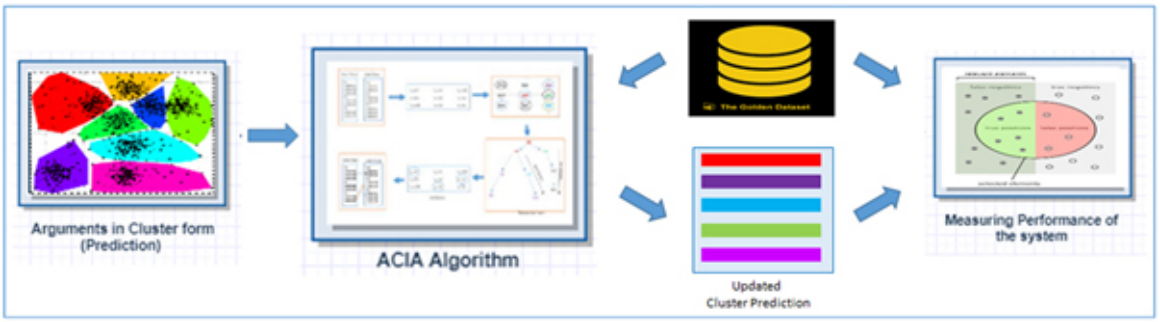

Figure 6.2: Evaluation of System Prediction

Figure 6.2 shows the procedure of the System Predication. Arguments in Cluster from Prediction are the arguments obtained after the DSCA. The ACIA is then applied to find the closest match between an argument and the gold-standard.

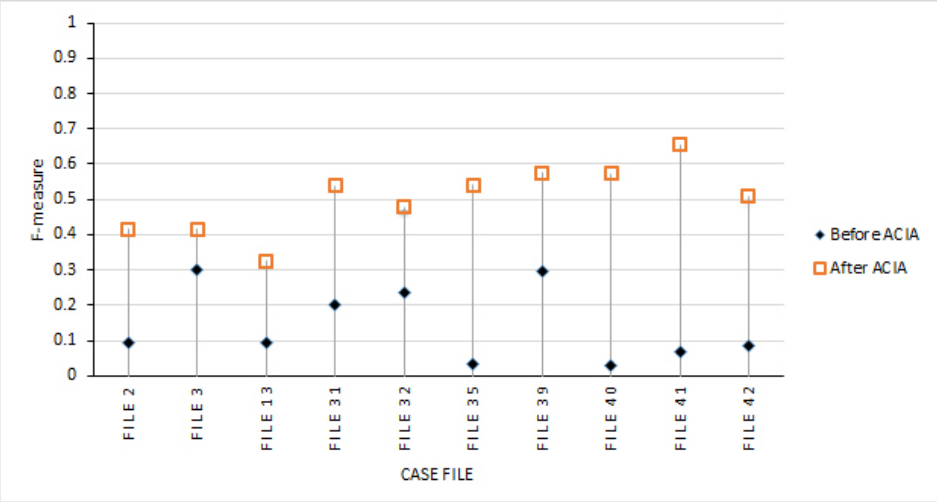

Figure 6.3: f_1 score before (sequence mapping) and after applying the ACIA

Figure 6.3 shows the comparative f_1 results obtained before (sequence mapping) and after applying the ACIA. It is important to note that f_1 before ACIA was obtained by performing a sequential mapping between the two sets of clusters. It can be observed that the value of 'After ACIA' (square symbol) is above 0.3 for all files, whereas in the case of 'Before ACIA' (diamond symbol) the maximum value is 0.3, which proves the relevance of this algorithm. However, there are some limitations in the algorithm. For instance, its time complexity is very high: $O(n^4)$. Since the algorithm's complexity is high. The experiment was performed using heavy computational resources. This is shown in appendix F.

6.2.1 Performance Measurement

The argument builder experiment was conducted with the features mentioned in section 5.2 with the fuzziness parameter, $m \in \{1.1, 1.3, 2.0\}$ and *threshold value* $t \in \{0.0001, 0.00001, 0.000001\}$ to convert from soft to hard clustering. The performance results (precision, recall and f_1) are presented in Table 6.15 showing the n-gram, Sentence Closeness, Word2vec and Combine features using a *threshold value* $t = 0.00001$ and a FCM fuzziness $(m) = 1.3$. The parameters that scored the highest f_1 value in most of the case-law files were selected. The sentence number for each case-law file was also included. The full set of results obtained from the other parameters are shown in Appendix C.

The highest f_1 value of each case-law file obtained from each feature is highlighted in bold and underline. Case-law files 03, 13, 16, 31, 32 and 42 obtained the highest value from Word2vec. Case-law file 02 scored the highest f_1 value from n-gram, and case-law files 30, 35 and 41, the

Case-law	Sen No.	n-gram			Sentence Closeness			Word2vec			Combine Feature		
		Pre	Rec	f_1	Pre	Rec	f_1	Pre	Rec	f_1	Pre	Rec	f_1
02	15	0.698	0.485	**0.573**	0.342	0.221	0.268	0.656	0.367	0.470	0.625	0.450	0.523
03	15	0.619	0.429	0.506	0.405	0.333	0.366	0.714	0.429	**0.536**	0.524	0.381	0.441
13	20	0.508	0.628	0.561	0.413	0.344	0.375	0.602	0.581	**0.591**	0.342	0.344	0.343
16	33	0.125	1.000	0.222	0.437	0.481	0.458	0.449	0.449	**0.449**	0.125	1.000	0.222
30	25	0.265	1.000	0.419	0.252	0.275	0.263	0.351	0.363	0.357	0.272	1.000	**0.428**
31	15	0.317	0.714	0.439	0.524	0.571	0.547	0.595	0.524	**0.557**	0.429	0.500	0.462
32	17	0.335	0.785	0.470	0.481	0.393	0.433	0.648	0.485	**0.555**	0.500	0.396	0.442
35	13	0.429	0.414	0.421	0.619	0.414	0.496	0.667	0.414	0.511	0.845	0.636	**0.726**
39	17	0.352	0.588	**0.440**	0.400	0.431	0.415	0.362	0.525	0.429	0.310	0.613	0.412
40	14	0.400	0.370	0.384	0.587	0.530	**0.557**	0.519	0.520	0.520	0.400	0.420	0.410
41	12	0.517	0.563	0.539	0.625	0.625	0.625	0.438	0.438	0.438	0.683	0.625	**0.653**
42	18	0.464	0.440	0.452	0.433	0.414	0.424	0.643	0.486	**0.553**	0.431	0.598	0.501

Table 6.15: Case-law, Number of Sentence, Precision, Recall and f_1 value of the System Prediction

highest f_1 from Combined Features. Likewise, case-law file 40 scored the highest f_1 value from Sentence Closeness. From this analyzes, it is concluded that Word2vec is the best approach.

In comparison to Word2Vec, n-gram (i.e. the combination of unigram, bigram, and trigram) did not perform well. The main reason for this effect is that n-gram is on the basis of 'bag of words' approach which is not effective in finding similarities between sentences. The results show that the performance of n-gram depends upon the number of sentences; if the number of the sentences in the case-law file is high, then the performance will be poor.

Sentence Closeness is another important feature that helps to understand the sequential context of the sentence. The following sentence has a huge impact on the argument as the meaning/context of a sentence usually flows sequentially. The results listed on this table show, that performance is satisfactory, but still lacking in comparison to Word2vec. The combined features also have an impact, as they are a combination of Word2vec, n-gram and Sentence Closeness. The combined features obtained the highest f_1 value in the case-law files in which Word2vec did not offer significant results except in case-law files 02, 39 and 40. Considering Word2vec and combined features, 66% of the case files obtained the highest f_1 using only these features. Furthermore, in the case of n-gram and combined features, recall is found further elevated by up to 1 and precision is very low for case files that have a large number of sentences. This shows that as the number of sentences increases, FCM provides an equal number of membership values from which sentences are distributed equally to the clusters. Therefore, as the number of sentences increase, the performance of the n-gram

feature decreases. However, n-gram is an appropriate feature for case files that have a lower number of sentences.

Case-law No.	No. of Sentence	n-gram	SP	Word2vec	Combined feature
02	15	**0.625**	0.412	0.563	0.600
03	15	0.563	0.412	**0.600**	0.533
13	20	0.500	0.400	**0.55**	0.400
16	33	0.125	**0.424**	**0.424**	0.125
30	25	0.263	0.320	**0.360**	0.275
31	15	0.313	**0.533**	**0.533**	0.400
32	17	0.326	0.474	**0.529**	0.474
35	13	0.467	0.571	0.615	**0.769**
39	17	0.346	**0.421**	0.368	0.250
40	14	0.467	0.533	**0.533**	0.467
41	12	0.500	**0.583**	0.417	**0.583**
42	18	0.389	0.389	**0.500**	0.414

Table 6.16: Case-law, number of sentences, cluster purity value on the basis of features

Table 6.16 presents the purity value of the cluster for each case-law file. Word2vec was found to play the leading role in case-law files 03, 13, 16, 30, 31, 32, 40 and 42. Sentence Closeness scored highest in four case-law files: 16, 31, 39 and 41. However case-law 16 and 31 tied with Word2vec. Overall the purity values are satisfactory, except in case-law file 16 and 33 with the combined features and n-gram. Case-law 16 which had 33 sentences had the lowest value (0.125) from Combined and n-gram features. Similarly case-law 30, that has 25 sentences, obtained 0.275. On the other hand, case-law 35 which had 13 sentences scored 0.726 (the highest value) from combined features. From this analysis, it is concluded that the number of sentences also affects the clustering quality negatively.

After analyzing both tables, it appears that Word2vec is the dominant feature for f_1 and cluster purity values. Figure 6.4 depicts three values: the bar graph represents sentences; the square box dot (in green) represents f_1 of Word2vec; and the diamond shape dot (in red) represents the cluster purity of Word2vec feature. The similarity of the graph patterns for f_1 and cluster purity is evident, except for 2 deviations.

Overall, the results obtained from the proposed framework are promising even if they cannot be compared with other researchers' results. Mochales and Moens [121] have a similar objective and

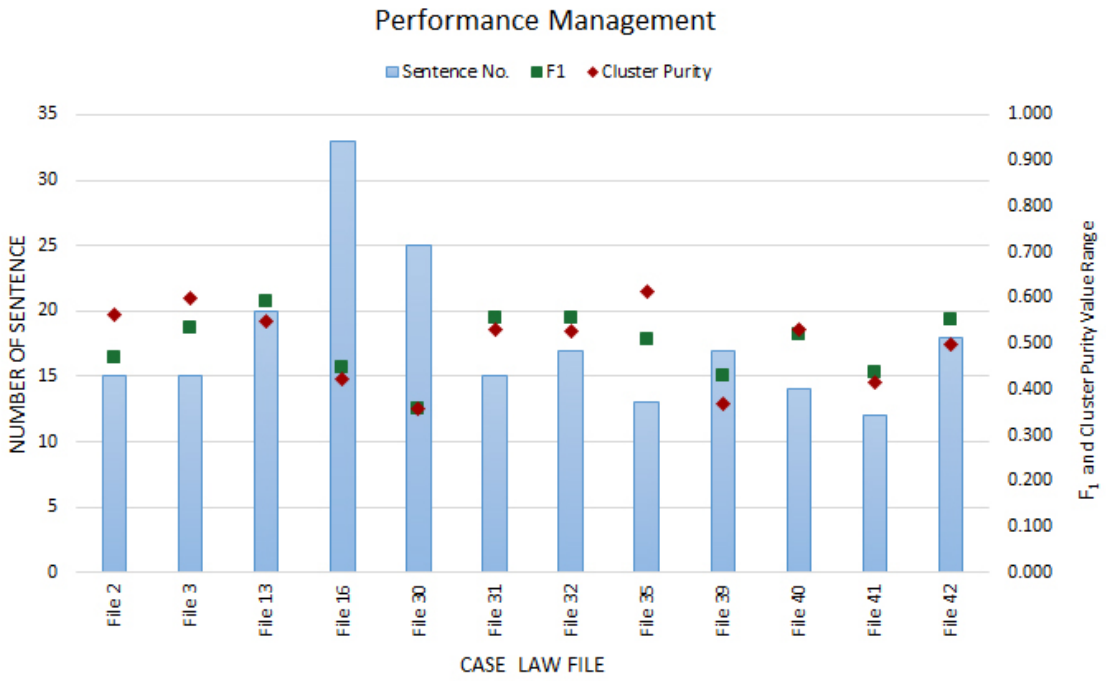

Figure 6.4: F_1 and Cluster Purity values of Word2vec for sentence number per case-law.

obtained 60% accuracy while focusing on argumentation structure detection. However, since different performance measurement are used, our results cannot be directly compared to theirs. Our approach, which can be used for any corpora and it is not limited to a specific domain, produced a set of results that are much closer to Mochales and Moens's than Goudas *et al.* [71], who obtained an F_1 of 0.424 while segmenting the argumentative sentence by Conditional Random Fields (CRF), or Lawrence *et al.* [99], whose precision and recall for identifying argument structure using automatically segmented propositions were 33.3% and 50.0%, respectively. Moreover, their manually segmented propositions reached precision rates of 33.3% and recall rates of 18.2%. Stab and Gurevych [173] also encountered problems dealing the support and attack relations. The main reason for this was that their approach was not able to identify the correct target of a relation especially in a paragraph with multiple claims or reasoning chains.

There are, nonetheless, several constraints in our approach. One major limitation is the computational complexity of the ACIA. For this reason, case-law files with less than 10 arguments were utilized, In the end only 12 files out of 42 available in the ECHR corpus were utlized. This is not a theoretical problem, but rather a technical one, which can be solved through the improvement of the algorithm implementation. The results obtained for the 12 files selected are still quite satisfactory.

After identifying the arguments using this clustering technique, the next step is to determine if the sentences are either premise or conclusion.

6.3 Argument Structurer

After grouping argumentative sentences to form an argument, it is necessary to know if these argumentative sentences are either premises or conclusions. Details of the proposed architecture were outlined in Section 5.3. To determine whether the argumentative sentences are premises or conclusions, we conducted an experiment in two phases. The first phase deals with a classification-based approach to identify the accuracy of the premise or conclusion of each sentence. In this second phase, a rule-based approach was applied to the 'Sentence Partition' category sentences to find the accuracy of the discourse indicators that separate the components of the arguments.

6.3.1 Classification-Based Approach

An approach to automatically identify premises and conclusions from argumentative sentences was proposed. A TF-IDF approach to 'bag of words' was used to classify the type of each sentence. This process has two parts: extraction of features and evaluation of the classifier. Next, two sets of experiments were performed: Premise Basis, and Conclusion Basis, which will be detailed in the following paragraphs.

No. of Features	Precision	Recall	F_1
500	0.907	0.997	0.950
1000	0.919	0.984	0.950
2000	0.925	0.965	0.944
5000	0.921	0.962	0.941
10000	0.918	0.951	0.934
20000	0.912	0.957	0.934
50000	0.905	0.962	0.933
75745	0.904	0.963	0.933

Table 6.17: Precision, Recall and f_1 of Premise Basis

Premise Basis: In this approach, a dataset was drafted to consider every sentence that is either classified as a premise itself or contains premise clauses as a premise, even if these sentences also display a conclusion clause. After that, sentences are transferred to the word n-gram and the top-n informative features are selected using the gain-ratio measure [148] with $n \in \{500, 1000, 2000, 5000, 10000, 20000, 50000$ and as well as the total amount of features available $75745\}$. An SVM classifier was used with linear kernel and complexity C parameter at value one. The performance of the SVM

classifier is presented in Table 6.17. As it can be seen, the top 500 features obtained the best f_1 results with a precision value of 0.907, a recall value of 0.997 and a f_1 value of 0.950. When statistical analysis was performed, there was no significant difference between the result of 500, 1000, 2000 and 5000 features, hence, the f_1 value 0.950 obtained using 500 features (the least number of features) is considered as the best value achieved on the 'Premise Basis'.

Conclusion Basis: In this approach, a dataset was set up to classify a sentence as a conclusion every time that it was already classified as such or, even though it has a premise clause, if it also contained a conclusion clause. After that, the sentences were transferred to the word n-gram and the top-n informative features were selected by using the gain-ratio measure [148] with $n \in \{500, 1000, 2000, 5000, 10000,\ 20000, 50000$ plus the total amount of features available $(75745)\}$. An SVM classifier was used with linear kernel and complexity C parameter at a value one. The performance of the SVM classifier, presented in Table 6.18, reveals that the top 2000 features obtained the best f_1 results to identify a conclusion, with a precision value of 0.896, a recall value of 0.635 and a f_1 value of 0.743. When statistical analysis was performed, there was no significant difference between the result obtained from the features, hence, the f_1 value 0.677 obtained using 500 features (less number of features compared to other) is considered as the best value achieved by the 'Conclusion Basis'.

No. of Features	Precision	Recall	F_1
500	0.98	0.517	0.677
1000	0.981	0.552	0.707
2000	0.896	0.635	0.743
5000	0.817	0.656	0.728
10000	0.766	0.632	0.692
20000	0.747	0.635	0.687
50000	0.697	0.584	0.636
75745	0.697	0.56	0.621

Table 6.18: Precision, Recall and f_1 of Conclusion Basis

6.3.2 Rule-Based Approach

As explained in Chapter 4, out of 2160 sentences, 254 of the argumentative sentences were already divided into components of the argument: premise and conclusion or premise and premise, etc). The indicators that separates the components of the argument in the partition categories are identified manually (see Section 4.6.3 for a complete lists). A rule-based approach is applied to identify the accuracy of the indicator/separator that separates the components of the argument in the sentences' partition categories.

Indicator	Total Number	Positive effect	Negative effect	Accuracy (%)
(see	84	67	17	80
colon	35	21	14	60
(cf.	15	13	2	87
because	12	11	1	92
since	23	10	13	43

Table 6.19: Accuracy Results (in percentage) of Partition Indicators

Table 6.19 presents the top five separators: '(see', colon, '(cf.', 'because', and 'since' partition indicators, with their accuracy scores. The results are entirely in line with expectations. It was found that 'because' is the most effective discourse marker that separates the conclusion and premise with 92% accuracy. The indicators 'because' and 'since', as signalling a premise, were also identified by Trudy Govier in his book A practical study of argument [72]. Similarly, indicators '(cf.' and '(see' hold the second and the third position, yielding 87% and 80%, respectively. These last two indicators refer the reader to previous case-law documents which constitute reasons for the corresponding claim.

6.4 Evaluation

Several possibilities to find the arguments in legal documents are explored. The contribution of each module and their corresponding result are evaluated.

In the Argument Element Identifier module, three sets of experiments Basic, Multi-feature and Tree Kernel in order to find the best set of features and algorithms. In the Basic experiment, the complexity parameter of SVM and the number of trees of RF, are changed so that these algorithms provide the best possible model to detect other, similar data. The Multi-feature experiment was

conducted in order to obtain the best accuracy scores. Although, a Tree Kernel was used to process syntactic features, the obtained result was not significant when compared with the other approaches tested.

Next, an Argument Builder module was developed that builds arguments from sets of argumentative sentences using fuzzy clustering techniques. Such a strategy is innovative in that it introduces a clustering technique to gather the components to form an argument. Word2vec and Sentence Closeness features have already proven their significant role in clustering the argumentative sentences.

During the process, the DSCA were developed to transfer fuzzy membership values to the appropriate cluster form. This algorithm play a vital role in accomplishing the proposed task. Furthermore, a threshold value was defined in the DSCA to create a hard-cluster. Such threshold values depends on the quality of the features. If used with well-performing features, a larger threshold value can be used when separating the components to the group. Similarly, the ACIA was developed for evaluation of the proposed system. The main role of the algorithm is to map the arguments that is identified by the system to arguments of the gold-standard. However, the time complexity of the ACIA presents a problem for case-laws that consists of large number of arguments during evaluation procedures. Since this is not a theoretical limitation, but an implementation problem which is believed to be solved easily.

In order to identify the premise and conclusion of the arguments in the legal domain texts we deployed two experiments: the first aiming to identify the premise and conclusion of the argumentative sentences in our corpora, the second searching for the notation that separates these components within the argumentative sentences. Albeit promising, the F_1 levels reached by our first experiments were very high. A possible reason for the accuracy results obtained may be due to an imbalance in the datasets, which are comprised of 70% premise sentences but only 30% conclusions. The second experiment deployed a rule-based approach, using punctuation or characters/words as separators. The results can be considered quite satisfactory for some of the separators such as '(see' and '(cf.'. However, the punctuation marks performed poorly. Once again, the overall results are quite encouraging and support the need for the creation of a new argument mining framework.

6.5 Summary

The obtained results for each module are presented and discussed. First, machine learning algorithms were evaluated. Then an attempt was made to identify the best set of features. Second, a clustering technique was applied to group argumentative sentences. In this procedure, several milestones were achieved, such as finding an optimum number of clusters/arguments, dealing with *threshold value* while converting soft clusters to hard clusters, and developing an algorithm to evaluate the performance of module. Even though this module needs further improvement, the approach deployed did fulfill its objective of clustering the argumentative sentence into arguments. For future work, more features such as semantic similarities will be added to try to improve the results. We also plan to reduce the time complexity of the ACIA, allowing analysis of corpora containing a higher number of arguments. Third, an analysis of the results of the argument structure is presented. In this module, a statistical approach was used to find the accuracy of detecting whether an argumentative sentence is a premise or a conclusion. Subsequently, a rule-based approach to identify tokens that may separate the components of arguments (i.e. premise and conclusion) within each argument was used.

7

Conclusion and Future Work

This thesis has discussed and presented a method for extracting arguments from legal documents. First of all, a set of case-law documents was annotated to establish a gold-standard dataset. After this, the task was divided into three stages:

The first stage was dedicated to extracting argumentative sentences from legal texts, along with finding the most appropriate machine learning algorithms to obtain the correct text from the legal documents. The most popular machine learning algorithms in argument mining were surveyed, but it was difficult to determine the best machine learning algorithm for dealing with argumentative sentences. Two of the most popular algorithms (SVM and RF) were evaluated to select the best performing algorithm, along with an appropriate set of parameters. The most viable and successful algorithm for detecting arguments from the legal domain turned out to be SVM. Once this was determined, several other experiments were conducted to find the most efficient and accurate results.

After analyzing experiments, and by considering performance, various features were 'merged' to obtain an improved f_1 value of 0.686. This is considered satisfactory, and an improvement over results achieved by previous researchers.

The second stage is concerned with grouping argumentative sentences to form a coherent argument. After identifying which are the argumentative sentences, it is necessary to organize these sentences into a related set, known as an argument. The usual method for carrying out this stage is to delineate arguments by identifying their boundaries, a technique which is extensively explored in the AI & Law literature (see in the Chapter 3). As detailed in Section 5.2, argument boundary detection is plagued by several limitations which we tried to overcome by applying a novel clustering technique that groups argumentative sentences into a cluster of potential arguments. Its primary goal was to collect the components (i.e. argumentative sentence consisting of premises and a conclusion) and group them into an argument. However, one of the most difficult parts of running the clustering algorithm is that it requires the number of arguments within a text to be specified. Determining this critical parameter in advance requires the application of prediction techniques as several other researchers have pointed out [11, 38, 53, 75, 205]. The ambiguity introduced by the dual nature of sentences (i.e. a sentence can be associated with more than one argument) has to be handled along with the issue of which clustering algorithm to use. To be able to evaluate algorithm performance, an evaluation procedure was also introduced, creating two new algorithms based on the ACIA and the DSCA. Using this combination of techniques, satisfactory levels of performance were achieved (described in Chapter 6). Our approach is innovative in that it brings together a number of diverse technologies for the handling of arguments.

The matter of argument structure was also investigated. For this, a statistical method was used to determine the accuracy of categorizing an argumentative sentence as either a premise or a conclusion. As mentioned in Chapter Chapter 6, there are three possible kinds of structure for each argumentative sentence: 'Single Annotation', 'Overlap Annotation' and 'Sentence Partition'. The statistically based approach to identifying argumentative sentences in the 'Single Annotation' category worked quite well. However, it is considerably more difficult to determine membership in the 'Overlap Annotation' and 'Sentence Partition' categories due to the potentially complex structure of an argumentative sentence. To address this issue, the training data was processed by annotating the sentences, considering one of the components (i.e. either premise or conclusion) and measuring the performance, then retesting using the remaining component while re-measuring the performance. As for the last category, 'Sentence Partition', individual sentences are examined looking for components of an argument which are separated from each other by certain characters, punctuation marks and/or discourse markers. A rule-based approach was used to identify the tokens that demarcate these components inside the sentence. Among the three categories deployed, 'Sentence Partition' was, by

far, the most problematic to deal with.

7.1 Future Direction

During the investigation period, some areas of future research were envisaged:

A Revision of ECHR Corpus
As described above, one of the tasks we had to perform was to transfer the print-format corpus to an electronic format. During the process of annotating several discrepancies were found between the datasets. For instance, citations are always considered as premises, but several citations were found to be erroneously marked as conclusions (a mistake made by the human annotator). The ECHR corpus probably needs to be reviewed again by an expert legal team. Albeit arduous, this requirement is extremely important as it has the potential to increase the value of the corpus immensely.

Further Work on Citations
It is characteristic of a legal corpus that many, if not all case-laws are interconnected to each other through citations. In this example, excerpted from the ECHR corpus, the first phrase in bold is the conclusion, and the text in italics is the premise. **The notion of security of person has not been given an independent interpretation** (see *in this respect Selçuk and Asker v. Turkey, nos. NUMBER, Commission's report of DATE, §§ 185-187)*.

A human annotator will usually identify the components of an argument on the basis of a citation. First, the annotator is required to check the report provided by the citation and then will identify the first phase of the sentence as the conclusion, requiring the use of extensive background knowledge. On the other hand, the classifier marks the citation (*in italics*) as the premise, and the bold part as a conclusion. The classifier is trained on datasets where citations are similar in format to each other but the conclusion always in a different format, making it difficult for the classifier to make the same judgments a human might make. As a result, the classifier may achieve low accuracy. Future work might address this issue by collecting the 'hidden information' required to train the classifier. For that, it is necessary to access the particular (related) content/information from the report that provided the citation link. After that, the classifier needs to be trained with the case-law document along with the additional data from the report, aiming to achieve higher classification performance.

Finding more precise features
General categories of features were used but there are many other significant features that can be

tested to improve performance. More research aiming to improve the system by adding further and more specific features is recommended. For instance, semantic similarity might be an appropriate feature for clustering argumentative sentences into arguments.

Standardization of Threshold Values

During the work, the DSCA was extended to allocate fuzzy membership values to the appropriate cluster. A threshold value was defined in the DSCA to create a hard-cluster in the appropriate format. The issue of threshold value selection was of primary concern during the process of development, as its value depends upon the quality of the features. If there are high-quality features, a larger threshold value is needed; if not, a lower threshold value is required when separating the components into each group (see Chapter 5 for a detailed discussion of these issues). The concept of a threshold value has already been extensively studied [6, 88], but it is important to perform further research in this domain to explore better ways of converting soft clustering into hard clustering.

Improve Clustering Techniques

Defining the size of clusters is still an open research problem. To define the optimum number of clusters in the FCM, a technique proposed by Xie and Beni [205] and the Latent Dirichlet Allocation (LDA) exploited by (Cao *et al.* [38]) have been used. Analysis of these achieved indicate that they are not very satisfactory (see more in Section 5.2.1) and are very sensitive to the quality of features available. It would be interesting to compare them to other metric of LDA: Arun *et al.* [11], Griffiths and Steyvers [75] and Deveaud *et al.*, [53] and also with other techniques the Elbow [181] and Silbocette [162] methods or the Gap Statistic [183] approach, which are competing schemes for identifying the optimum cluster size.

Improving ACIA

The ACIA was used to find the best mapping between the proposed clustering and the gold-standard clustering. The algorithm matches an argument predicted by the system to its closest match in the gold-standard. However, the complexity of the algorithm is very high: $O(n^4)$. Due to this complexity, we were forced to select only case-laws with fewer than ten arguments. This is understood to be implementation problem that can be easily improved upon.

Relation Identification

The clustering technique was chosen over that of using boundary detection. Boundary detection was unsuitable due to the lack of the existing comparable relation between components in the ECHR

gold-standard sets. Therefore, upgrading the ECHR corpus with relations between the components of the argument and also between arguments marked in the gold-standard set is required, one can then compare the results of the boundary detection technique to those obtained by clustering.

7.2 Summary

The main goal of the work was to propose an architecture and equip it with the best set of features and algorithms to find arguments from legal cases. Some parts of the investigation were quite successful; other parts have identified significant research gaps suitable for future exploration. It is possible to conclude that with the appropriate features, the right classifier and suitable techniques applied to a well-structured text, the goal of our research can be achieved. Overall, the obtained results are quite promising and support the proposal for the creation of a new argument mining framework. [145]

A

Sample of Judgment Case-laws

This case-laws is adapted from [134] “

THIRD SECTION

CASE OF GIRARDI v. AUSTRIA

(Application no. 50064/99)

JUDGMENT

STRASBOURG

11 December 2003

FINAL

11/03/2004

This judgment will become final in the circumstances set out in Article 44 § 2 of the Convention. It may be subject to editorial revision. In the case of Girardi v. Austria,
The European Court of Human Rights (Third Section), sitting as a Chamber composed of:
Mr G. RESS, President,
Mr L. CAFLISH,
Mr P. KŪRIS,
Mr R. TÜRMEN,
Mr J. HEDIGAN,
Mrs H.S. GREVE
Mrs E. STEINER, judges,
and Mr V. BERGER, Section Registrar,
Having deliberated in private on 20 November 2003, Delivers the following judgment, which was adopted on that date:

PROCEDURE

1. The case originated in an application (no. 50064/99) against the Republic of Austria lodged with the Court under Article 34 of the Convention for the Protection of Human Rights and Fundamental Freedoms ("the Convention") by an Austrian national, Elisabeth Girardi ("the applicant"), on 9 July 1999.

2. The Austrian Government ("the Government") were represented by their Agent, Mr Mautner-Markhof.

3. On 4 July 2002 the Third Section declared the application partly inadmissible and decided to communicate the complaint concerning the length of the proceedings to the Government. Under the provisions of Article 29 § 3 of the Convention, it decided to examine the merits of the application at the same time as its admissibility.

THE FACTS

4. The applicant was born in 1951 and lives in Vienna.
   She is the mother of M, L and R, born in wedlock in 1973, 1974 and 1976, respectively. The spouses separated in 1982. Custody of L and M was assigned to the applicant, the custody of R to the father.

5. In December 1989 M was admitted in a public girls' home as she refused to stay with her mother. She stayed there until January 1992. From December 1989 until September 1995 custody proceedings concerning the temporary transfer of M's custody to the Vienna Youth Welfare Office for the time M had spent at the girls' home were pending before the Austrian courts.

A. The Youth Welfare Office's request for reimbursement of expenses

6. On 3 January 1990 the Vienna Youth Welfare Office, on behalf of M, filed a request with the Floridsdorf District Court that the applicant should pay a monthly contribution to the expenses incurred for M's stay in the girls' home.

7. The file was later on transferred to the competent Juvenile Court and, in January 1990, the court heard M's parents.

8. On 8 March 1991 the Youth Welfare Office reduced the amount of the requested monthly contribution.

9. On 10 April 1991 the President of the Juvenile Court granted the applicant's motion for bias against the competent court clerk (Rechtspfleger).

10. A hearing scheduled for 25 July 1991 was cancelled due to the applicant's illness. Further hearings scheduled for 2 September 1991 and 11 September 1991 had to be cancelled because the court's attempts to deliver the summons to the applicant were unsuccessful.

11. On 10 February 1992 the Juvenile Court ordered that the applicant had to pay ATS 2,500 in monthly maintenance for M. The applicant appealed, claiming that she was fit to work to an extent of 75% only.

12. On 4 March 1992 the case was assigned to another judge as the competent judge had declared himself biased.

13. On 13 May 1992 the Appeal Chamber quashed the decision and remitted the case back to the Juvenile Court, instructing the latter to take a new decision after having supplemented its proceedings. In particular, it stated that the first instance court ought to appoint a forensic medical expert in order to establish the applicant's fitness to work.

14. On 20 May 1998 the Juvenile Court ordered the applicant to pay ATS 1,550 in monthly maintenance for M. At that stage of the proceedings, no expert had been heard yet.

15. Referring to the Appeal Chamber's decision of 13 May 1992, the applicant appealed, again relying on her reduced fitness to work.

16. On 13 August 1998 the Juvenile Court appointed an expert in forensic medicine to file a report on the question as to which extent the applicant's capacities to earn her living were reduced.

17. The applicant appealed against this decision, claiming that it no longer made sense to appoint a medical expert, now that the court had already dismissed her request by a decision of 20 May 1998. Further, she claimed that there was no need for a further report as, in this respect, she had already submitted two reports of different medical officers (*Amtsarzt*).

18. On 17 and 20 August 1998 the applicant filed motions for bias against the court clerk (Rechtspfleger) I.S., who was dealing with her case, claiming that the appointment of a further medical expert was not justified, that I.S. was handling the case file in a negligent manner, namely that several documents were missing from the file, and that I.S. had been rude to her on the telephone.

19. On 25 August 1998 the President of the Vienna Juvenile Court (*Präsident des Jugendgerichtshofs*) dismissed her motion for bias, finding that the mere fact that she had appointed a medical expert was not sufficient to cast doubt upon I.S.' impartiality. He also noted that there were no documents missing from the file.

20. On 17 September 1998 the Appeal Chamber dismissed the applicant's appeal against the appointment of a medical expert, but granted her appeal against the decision of 20 May 1998. In this respect, it referred the case to the Juvenile Court for supplementing the taking of evidence, namely to comply with its decision of 13 May 1992.

21. On 21 and 23 March 1999 the applicant requested that, pursuant to Section 91 of the Courts Act (*Gerichtsorganisationsgesetz*), a time-limit be fixed for the decision on the Youth Welfare Office's application of 3 January 1990.

22. On 23 March 1999 the applicant filed a motion for bias against I.S., claiming that the latter had not been available to her during office hours and that she had refused to give her information requested over the telephone.

23. On 29 March 1999 the President of the Vienna Juvenile Court dismissed her motion as being unfounded.

24. On 30 March 1999 the President rejected her appeal against this decision, as the relevant provisions of the Court Clerks Act (*Rechtspflegergesetz*) did not provide for such remedy.

25. On 8 April 1999 the applicant was summoned by the appointed medical expert to undergo a medical examination at the Institute for Forensic Medicine (*Institut für Gerichtsmedizin*) on 22 April 1999.

26. It appears that the applicant filed numerous complaints with the President of the Juvenile Court, again claiming that documents were missing from the file and that I.S. as well as various judges of the Juvenile Court were biased.

27. On 4 May 1999 the President of the Juvenile Court decided to exclude I.S. from the proceedings. He noted that the latter had expressed that she considered herself biased following a telephone conversation in the course of which the applicant had said she would kill her daughter if I.S. continued to harass her. In these circumstances, the President found it advisable that the matter be re-assigned in accordance with the Juvenile Court's rules on the distribution of cases (*Geschäftsverteilung*).

28. On the same day, the Juvenile Court dismissed the applicant's requests for a time-limit to be set. Referring to the applicant's numerous requests, complaints and motions for bias filed with the court, it found that there was no indication of a lack of due diligence on behalf of the Juvenile Court, it being rather the applicant who prevented that a decision on the merits had been taken so far.

29. On 17 May 1999 the Vienna Youth Welfare Office withdrew its request dated of 3 January 1990.

30. Thereupon, the applicant, on 27 May 1999, withdrew all requests and complaints still pending before the Juvenile Court at that stage.

B. The applicant's request for reimbursement of expenses

31. From 30 July 1990 to 3 September 1990 M stayed with her mother. The latter, on 4 September 1990 filed a request with the Juvenile Court, claiming reimbursement of her expenses incurred during this period.

32. In September 1990 the Vienna Youth Welfare Office reimbursed the applicant for M's stay with her from 30 July 1990 to 21 August 1990.

33. On 10 August 1993 the Juvenile Court dismissed the applicant's request for expenses incurred during the rest of the period.

34. On 30 August 1993 the President of the Vienna Juvenile Court dismissed the applicant's motion of bias against the competent judge. On 30 December 1993 the Vienna Court of Appeal granted the applicant's appeal against this decision and quashed the decision.

35. On 20 January 1994 the Appeal Chamber of the Juvenile Court again dismissed the applicant's motion for bias. On 6 May 1994 the Court of Appeal rejected the applicant's appeal. A further appeal to the Supreme Court was to no avail. A further motion for bias against the President of the Juvenile Court was to no avail either.

36. On 5 January 1995 the Appeal Chamber quashed the decision of 10 August 1993 and remitted the case back to the first instance court.

37. On 19 April 1998 the applicant requested that, pursuant to Section 91 of the Courts Act, a time-limit be fixed for the decision on her application of 4 September 1990.

38. On 8 June 1998 the President of the Vienna Juvenile Court ordered the Juvenile Court to decide on the applicant's request no later than on 31 July 1998.

39. On 5 August 1998 the Juvenile Court dismissed the applicant's request for maintenance payments of 4 September 1990.

40. The applicant appealed against this decision.

41. It appears from the documents submitted that the applicant filed several complaints with the Vienna Court of Appeal (*Oberlandesgericht*), claiming that I.S. had not complied with the time limit set by the President of the Juvenile Court because she had gone on holidays, that the competent judicial officer, I.S. was to be found at her office only twice a week and that she had been extraordinarily impolite to her.

42. Thereupon, the President of the Juvenile Court, on 31 August 1998, informed the applicant that both I.S.'s office hours as well as her right to vacation were in accordance with her assignment. He also expressed his regret that, if, in the course of one of the applicant's numerous telephone calls, I.S. might have acted in a slightly indignant way. However, he emphasised that the applicant's allegations had remained unproved.

43. On 17 September 1998 the Appeal Chamber dismissed her appeal against the Juvenile Court's decision of 5 August 1998 as being unfounded. Further, it stated that there was no further appeal on points of law in the applicant's case as it did not raise questions of law of fundamental importance (*Ausspruch über die Unzulässigkeit der ordentlichen Revision*).

44. Nevertheless, the applicant filed an extraordinary appeal on points of law (*ausserordentliche Revision*) with the Supreme Court.

45. Referring to an amendment of Section 14 a of the Non-Contentious Proceedings Act (*Ausserstreitgesetz*), the Supreme Court on 18 December 1998 remitted the case back to the Vienna Juvenile Appeal Court. According to that provision, instead of filing an extraordinary appeal on points of law with the Supreme Court, a party to non-contentious proceedings must now request the Court of Appeal to re-consider its opinion on the admissibility of an ordinary appeal on points of law. The Supreme Court found that, even if in her appeal the applicant had not explicitly requested the Juvenile Appeal Court to declare that a further appeal on points of law be allowed, her appeal should have been understood in such a way.

46. Thereupon, on 11 January 1999 the Juvenile Appeal Court requested the applicant to remedy procedural defects of her appeal, namely to request that an ordinary appeal in her case be allowed.

47. As the applicant did not comply with this request, the Juvenile Appeal Court, on 25 February 1999, rejected her appeal.

THE LAW

I. ALLEGED VIOLATION OF ARTICLE 6 § 1 OF THE CONVENTION

48. The applicant complained that the length of the maintenance payment proceedings had been incompatible with the "reasonable time" principle as provided in Article 6 § 1 of the Convention, which reads as follows: In the determination of his civil rights and obligations....., everyone is entitled to a fair...hearing within reasonable time... by[a]... tribunal"

49. As regards the first set of proceedings, the period to be taken into consideration began on 3 January 1990 and ended on 22 May 1999. Thus, they lasted more than nine years and four months.

50. As regards the second set of proceedings, the period to be taken into consideration began on 4 September 1990 and ended on 25 February 1999. Thus, they lasted for more than eight years and five months.

A. Admissibility

51. The Court notes that this complaint is not manifestly ill-founded within the meaning of Article 35 § 3 of the Convention. It further notes that it is not inadmissible on any other grounds. It must therefore be declared admissible.

B. Merits

52. The Government submitted that the maintenance proceedings were complex. In particular, they had to be seen as a part of highly complex custody proceedings which required extensive expert opinions. While the authorities tried to conduct the proceedings expeditiously, the applicant filed a multitude of motions of bias, appeals and requests for extension of time-limits and therefore herself contributed considerably to the length of the proceedings. The Government further stressed that the applicant repeatedly thwarted attempts to deliver summons on her and failed to obey them.

53. The applicant did not submit any observations on these issues.

54. The Court reiterates that the reasonableness of the length of proceedings must be assessed in the light of the circumstances of the case and with reference to the criteria established by its case-law, particularly the complexity of the case, the conduct of the applicant and of the relevant authorities and what was at stake for the applicant in the dispute (*see, among many other authorities, Frydlender v. France [GC], no. 30979/96, § 43, ECHR 2000-VII).*

55. The Court considers that the present proceedings can clearly be distinguished from the custody proceedings, as they concerned merely the fixing of maintenance payments and were not particularly complex.

56. As regards the conduct of the applicant the Court has consistently held that applicants cannot be blamed for making full use of the remedies available to them under domestic law. However, an applicant's behaviour constitutes an objective fact which cannot be attributed to the respondent State and which must be taken into account for the purpose of determining whether or not the reasonable time referred to in Article 6 § 1 has been exceeded (*see Erkner and Hofbauer v. Austria, no. 9616/81, Commission decision of 23 April 1987, A 117, § 68*)

57. In the present case, the Court acknowledges that the applicant had filed numerous requests, complaints and motions and had repeatedly failed to obey the authorities' summons. Although such conduct contributed to prolonging the proceedings, it is not in itself sufficient to explain the length of the extensive proceedings.

58. On the other hand, the Court notes that there are substantial delays attributable to the authorities. In particular, in the first set of proceedings, there is a period of inactivity of more than two years (from 3 January 1990 to 10 February 1992) while the case was pending before the Vienna Juvenile Court, and a further one of six years (from 13 May 1992 to 20 May 1998) before that court took a new decision after the first one had been quashed on appeal. In the second set of proceedings, there is a period of inactivity of some three years (from 4 September 1990 to 10 August 1993), while the case was pending before the Vienna Juvenile Court, and a further such period of three years and seven months (from 5 January 1995 to 5 August 1998) before that court took a new decision after the first one had been quashed on appeal. The Court cannot find that the Government has given sufficient explanation for these delays that occurred.

59. The Court therefore finds that the overall length of the proceedings cannot be regarded as "reasonable". Accordingly, there has been a violation of Article 6 § 1 of the Convention.

II. APPLICATION OF ARTICLE 41 OF THE CONVENTION

60. Article 41 of the Convention provides:

"If the Court finds that there has been a violation of the Convention or the Protocols thereto, and if the internal law of the High Contracting Party concerned allows only partial reparation to be made, the Court shall, if necessary, afford just satisfaction to the injured party."

61. The applicant has not filed a claim for just satisfaction. Accordingly, the Court considers that no award can be made under this provision.

FOR THESE REASONS, THE COURT UNANIMOUSLY

1. Declares the application admissible;

2. Holds that there has been a violation of Article 6 § 1 of the Convention;

Done in English, and notified in writing on 11 December 2003, pursuant to Rule 77 §§ 2 and 3 of the Rules of Court.

Vincent BERGER	Georg RESS
Registrar	President

"

B

Sample of Decision Case-laws

This case-laws is adapted from [135] “

AS TO THE ADMISSIBILITY OF

Application No. 16841/90 by Harald PFARRMEIER against Austria

The European Commission of Human Rights sitting in private on 10 May 1993, the following members being present:

MM. C.A. NØRGAARD, President
J.A. FROWEIN
F. ERMACORA

E. BUSUTTIL
G. JÖRUNDSSON
A.S. GÖZÜBÜYÜK
A. WEITZEL
H.G. SCHERMERS
H. DANELIUS
Mrs. G.H. THUNE
Sir Basil HALL
Mr. C.L. ROZAKIS
Mrs. J. LIDDY
MM. M.P. PELLONPÄÄ
B. MARXER
G.B. REFFI
M.A. NOWICKI

Mr. H.C. KRÜGER, Secretary to the Commission

Having regard to Article 25 of the Convention for the Protection of Human Rights and Fundamental Freedoms;

Having regard to the application introduced on 13 June 1990 by Harald Pfarrmeier against Austria and registered on 10 July 1990 under file No. 16841/90;

Having regard to:

- the report provided for in Rule 47 of the Rules of Procedure of the Commission;

- the observations submitted by the respondent Government on 21 February 1992 and the observations in reply submitted by the applicant on 5 October 1992 ;

- the submissions of the parties at the oral hearing on 10 May 1993;

Having deliberated;

Decides as follows:

THE FACTS

The applicant is an Austrian citizen who lives in Bregenz. He is represented before the Commission by Mr. L. W. Weh, a lawyer practising in Bregenz.

The facts of the case, as submitted by the parties, may be summarised as follows:

On 11 June 1987 the applicant was fined by a penal order (Straferkenntnis) AS 9,000 with provision for 360 hours' detention in default for failure to submit to a breath test, contrary to Section 99 (1) (b) of the Road Traffic Act 1960 (Straßenverkehrsordnung). He appealed to the Regional Government (Landesregierung) which, on 11 November 1987, rejected his appeal.

On 23 March 1988 the Administrative Court (Verwaltungsgerichtshof) quashed the decision of the Regional Government of 11 November 1987 and remitted the case to that authority. The Regional Government's second decision, of 23 December 1988, reduced the penalty from AS 9,000 to AS 5,000 and the period of imprisonment in default from 360 hours to 200 hours.

The applicant's complaint to the Constitutional Court (Ver- fassungsgerichtshof) was rejected on 10 March 1989 on the ground that it had no sufficient prospect of success and that the case was not outside the competence of the Administrative Court. The Constitutional Court referred principally to its own case-law on Article 6 of the Convention in finding that the application had no sufficient prospect of success.

On 10 November 1989 the Administrative Court gave its second decision in the case. It found that it was not prevented from considering that the factual position had been determined in a relevant and conclusive way, although it was not able to review whether the defence's version of the facts was correct. Accordingly, the Administrative Court could not decide whether the applicant had or had not spoken of a "good session" (drinking). As to the applicant's complaint that his lawyer had not been able to examine a witness, the court noted that an oral hearing was not a necessary part of the administrative criminal proceedings. As to the alleged unconstitutionality of the Austrian reservation to Article 5 of the Convention, the Court referred to previous case-law. The complaint was dismissed as a whole.

COMPLAINTS

The applicant alleges a violation of Article 6 of the Convention in that the administrative criminal proceedings brought against him were determined initially by the administrative authorities which did not constitute independent and impartial tribunals within the meaning of Article 6 para. 1 of the Convention, and subsequently by the Constitutional Court and the Administrative Court, the scope of whose review is not sufficient to comply with Article 6 of the Convention, and which cannot decide the case themselves.

He also makes specific complaints about the nature of the administrative authorities' examination of the case, including his inability to put questions to prosecution witnesses, and about the inevitably

partial status of experts.

PROCEEDINGS BEFORE THE COMMISSION

The application was introduced on 13 June 1990 and registered on 10 July 1990.

On 16 October 1991 the Commission decided to request the parties to submit their written observations on the admissibility and merits of the application.

The respondent Government submitted their observations on 21 February 1992 and the applicant submitted his observations on 5 October 1992.

On 15 February 1993 the Commission decided to hear the parties as to the admissibility and merits of this case and Applications Nos. 15523/89, 15527/89, 15963/90, 16713/90 and 16718/90. At the hearing the parties were represented as follows:

For the Government:

Ambassador F. Cede Legal Adviser, Federal Ministry for Foreign Affairs, Agent

Ms. S. Bernegger Federal Chancellery, Adviser

For the applicant:

Mr. W.L. Weh Representative

THE LAW

The applicant alleges violation of Article 6 (Art. 6) of the Convention.

The Government submit that the Austrian reservation to Article 5 (Art. 5) of the Convention prevents the Commission from examining the case. They accept, however, that if the reservation does not prevent an examination of the case, then the review of administrative decisions by the Administrative Court and Constitutional Court was not sufficiently wide to comply with Article 6 para. 1 (Art. 6-1) of the Convention. They add, in this respect, that although the offence for which the applicant was convicted under Section 99 (1)(b) of the Road Traffic Act 1960 (refusing to take a breath test) was not, as such, in force on the date of the reservation, the law then in force did impose an obligation on road users to drive with reasonable consideration for other road users and to pay such attention as is required for the maintenance of order, safety and traffic efficiency.

The Government consider that the absence of an oral public and direct hearing is covered by the Austrian reservation to Article 6 (Art. 6) of the Convention. They also point out that the applicant did not make a complaint about the absence of a hearing before the Administrative Court.

The applicant considers that the Austrian reservation to Article 5 (Art. 5) of the Convention is neither valid nor applicable in the present case. He agrees that the scope of review by the Constitutional Court and Administrative Court does not comply with Article 6 (Art. 6) of the Convention. He considers that the reservation to Article 6 (Art. 6), if valid, is not applicable to the present proceedings.

In connection with Article 144 para. 2 of the Federal Constitution, the Government consider that, although that provision provides for non-acceptance of a constitutional complaint on grounds which were not in force in 1958 when the reservation was made, the possibility for the Constitutional Court to refuse to deal with appeals against decisions without giving detailed reasons is only a procedural limitation and not a substantive one. They point out that any appeal lodged with the Constitutional Court against a decision is subject to comprehensive review.

The applicant in this respect considers that the limitation of the Constitutional Court's jurisdiction by Article 144 para. 2 of the Federal Constitution does not meet the requirements of the reservation, even if it applies.

The Commission finds that the application raises complex issues of law under the Convention, including questions concerning the Austrian reservations to Articles 5 and 6 (Art. 5, 6) of the Convention, the determination of which must be reserved for an examination on the merits.

The application cannot therefore be declared manifestly ill- founded within the meaning of Article 27 para. 2 (Art. 27-2) of the Convention. No other ground for declaring it inadmissible has been established.

For these reasons the Commission unanimously

DECLARES THE APPLICATION ADMISSIBLE, without prejudging the merits of the case."

Secretary to the Commission	President of the Commission
(H. C. KRÜGER)	(C. A. NØRGAARD)

C

Rest of the results of Clustering technique

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	F_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.521	0.379	0.439	0.500	0.200	0.346	0.253	0.245	0.355	0.479	0.408	0.344	0.438	0.367	0.399	0.444
03	0.321	0.333	0.327	0.389	0.239	0.702	0.356	0.242	0.412	0.762	0.535	0.410	0.333	0.405	0.366	0.421
13	0.270	0.453	0.338	0.289	0.247	0.578	0.346	0.239	0.244	0.467	0.321	0.289	0.255	0.461	0.329	0.273
16	0.125	1.000	0.222	0.125	0.161	0.740	0.265	0.162	0.452	0.412	0.431	0.429	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.358	0.588	0.445	0.341	0.343	0.375	0.358	0.321	0.250	1.000	0.400	0.250
31	0.548	0.595	0.570	0.556	0.206	0.738	0.322	0.213	0.398	0.357	0.376	0.318	0.367	0.476	0.414	0.333
32	0.461	0.563	0.507	0.400	0.200	0.824	0.322	0.206	0.224	0.552	0.319	0.233	0.341	0.502	0.406	0.385
35	0.452	0.514	0.481	0.474	0.237	0.543	0.330	0.267	0.369	0.514	0.430	0.364	0.571	0.529	0.549	0.571
39	0.542	0.706	0.613	0.526	0.171	0.669	0.272	0.167	0.290	0.575	0.385	0.294	0.379	0.488	0.427	0.400
40	0.387	0.690	0.496	0.364	0.282	0.710	0.403	0.289	0.400	0.640	0.492	0.381	0.311	0.480	0.377	0.364
41	0.411	0.688	0.514	0.381	0.336	0.750	0.464	0.333	0.454	0.500	0.476	0.429	0.375	0.688	0.485	0.364
42	0.281	0.367	0.318	0.400	0.189	0.662	0.294	0.200	0.319	0.474	0.382	0.346	0.440	0.469	0.454	0.421

Table C.1: Case laws Number, Precision, Recall and f_1 and Cluster Purity value of the System Prediction at m=12, t=0.001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.479	0.354	0.407	0.556	0.212	0.279	0.241	0.270	0.363	0.388	0.375	0.421	0.375	0.325	0.348	0.438
03	0.393	0.333	0.361	0.438	0.353	0.524	0.422	0.333	0.426	0.548	0.479	0.429	0.405	0.333	0.366	0.438
13	0.368	0.531	0.435	0.391	0.228	0.192	0.208	0.276	0.206	0.233	0.219	0.258	0.333	0.344	0.339	0.364
16	0.125	1.000	0.222	0.125	0.285	0.445	0.348	0.270	0.504	0.452	0.477	0.455	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.347	0.383	0.364	0.357	0.408	0.375	0.391	0.360	0.254	1.000	0.405	0.253
31	0.571	0.548	0.559	0.563	0.250	0.619	0.357	0.256	0.429	0.286	0.343	0.375	0.560	0.548	0.554	0.438
32	0.491	0.541	0.515	0.429	0.177	0.496	0.261	0.187	0.224	0.419	0.292	0.245	0.489	0.480	0.484	0.450
35	0.536	0.514	0.525	0.600	0.254	0.507	0.338	0.303	0.533	0.486	0.508	0.500	0.571	0.529	0.549	0.571
39	0.583	0.806	0.677	0.556	0.157	0.456	0.234	0.173	0.542	0.519	0.530	0.471	0.379	0.488	0.427	0.400
40	0.533	0.490	0.511	0.429	0.317	0.570	0.407	0.333	0.500	0.540	0.519	0.500	0.381	0.480	0.425	0.444
41	0.417	0.500	0.455	0.400	0.295	0.375	0.330	0.400	0.500	0.500	0.500	0.500	0.408	0.563	0.473	0.400
42	0.281	0.367	0.318	0.400	0.210	0.436	0.284	0.229	0.390	0.379	0.384	0.389	0.464	0.469	0.467	0.444

Table C.2: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=12, t=0.0001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.479	0.354	0.407	0.556	0.438	0.260	0.326	0.421	0.469	0.425	0.446	0.444	0.375	0.325	0.348	0.438
03	0.393	0.333	0.361	0.438	0.452	0.452	0.452	0.500	0.426	0.440	0.433	0.474	0.429	0.321	0.367	0.467
13	0.423	0.539	0.474	0.400	0.242	0.178	0.205	0.333	0.250	0.331	0.285	0.333	0.408	0.483	0.443	0.350
16	0.145	1.000	0.254	0.147	0.439	0.382	0.408	0.400	0.504	0.452	0.477	0.455	0.132	0.978	0.233	0.131
30	0.339	0.683	0.453	0.327	0.392	0.358	0.374	0.360	0.408	0.375	0.391	0.360	0.285	0.575	0.381	0.298
31	0.571	0.548	0.559	0.563	0.595	0.500	0.543	0.471	0.429	0.286	0.343	0.375	0.583	0.548	0.565	0.467
32	0.481	0.452	0.466	0.500	0.361	0.576	0.444	0.333	0.389	0.341	0.363	0.400	0.489	0.480	0.484	0.474
35	0.536	0.514	0.525	0.600	0.250	0.443	0.320	0.281	0.533	0.486	0.508	0.500	0.571	0.529	0.549	0.571
39	0.563	0.744	0.641	0.529	0.204	0.306	0.245	0.273	0.542	0.519	0.530	0.471	0.400	0.488	0.439	0.421
40	0.533	0.490	0.511	0.429	0.430	0.530	0.475	0.412	0.500	0.540	0.519	0.500	0.514	0.480	0.497	0.533
41	0.350	0.375	0.362	0.417	0.313	0.375	0.341	0.462	0.500	0.500	0.500	0.500	0.533	0.563	0.548	0.429
42	0.281	0.367	0.318	0.400	0.505	0.355	0.417	0.400	0.390	0.379	0.384	0.389	0.464	0.469	0.467	0.444

Table C.3: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=12, t=0.00001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.479	0.354	0.407	0.556	0.438	0.260	0.326	0.444	0.344	0.400	0.370	0.389	0.375	0.325	0.348	0.438
03	0.393	0.333	0.361	0.438	0.581	0.429	0.493	0.529	0.521	0.440	0.478	0.529	0.429	0.321	0.367	0.467
13	0.423	0.539	0.474	0.400	0.242	0.178	0.205	0.350	0.250	0.331	0.285	0.333	0.408	0.483	0.443	0.350
16	0.271	0.402	0.324	0.273	0.494	0.382	0.431	0.424	0.504	0.452	0.477	0.455	0.287	0.382	0.327	0.304
30	0.392	0.438	0.414	0.414	0.392	0.358	0.374	0.360	0.408	0.375	0.391	0.360	0.401	0.400	0.401	0.440
31	0.571	0.548	0.559	0.563	0.619	0.500	0.553	0.500	0.429	0.286	0.343	0.375	0.583	0.548	0.565	0.467
32	0.481	0.452	0.466	0.500	0.452	0.498	0.474	0.450	0.481	0.341	0.399	0.556	0.489	0.480	0.484	0.474
35	0.536	0.514	0.525	0.600	0.207	0.271	0.235	0.333	0.533	0.486	0.508	0.500	0.571	0.529	0.549	0.571
39	0.563	0.744	0.641	0.529	0.267	0.306	0.285	0.286	0.542	0.519	0.530	0.471	0.400	0.488	0.439	0.421
40	0.533	0.490	0.511	0.429	0.400	0.430	0.414	0.400	0.500	0.540	0.519	0.500	0.500	0.430	0.462	0.500
41	0.350	0.375	0.362	0.417	0.313	0.375	0.341	0.462	0.500	0.500	0.500	0.500	0.521	0.438	0.476	0.417
42	0.281	0.367	0.318	0.400	0.469	0.283	0.353	0.389	0.390	0.379	0.384	0.389	0.464	0.469	0.467	0.444

Table C.4: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=12, t= 0.000001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.295	0.825	0.434	0.302	0.294	0.406	0.341	0.364	0.646	0.329	0.436	0.500	0.265	1.000	0.420	0.261
03	0.357	0.524	0.425	0.314	0.334	0.571	0.421	0.324	0.421	0.429	0.425	0.391	0.524	0.381	0.441	0.533
13	0.187	0.833	0.305	0.186	0.365	0.525	0.431	0.370	0.380	0.581	0.459	0.407	0.342	0.344	0.343	0.381
16	0.125	1.000	0.222	0.125	0.273	0.425	0.332	0.286	0.449	0.449	0.449	0.424	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.252	0.275	0.263	0.296	0.334	0.375	0.353	0.346	0.250	1.000	0.400	0.250
31	0.152	1.000	0.264	0.152	0.231	0.714	0.349	0.244	0.595	0.524	0.557	0.533	0.429	0.500	0.462	0.400
32	0.150	1.000	0.261	0.150	0.284	0.637	0.393	0.286	0.335	0.544	0.415	0.317	0.260	0.817	0.394	0.250
35	0.429	0.414	0.421	0.467	0.367	0.586	0.451	0.370	0.486	0.414	0.447	0.471	0.845	0.636	0.726	0.769
39	0.125	1.000	0.222	0.125	0.241	0.644	0.351	0.244	0.333	0.550	0.415	0.296	0.125	1.000	0.222	0.125
40	0.317	0.370	0.341	0.412	0.407	0.570	0.475	0.409	0.686	0.620	0.651	0.563	0.357	0.420	0.386	0.412
41	0.450	0.625	0.523	0.438	0.617	0.813	0.701	0.563	0.458	0.500	0.478	0.429	0.475	0.625	0.540	0.467
42	0.221	0.805	0.347	0.211	0.324	0.390	0.354	0.310	0.643	0.486	0.553	0.500	0.167	1.000	0.286	0.167

Table C.5: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=13, t=0.001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.573	0.485	0.526	0.556	0.321	0.221	0.262	0.368	0.656	0.367	0.470	0.563	0.583	0.471	0.521	0.556
03	0.619	0.429	0.506	0.563	0.307	0.488	0.377	0.303	0.714	0.429	0.536	0.600	0.524	0.381	0.441	0.533
13	0.204	0.558	0.299	0.211	0.329	0.344	0.337	0.381	0.602	0.581	0.591	0.524	0.342	0.344	0.343	0.400
16	0.125	1.000	0.222	0.125	0.426	0.481	0.452	0.400	0.449	0.449	0.449	0.424	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.252	0.275	0.263	0.320	0.351	0.363	0.357	0.360	0.250	1.000	0.400	0.250
31	0.163	0.952	0.279	0.161	0.381	0.476	0.423	0.471	0.595	0.524	0.557	0.533	0.429	0.500	0.462	0.400
32	0.182	0.956	0.305	0.176	0.470	0.526	0.497	0.440	0.593	0.485	0.534	0.474	0.444	0.424	0.434	0.435
35	0.429	0.414	0.421	0.467	0.605	0.443	0.511	0.563	0.595	0.414	0.489	0.571	0.845	0.636	0.726	0.769
39	0.132	1.000	0.233	0.129	0.442	0.619	0.515	0.429	0.362	0.525	0.429	0.368	0.135	1.000	0.238	0.134
40	0.350	0.370	0.360	0.438	0.567	0.530	0.548	0.500	0.486	0.520	0.502	0.500	0.367	0.420	0.392	0.438
41	0.517	0.563	0.539	0.500	0.625	0.625	0.625	0.583	0.438	0.438	0.438	0.417	0.683	0.625	0.653	0.583
42	0.440	0.498	0.467	0.429	0.433	0.414	0.424	0.389	0.643	0.486	0.553	0.500	0.230	0.964	0.371	0.225

Table C.6: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=13, t=0.0001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	F_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.698	0.485	0.573	0.625	0.342	0.221	0.268	0.412	0.656	0.367	0.470	0.563	0.625	0.450	0.523	0.600
03	0.619	0.429	0.506	0.563	0.405	0.333	0.366	0.412	0.714	0.429	0.536	0.600	0.524	0.381	0.441	0.533
13	0.508	0.628	0.561	0.500	0.413	0.344	0.375	0.400	0.602	0.581	0.591	0.550	0.342	0.344	0.343	0.400
16	0.125	1.000	0.222	0.125	0.437	0.481	0.458	0.424	0.449	0.449	0.449	0.424	0.125	1.000	0.222	0.125
30	0.265	1.000	0.419	0.263	0.252	0.275	0.263	0.320	0.351	0.363	0.357	0.360	0.272	1.000	0.428	0.275
31	0.317	0.714	0.439	0.313	0.524	0.571	0.547	0.533	0.595	0.524	0.557	0.533	0.429	0.500	0.462	0.400
32	0.335	0.785	0.470	0.326	0.481	0.393	0.433	0.474	0.648	0.485	0.555	0.529	0.500	0.396	0.442	0.474
35	0.429	0.414	0.421	0.467	0.619	0.414	0.496	0.571	0.667	0.414	0.511	0.615	0.845	0.636	0.726	0.769
39	0.352	0.588	0.440	0.346	0.400	0.431	0.415	0.421	0.362	0.525	0.429	0.368	0.310	0.613	0.412	0.250
40	0.400	0.370	0.384	0.467	0.587	0.530	0.557	0.533	0.519	0.520	0.520	0.533	0.400	0.420	0.410	0.467
41	0.517	0.563	0.539	0.500	0.625	0.625	0.625	0.583	0.438	0.438	0.438	0.417	0.683	0.625	0.653	0.583
42	0.464	0.440	0.452	0.389	0.433	0.414	0.424	0.389	0.643	0.486	0.553	0.500	0.431	0.598	0.501	0.414

Table C.7: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=13, t=0.00001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.760	0.485	0.593	0.667	0.342	0.221	0.268	0.412	0.656	0.367	0.470	0.563	0.625	0.450	0.523	0.600
03	0.619	0.381	0.472	0.533	0.405	0.333	0.366	0.412	0.714	0.429	0.536	0.600	0.524	0.381	0.441	0.533
13	0.521	0.614	0.564	0.500	0.413	0.344	0.375	0.400	0.602	0.581	0.591	0.550	0.342	0.344	0.343	0.400
16	0.237	0.629	0.344	0.233	0.437	0.481	0.458	0.424	0.449	0.449	0.449	0.424	0.233	0.647	0.342	0.222
30	0.479	0.617	0.539	0.483	0.252	0.275	0.263	0.320	0.351	0.363	0.357	0.360	0.374	0.475	0.418	0.375
31	0.536	0.524	0.530	0.467	0.524	0.571	0.547	0.533	0.595	0.524	0.557	0.533	0.429	0.500	0.462	0.400
32	0.574	0.396	0.469	0.611	0.481	0.393	0.433	0.474	0.648	0.485	0.555	0.529	0.500	0.396	0.442	0.529
35	0.500	0.414	0.453	0.500	0.619	0.414	0.496	0.571	0.667	0.414	0.511	0.615	0.845	0.636	0.726	0.769
39	0.383	0.656	0.484	0.421	0.400	0.431	0.415	0.421	0.383	0.463	0.419	0.389	0.542	0.588	0.564	0.471
40	0.400	0.370	0.384	0.467	0.587	0.530	0.557	0.533	0.519	0.520	0.520	0.533	0.400	0.420	0.410	0.467
41	0.517	0.563	0.539	0.500	0.625	0.625	0.625	0.583	0.438	0.438	0.438	0.417	0.683	0.625	0.653	0.583
42	0.464	0.440	0.452	0.389	0.433	0.414	0.424	0.389	0.643	0.486	0.553	0.500	0.583	0.521	0.551	0.556

Table C.8: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=13, t= 0.000001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.200	1.000	0.333	0.200	0.571	0.392	0.465	0.529	0.471	0.423	0.446	0.526	0.200	1.000	0.333	0.200
03	0.200	1.000	0.333	0.200	0.343	0.333	0.338	0.412	0.450	0.500	0.474	0.500	0.200	1.000	0.333	0.200
13	0.183	0.833	0.301	0.183	0.492	0.497	0.494	0.364	0.358	0.544	0.432	0.357	0.183	0.833	0.301	0.183
16	0.125	1.000	0.222	0.125	0.384	0.377	0.380	0.308	0.137	1.000	0.241	0.135	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.294	0.358	0.323	0.357	0.407	0.433	0.420	0.407	0.250	1.000	0.400	0.250
31	0.152	1.000	0.264	0.152	0.548	0.571	0.559	0.533	0.429	0.500	0.462	0.381	0.152	1.000	0.264	0.152
32	0.150	1.000	0.261	0.150	0.472	0.461	0.467	0.474	0.422	0.489	0.453	0.421	0.150	1.000	0.261	0.150
35	0.198	1.000	0.330	0.198	0.500	0.414	0.453	0.500	0.409	0.693	0.514	0.400	0.198	1.000	0.330	0.198
39	0.125	1.000	0.222	0.125	0.285	0.550	0.376	0.300	0.394	0.575	0.467	0.417	0.125	1.000	0.222	0.125
40	0.214	1.000	0.353	0.214	0.533	0.430	0.476	0.429	0.413	0.480	0.444	0.438	0.214	1.000	0.353	0.214
41	0.250	1.000	0.400	0.250	0.617	0.688	0.650	0.533	0.438	0.438	0.438	0.417	0.250	1.000	0.400	0.250
42	0.167	1.000	0.286	0.167	0.390	0.462	0.423	0.429	0.294	0.486	0.366	0.290	0.167	1.000	0.286	0.167

Table C.9: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=20, t= 0.001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.200	1.000	0.333	0.200	0.571	0.392	0.465	0.529	0.533	0.410	0.464	0.500	0.200	1.000	0.333	0.200
03	0.200	1.000	0.333	0.200	0.343	0.333	0.338	0.412	0.524	0.405	0.457	0.529	0.200	1.000	0.333	0.200
13	0.183	0.833	0.301	0.183	0.439	0.497	0.466	0.400	0.308	0.331	0.319	0.350	0.183	0.833	0.301	0.183
16	0.125	1.000	0.222	0.125	0.219	0.271	0.242	0.250	0.258	0.529	0.347	0.279	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.436	0.421	0.428	0.400	0.411	0.408	0.410	0.400	0.250	1.000	0.400	0.250
31	0.152	1.000	0.264	0.152	0.548	0.571	0.559	0.533	0.643	0.452	0.531	0.467	0.152	1.000	0.264	0.152
32	0.150	1.000	0.261	0.150	0.491	0.461	0.475	0.500	0.517	0.522	0.519	0.444	0.150	1.000	0.261	0.150
35	0.198	1.000	0.330	0.198	0.500	0.414	0.453	0.538	0.548	0.493	0.519	0.500	0.198	1.000	0.330	0.198
39	0.125	1.000	0.222	0.125	0.390	0.588	0.468	0.316	0.300	0.500	0.375	0.389	0.125	1.000	0.222	0.125
40	0.214	1.000	0.353	0.214	0.533	0.430	0.476	0.429	0.547	0.480	0.511	0.500	0.214	1.000	0.353	0.214
41	0.250	1.000	0.400	0.250	0.688	0.625	0.655	0.583	0.438	0.438	0.438	0.417	0.250	1.000	0.400	0.250
42	0.167	1.000	0.286	0.167	0.429	0.462	0.445	0.474	0.345	0.402	0.372	0.368	0.167	1.000	0.286	0.167

Table C.10: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=20, t=0.0001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.213	1.000	0.352	0.212	0.571	0.392	0.465	0.529	0.533	0.410	0.464	0.500	0.206	1.000	0.342	0.203
03	0.200	1.000	0.333	0.200	0.343	0.333	0.338	0.412	0.524	0.405	0.457	0.529	0.200	1.000	0.333	0.200
13	0.198	0.819	0.319	0.196	0.439	0.497	0.466	0.400	0.308	0.331	0.319	0.350	0.221	0.806	0.347	0.215
16	0.125	1.000	0.222	0.125	0.219	0.271	0.242	0.250	0.321	0.301	0.311	0.314	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.436	0.421	0.428	0.400	0.411	0.408	0.410	0.400	0.250	1.000	0.400	0.250
31	0.152	1.000	0.264	0.152	0.548	0.571	0.559	0.533	0.643	0.452	0.531	0.467	0.152	1.000	0.264	0.152
32	0.156	1.000	0.269	0.153	0.491	0.461	0.475	0.500	0.517	0.522	0.519	0.444	0.161	1.000	0.277	0.156
35	0.215	1.000	0.354	0.212	0.500	0.414	0.453	0.538	0.548	0.493	0.519	0.500	0.215	1.000	0.354	0.214
39	0.125	1.000	0.222	0.125	0.390	0.588	0.468	0.316	0.300	0.500	0.375	0.389	0.134	1.000	0.236	0.132
40	0.251	0.950	0.397	0.255	0.533	0.430	0.476	0.429	0.547	0.480	0.511	0.500	0.266	1.000	0.420	0.268
41	0.377	0.938	0.538	0.367	0.688	0.625	0.655	0.583	0.438	0.438	0.438	0.417	0.321	0.813	0.461	0.323
42	0.179	1.000	0.303	0.176	0.429	0.462	0.445	0.474	0.345	0.402	0.372	0.368	0.180	0.964	0.303	0.179

Table C.11: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=20, t=0.00001

Case No.	N-gram				Sentence Closeness				Word2Vec				Combine Feature			
	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur	Pre	Rec	f_1	Pur
02	0.545	0.423	0.476	0.500	0.571	0.392	0.465	0.529	0.533	0.410	0.464	0.500	0.440	0.421	0.430	0.444
03	0.652	0.571	0.609	0.522	0.343	0.333	0.338	0.412	0.524	0.357	0.425	0.500	0.508	0.488	0.498	0.400
13	0.357	0.553	0.434	0.429	0.439	0.497	0.466	0.400	0.308	0.331	0.319	0.350	0.273	0.253	0.262	0.440
16	0.125	1.000	0.222	0.125	0.219	0.271	0.242	0.250	0.340	0.301	0.319	0.324	0.125	1.000	0.222	0.125
30	0.250	1.000	0.400	0.250	0.436	0.421	0.428	0.400	0.411	0.408	0.410	0.400	0.254	1.000	0.405	0.253
31	0.217	0.738	0.335	0.229	0.548	0.571	0.559	0.533	0.643	0.452	0.531	0.467	0.452	0.500	0.475	0.364
32	0.452	0.507	0.478	0.414	0.491	0.461	0.475	0.500	0.517	0.522	0.519	0.444	0.340	0.669	0.451	0.333
35	0.616	0.657	0.636	0.478	0.500	0.414	0.453	0.538	0.548	0.493	0.519	0.500	0.540	0.586	0.562	0.476
39	0.336	0.788	0.471	0.375	0.390	0.588	0.468	0.316	0.300	0.500	0.375	0.389	0.504	0.831	0.627	0.344
40	0.518	0.860	0.647	0.444	0.533	0.430	0.476	0.429	0.547	0.480	0.511	0.500	0.406	0.770	0.531	0.385
41	0.313	0.375	0.341	0.385	0.688	0.625	0.655	0.583	0.438	0.438	0.438	0.417	0.470	0.625	0.536	0.421
42	0.265	0.757	0.393	0.268	0.429	0.462	0.445	0.474	0.345	0.402	0.372	0.368	0.286	0.650	0.397	0.310

Table C.12: Case laws Number, Precision, Recall, f_1 and Cluster Purity value of the System Prediction at m=20, t=0.00001

D

Python code for Sentence representation using Word2vec

The average of all the vectors of the words present in the sentence is calculated using code developed Python. The code is as follows:

```
#!/usr/bin/env python
#!/home/darklord/anaconda2/bin/python

import sys
import getopt
import bleach
import xml.etree.ElementTree as ET
```

```
import os
import re
import csv
import pickle



import pandas as pd
import numpy as np
import re
import timeit
import gensim

from sklearn import cross_validation
from sklearn import svm
from sklearn import metrics
from sklearn import preprocessing
from sklearn.feature_extraction.text import TfidfVectorizer
from bs4 import BeautifulSoup
from xml.etree.ElementTree import ParseError



reload(sys)
sys.setdefaultencoding("ISO-8859-1")

def makeFeatureVec(words, model, num_features):
    featureVec = np.zeros((num_features,),dtype="float32")
    nwords = 0.
    index2word_set = set(model.index2word)
    wordsNotInDict = []

    try:
        words = words.split(" ")
    except AttributeError:
        featureVec = np.random.rand(num_features)
        return featureVec, wordsNotInDict
```

```
    for word in words:
        if word in index2word_set:
            #import ipdb; ipdb.set_trace()
            nwords = nwords + 1.
            featureVec = np.add(featureVec,model[word])
        else:
            wordsNotInDict.append(word)

    if nwords == 0.:
        featureVec = np.random.rand(num_features)
    else:
        featureVec = np.divide(featureVec,nwords)
    return featureVec, wordsNotInDict

def getAvgFeatureVecs(all_texts, model, num_features):
    counter = 0.
    reviewFeatureVecs = np.zeros((len(all_texts),num_features),dtype="float32")
    lineOfWordsNotInDict = []

    for one_line in all_texts:
        #print "============"+ str(counter)+"============="
        reviewFeatureVecs[counter], wordsNotInDict =
        makeFeatureVec(one_line, model, num_features)
        lineOfWordsNotInDict.append(wordsNotInDict)
        #print "============"+str(counter)+"============="
        counter = counter + 1.

    return reviewFeatureVecs, lineOfWordsNotInDict



def main(argv):
    num_features = 100
    current_working_dir = './'
    model_dir = "word2vec-models/wikipedia-only-trained-on-my-machine/"
```

```
    relations = {'english': {'truth_file': 'summary-english-truth.txt',\
                             'model_file': 'wiki.en.tex.d100.model'
                             }
                 }

    model_file = current_working_dir + model_dir + relations['english']['model_file']
    model = gensim.models.Word2Vec.load(model_file)

    all_lines = []

    with open('corpus.txt', 'r') as fp:
        for line in fp:
            all_lines.append(line)


    trainDataVecs, trashedWords = getAvgFeatureVecs( all_lines,
    model, num_features )
    return trainDataVecs



if __name__ == "__main__":
    trainDataVecs = main(sys.argv[1:])
```

E

Java Code for ACIA algorithm

Java code for ACIA algorithm

```
1  package ACIA;
2
3  import java.util.ArrayList;
4  import java.util.Arrays;
5  import java.util.Collections;
6  import java.util.LinkedHashMap;
7  import java.util.Scanner;
8
9  public class Solver {
10     final int dx[] = {+1, -1, 0, 0};
```

```
11      final int dy[] = {0, 0, +1, -1};
12
13      double[][] mat;
14      int[][] taken;
15      int n, m;
16      ArrayList<Entry> path, bestPath;
17      double current, best;
18      long iter=0;
19
20
21  //    public void solve(){
22      public LinkedHashMap<Integer,
23      ArrayList<Integer>> solve(double [][] matrix){
24          LinkedHashMap<Integer, ArrayList<Integer>> newLinked = new
25          LinkedHashMap<Integer, ArrayList<Integer>>();
26          mat=matrix;
27          m=mat.length;
28          n=mat.length;
29          taken = new int[m][m];
30
31          taken = new int[n][m];
32          path = new ArrayList<>();
33          bestPath = new ArrayList<>();
34          current = 0;
35          best = 0;
36          backtrack();
37          System.out.println("Best Value: " + best);
38          int counter=1;
39          for(Entry e: bestPath){
40              ArrayList<Integer>
41              finalArrayList = new ArrayList<Integer>();
42              finalArrayList.add(e.x);
43              finalArrayList.add(e.y);
44              counter++;
45              newLinked.put(counter, finalArrayList);
46              System.out.println(e.x + " " + e.y);
```

```
47          }
48
49          return newLinked;
50      }
51      private void backtrack() {
52      //      System.out.println(iter++);
53          if(check()){
54              updateBest();
55              return;
56          }
57          double mxVal=0;
58          for (int i = 0; i < n; i++) {
59              for (int j = 0; j < m; j++) {
60                  if(taken[i][j]==0){
61                      mxVal=Math.max(mxVal, mat[i][j]);
62                  }
63              }
64          }
65
66          for (int i = 0; i < n; i++) {
67              for (int j = 0; j < m; j++) {
68                  if(taken[i][j] == 0 && mat[i][j] == mxVal){
69                      add(i, j);
70                      backtrack();
71                      rem(i, j);
72                      if(mxVal == 0){
73                          updateBest();
74                          return; //Since improving now is impossible
75                      }
76                  }
77              }
78          }
79      }
80
81      private void updateBest() {
82          if(current > best){
```

```
83              best = current;
84              bestPath.clear();
85              for(Entry e: path) bestPath.add(e);
86              Collections.sort(bestPath);
87          }
88      }
89
90      private void add(int x, int y) {
91          for (int i = 0; i < 4; i++) {
92              for (int nx = x, ny = y; isValid(nx, ny);
93              nx += dx[i], ny += dy[i]){
94                  taken[nx][ny]++;
95              }
96          }
97          path.add(new Entry(x, y));
98          current += mat[x][y];
99      }
100
101     private void rem(int x, int y) {
102         for (int i = 0; i < 4; i++) {
103             for (int nx = x, ny = y; isValid(nx, ny);
104             nx += dx[i], ny += dy[i]){
105                 taken[nx][ny]--;
106             }
107         }
108         path.remove(path.size() - 1);
109         current -= mat[x][y];
110     }
111
112     private boolean isValid(int x, int y) {
113         return (0 <= x && x < n) && (0 <= y && y < m);
114     }
115
116     private boolean check() {
117         for (int i = 0; i < n; i++) {
118             for (int j = 0; j < m; j++) {
```

```
119                 if(taken[i][j] == 0) return false;
120             }
121         }
122         return true;
123     }
124 
125     private class Entry implements Comparable<Entry>{
126         int x, y;
127 
128         public Entry(int x, int y){
129             this.x = x;
130             this.y = y;
131         }
132 
133         public int compareTo(Entry other) {
134             if(x == other.x) return y - other.y;
135             return x - other.x;
136         }
137     }
138 }
```

F

Computational Resources

Computational Resources that were used for our experiments are as follows:

Hardware Overview

There are all together 18 nodes along with management node in the computing machine.

- 18 nodes + management node
  - 554 cores
  - 1108 threads
  - 2336GB RAM
  - Shared home (NFS)

- Connectivity
  - Infineband + Gigabit ethernet
- GPGPUs
  - NVIDIA Tesla K20c
  - AMD/ATI Radeon HD 7990
- Co processors
  - 2 × Intel Xeon Phi 7210
    - 61 cores
    - 16GB RAM

Operating Software Overview

- Operating System
  - Debian Jessie (8.5)
- Common libraries and compilers
  - C, C++, python, gfortran, . . .
- Run environments
  - OpenCL
  - OpenMP
  - OpenMPI

Bibliography

[1] Rob Abbott, Pranav Anand, Brian Ecker, and Marilyn A Walker. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *LREC*, 2016.

[2] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. The evaluation of sentence similarity measures. *Data warehousing and knowledge discovery*, pages 305–316, 2008.

[3] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *ArgMining@ ACL*, pages 64–68, 2014.

[4] Latifa Mohammed Al-Abdulkarim. *Representation of case law for argumentative reasoning*. PhD thesis, University of Liverpool, 2017.

[5] Khalid Al-Khatib, Henning Wachsmuth, Mathias Hagen, Jonas Köhler, and Benno Stein. Cross-domain mining of argumentative text through distant supervision. In *Proceedings of NAACL-HLT*, pages 1395–1404, 2016.

[6] Moh'd Belal Al-Zoubi, Amjad Hudaib, and Bashar Al-Shboul. A fast fuzzy clustering algorithm. In *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, volume 3, pages 28–32, 2007.

[7] Vincent Aleven and Kevin D Ashley. Teaching case-based argumentation through a model and examples empirical evaluation of an intelligent learning environment. In *Artificial intelligence in education*, volume 39, pages 87–94. IOS Press, 1997.

[8] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[9] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23(10):1062–1093, 2008.

[10] Ron Artstein. *Inter-annotator Agreement*, pages 297–313. Springer Netherlands, Dordrecht, 2017.

[11] Rajkumar Arun, Venkatasubramaniyan Suresh, C. E. Veni Madhavan, and M. N. Narasimha Murthy. On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 391–402. Springer, 2010.

[12] Katie Atkinson. Google scholar citations. Available at `https://scholar.google.com/citations?user=fvQicksAAAAJ&hl=en` (2018-03-01).

[13] Katie Atkinson, Trevor Bench-Capon, and Peter McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.

[14] M. Azar. Argumentative text as rhetorical structure: An application of rhetorical structure theory. *Argumentation*, 13(1):97–114, 1999.

[15] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[16] T.J.M. Bench-Capon. Deep models, normative reasoning and legal expert systems. In *Proceedings of the 2nd international conference on Artificial intelligence and law*, pages 37–45. ACM, 1989.

[17] Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G. Conrad, Enrico Francesconi, Thomas F. Gordon, Guido Governatori, Jochen L. Leidner, David D. Lewis, Ronald P. Loui, L. Thorne McCarty, Henry Prakken, Frank Schilder, Erich Schweighofer, Paul Thompson, Alex Tyrrell, Bart Verheij, Douglas N. Walton, and Adam Z. Wyner. A history of ai and law in 50 papers: 25 years of the international conference on ai and law. *Artificial Intelligence and Law*, 20(3):215–319, 2012.

[18] Trevor J M Bench-Capon. Google scholar citations. Available at `https://scholar.google.ca/citations?user=8qMKdPwAAAAJ&hl=en` (2018-03-01).

[19] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence Review*, 33(3):211–259, 2010.

[20] Jamal Bentahar, Bernard Moulin, and Brahim Chaib-draa. Commitment and argument network: a new formalism for agent communication. In *Workshop on Agent Communication Languages*, pages 146–165. Springer, 2003.

[21] Jamal Bentahar, Bernard Moulin, John-Jules Ch Meyer, and Brahim Chaib-draa. A computational model for conversation policies for agent communication. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 178–195. Springer, 2004.

[22] Philippe Besnard and Anthony Hunter. *Elements of argumentation*, volume 47. MIT press Cambridge, 2008.

[23] Floris Bex. Google scholar citations. Available at `https://scholar.google.com/citations?user=7fibZnIAAAAJ&hl=en` (2018-03-01).

[24] Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. Implementing the argument web. *Communications of the ACM*, 56(10):66–73, 2013.

[25] Floris Bex, Henry Prakken, Chris Reed, and Douglas Walton. Towards a formal account of reasoning about evidence: argumentation schemes and generalisations. *Artificial Intelligence and Law*, 11(2-3):125–165, 2003.

[26] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[27] James C. Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2):191 – 203, 1984.

[28] Or Biran and Owen Rambow. Identifying justifications in written dialogs. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 162–168. IEEE, 2011.

[29] Lawrence Birnbaum, Margot Flowers, and Rod McGuire. Towards an ai model of argumentation. In *Proceedings of the First AAAI Conference on Artificial Intelligence*, AAAI'80, pages 313–315. AAAI Press, 1980.

[30] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[31] Filip Boltuzic and Jan Snajder. Back up your stance: Recognizing arguments in online discussions. In *ArgMining@ ACL*, pages 49–58, 2014.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Simon J. Buckingham Shum, Victoria Uren, Gangmin Li, Bertrand Sereno, and Clara Mancini. Modeling naturalistic argumentation in research literatures: Representation and interaction design issues. *International Journal of Intelligent Systems*, 22(1):17–47, 2007.

[34] Mann William C. and Sandra A. Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[35] Elena Cabrio and Serena Villata. Natural language arguments: A combined approach. In *ECAI*, volume 242, pages 205–210, 2012.

[36] Elena Cabrio and Serena Villata. Towards a benchmark of natural language arguments. *CoRR*, abs/1405.0941, 2014.

[37] Colin Campbell and Yiming Ying. Learning with support vector machines. *Synthesis lectures on artificial intelligence and machine learning*, 5(1):1–95, 2011.

[38] Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. A density-based method for adaptive lda model selection. *Neurocomputing*, 72(7-9):1775–1781, 2009.

[39] Chesñevar Carlos, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, and Steven Willmott. Towards an argument interchange format. *The knowledge engineering review*, 21(4):293–316, 2006.

[40] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer, 2003.

[41] Lucas Carstens and Francesca Toni. Towards relation based argumentation mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34, 2015.

[42] Lucas Carstens and Francesca Toni. Using argumentation to improve classification in natural language problems. *ACM Transactions on Internet Technology (TOIT)*, 17(3):30, 2017.

[43] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. *Ann arbor mi*, 48113(2):161–175, 1994.

[44] International Justice Resource Center. European court of human rights | international justice resource center. Available at `http://www.ijrcenter.org/european-court-of-human-rights/#gsc.tab=0` (2018-04-015).

[45] Marilyn J. Chambliss. Text cues and strategies successful readers use to construct the gist of lengthy written arguments. *Reading Research Quarterly*, 30(4):778–807, 1995.

[46] Shu-Nu Chang. Teaching argumentation through the visual models in a resource-based learning environment. In *Asia-Pacific Forum on Science Learning and Teaching*, volume 8, pages 1–15. The Education University of Hong Kong, Department of Science and Environmental Studies, 2007.

[47] Y. Chevaleyre, P.E. Dunne, U. Endriss, J. Lang, N. Maudet, and J. A. Rodrıguez-Aguilar. Final report on the technical forum group on multiagent resource allocation during agentlink iii.

[48] Danish Contractor, Yufan Guo, and Anna Korhonen. Using argumentative zones for extractive summarization of scientific articles. In *coling*, volume 12, pages 663–678, 2012.

[49] Irving M. Copi, Carl Cohen, and Kenneth McMahon. *Introduction to logic*. Routledge, 2016.

[50] Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[51] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM SIGIR Forum*, volume 51, pages 148–159. ACM, 2017.

[52] Gil Filipe da Rocha. Argmine: Argumentation mining from text. 2016.

[53] Romain Deveaud, Eric SanJuan, and Patrice Bellot. Accurate and effective latent concept modeling for ad hoc information retrieval. *Document numérique*, 17(1):61–84, 2014.

[54] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321 – 357, 1995.

[55] William N. Dunn. *Public policy analysis*. Routledge, 2015.

[56] Paul E. Dunne. Google scholar citations. Available at `https://scholar.google.ca/citations?user=anFySfEAAAAJ&hl=en` (2018-03-01).

[57] S DUTRA, E DAGNEAUX, F MEUNIER, and M PAQUOT. International corpus of learner english-version 2. *Studies in Second Language Acquisition*, 33(1):140, 2011.

[58] Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *EMNLP*, pages 2236–2242, 2015.

[59] Frans H. van Eemeren and Tjark Kruiger. Identifying argumentation schemes. In *Reasonableness and Effectiveness in Argumentative Discourse*, pages 703–712. Springer, 2015.

[60] Sibel Erduran, Shirley Simon, and Jonathan Osborne. Tapping into argumentation: Developments in the application of toulmin's argument pattern for studying science discourse. *Science education*, 88(6):915–933, 2004.

[61] Vanessa Wei Feng and Graeme Hirst. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 987–996. Association for Computational Linguistics, 2011.

[62] Gregory Ferenstein. How technology destroyed the once substantive presidential debate and techcrunch. Available at `https://www.independent.co.uk/life-style/gadgets-and-tech/artificial-intelligence-debate-argue-bbc-science-tech-research-a8118191.html` (2017-10-03).

[63] Eirini Florou, Stasinos Konstantopoulos, Antonis Kukurikos, and Pythagoras Karampiperis. Argument extraction for supporting public policy formulation. In *LaTeCH@ ACL*, pages 49–54, 2013.

[64] John Fox, Paul Krause, and Morten Elvang-Gøransson. Argumentation as a general framework for uncertain reasoning. In *Uncertainty in Artificial Intelligence, 1993*, pages 428–434. Elsevier, 1993.

[65] James B. Freeman. *Dialectics and the macrostructure of arguments: A theory of argument structure*, volume 10. Walter de Gruyter, 1991.

[66] Merce Garcia-Mila, Sandra Gilabert, Sibel Erduran, and Mark Felton. The effect of argumentative task goal on the quality of argumentative discourse. *Science Education*, 97(4):497–523, 2013.

[67] Tim van Gelder. Argument mapping with reason! able. *The American Philosophical Association Newsletter on Philosophy and Computers*, 2(1):85–90, 2002.

[68] Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *ArgMining@ ACL*, pages 39–48, 2014.

[69] Thomas F. Gordon. Google scholar citations. Available at `https://scholar.google.com/citations?user=rkrKQeEAAAAJ&hl=en` (2018-03-01).

[70] Thomas F. Gordon, Henry Prakken, and Douglas Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875–896, 2007.

[71] Theodosis Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news, blogs, and social media. In *Hellenic Conference on Artificial Intelligence*, pages 287–299. Springer, 2014.

[72] Trudy Govier. *A practical study of argument*. Cengage Learning, 2013.

[73] Sylvianne Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. International corpus of learner english. version 2, 2009.

[74] Wayne Grennan. *Informal logic: Issues and techniques*. McGill-Queen's Press-MQUP, 1997.

[75] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[76] Leo Groarke. Logic, art and argument. *Informal logic*, 18(2), 1996.

[77] Chinnappa Guggilla, Tristan Miller, and Iryna Gurevych. Cnn-and lstm-based claim classification in online user comments. In *COLING*, pages 2740–2751, 2016.

[78] Ivan Habernal and Iryna Gurevych. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, 2016.

[79] Ivan Habernal and Iryna Gurevych. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *ACL (1)*, 2016.

[80] Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 2017.

[81] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[82] Charles Leonard Hamblin. Fallacies. *Methuen, London*, 1970.

[83] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society.

[84] Andreas Hotho, Steffen Staab, and Gerd Stumme. Ontologies improve text document clustering. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 541–, Washington, DC, USA, 2003. IEEE Computer Society.

[85] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56, 2008.

[86] Patrick J. Hurley. *A concise Introduction to Logic*. MPS Limited, 2015.

[87] Compendium Institute. Compendium institute the community showcase. Available at `http://www.compendiuminstitute.org/` (2018-03-01).

[88] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[89] Thorsten Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999.

[90] Ralph H. Johnson and J. Anthony Blair. *Logical self-defense*. Idea, 2006.

[91] Daniel Jurafsky and James H. Martin. *Speech and language processing*, volume 3. Pearson London:, 2014.

[92] Juyeon Kang and Patrick Saint-Dizier. A discourse grammar for processing arguments in context. *Computational Models of Argument: Proceedings of COMMA 2014*, 266:43, 2014.

[93] Victor Alvin Ketcham. *The theory and practice of argumentation and debate*. Macmillan, 1914.

[94] Manfred Kienpointner. Alltagslogik struktur und funktion von argumentationsmustern. 1992.

[95] Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *ArgMining@ HLT-NAACL*, pages 1–11, 2015.

[96] Namhee Kwon, Eduard Hovy, Liang Zhou, and Stuart W. Shulman. Identifying and classifying subjective claims. In *Proceedings of the 8th annual international conference on Digital government research: bridging disciplines & domains*, pages 76–81. Digital Government Society of North America, 2007.

[97] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. Aifdb: Infrastructure for the argument web. In *COMMA*, pages 515–516, 2012.

[98] John Lawrence and Chris Reed. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 127–136, 2015.

[99] John Lawrence, Chris Reed, Colin Allen, Simon McAlister, Andrew Ravenscroft, and Bourget David. Mining arguments from 19th century philosophical texts using topic based modelling. In *Proceedings of the First Workshop on Argumentation Mining*, pages 79–87, 2014.

[100] KU LEUVEN. Liir - raquel mochales palau ph.d. Available at `http://liir.cs.kuleuven.be/phdStudents/raquelPhd.php` (2018-03-01).

[101] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, 2014.

[102] Yanjun Li, Soon M Chung, and John D Holt. Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering*, 64(1):381–404, 2008.

[103] Yuhua Li, Zuhair Bandar, David McLean, and James O'Shea. A method for measuring sentence similarity and its application to conversational agents. In *FLAIRS Conference*, pages 820–825, 2004.

[104] Brooks Lindsay. Debatepedia:about - debatepedia. Available at `http://www.debatepedia.org` (2011-11-14).

[105] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In *IJCAI*, volume 15, pages 185–191, 2015.

[106] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)*, 16(2):10, 2016.

[107] LiveJournal. Livejournal: Discover global communities of friends who share your unique passions and interests. Available at `https://www.livejournal.com` (2016-12-19).

[108] Clare Llewellyn, Claire Grover, Jon Oberlander, and Ewan Klein. Re-using an argument corpus to aid in the curation of social media collections. In *LREC*, volume 14, pages 462–468, 2014.

[109] Fabrizio Macagno and Douglas Walton. Argumentative reasoning patterns. 2006.

[110] Elias J MacEwan. *The essentials of argumentation*. DC Heath & Company, 1898.

[111] Jim D Mackenzie. Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8(1):117–133, 1979.

[112] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[113] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[114] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[115] Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[116] MathWorks. Fuzzy c-means clustering - matlab - simulink example. Available at `https://www.mathworks.com/help/fuzzy/examples/fuzzy-c-means-clustering.html` (2018-01-01).

[117] L. Thorne McCarty. Reflections on taxman: An experiment in artificial intelligence and legal reasoning. *Harv. L. Rev.*, 90:837, 1976.

[118] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[119] George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[120] Raquel Mochales and Marie-Francine Moens. Study on the structure of argumentation in case law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems*, pages 11–20, 2008.

[121] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[122] Raquel Mochales-Palau and Marie-Francine Moens. Study on sentence relations in the automatic detection of argumentation in legal cases. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 165:89, 2007.

[123] Marie-Francine Moens. Google scholar citations. Available at `https://scholar.google.pt/citations?user=O9hYMUUAAAAJ&hl=en&oi=sra` (2018-03-01).

[124] Marie-Francine Moens. *Information extraction: algorithms and prospects in a retrieval context*, volume 21. Springer Science & Business Media, 2006.

[125] Marie-Francine Moens. Argumentation mining: Where are we now, where do we want to be and how do we get there? In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, page 2. ACM, 2013.

[126] Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM, 2007.

[127] Prof. Marie-Francine Moens. Language intelligence & information retrieval lab (liir). Available at `http://liir.cs.kuleuven.be/index.php` (2018-01-01).

[128] Alessandro Moschitti. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 335. Association for Computational Linguistics, 2004.

[129] Andrew Nevins, David Pesetsky, and Cilene Rodrigues. Evidence and argumentation: A reply to everett (2009). *Language*, 85(3):671–681, 2009.

[130] Huy V. Nguyen and Diane J. Litman. Argument mining for improving the automated scoring of persuasive essays. 2018.

[131] Nidhi. Number of topics for lda on poems from elliston poetry archive. Available at `http://www.rpubs.com/MNidhi/NumberoftopicsLDA` (2017-03-31).

[132] Robert Nisbet, John Elder, and Gary Miner. *Handbook of statistical analysis and data mining applications*. Academic Press, 2009.

[133] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *LREC*, 2016.

[134] European Court of Human Rights. Hudoc - european court of human rights - judgment. Available at `http://hudoc.echr.coe.int/eng?i=001-61535` (2003-11-12).

[135] European Court of Human Rights. Hudoc - european court of human rights-decision. Available at `http://hudoc.echr.coe.int/eng?i=001-1573` (2003-11-12).

[136] University of Pittsburgh. Kevin d. ashley | pittlaw. Available at http://law.pitt.edu/people/kevin-d-ashley (2018-03-01).

[137] Nathan Ong, Diane Litman, and Alexandra Brusilovsky. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28, 2014.

[138] Raquel Mochales Palau and Aagje Ieven. Creating an argumentation corpus: do theories apply to real arguments?: a case study on the legal argumentation of the echr. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 21–30. ACM, 2009.

[139] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM, 2009.

[140] Dae Hoon Park and Catherine Blake. Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 1–9. Association for Computational Linguistics, 2012.

[141] Joonsuk Park and Claire Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, 2014.

[142] Joonsuk Park, Arzoo Katiyar, and Bishan Yang. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 39–44, 2015.

[143] Isaac Persing and Vincent Ng. Modeling prompt adherence in student essays. In *ACL (1)*, pages 1534–1543, 2014.

[144] John L Pollock. Defeasible reasoning. *Cognitive science*, 11(4):481–518, 1987.

[145] Prakash Poudyal. A machine learning approach to argument mining in legal documents. In *AI Approaches to the Complexity of Legal Systems*, pages 443–450. Springer, 2015.

[146] Henry Prakken. Google scholar citations. Available at = https://scholar.google.com/citations?user=ZyaLOy4AAAAJ&hl=en (2018-03-01).

[147] ProCon.org. Procon.org - pros and cons of controversial issues. Available at https://www.procon.org/ (2007-07-24) urldate = 2007-07-24.

[148] J. R. Quinlan. Induction of decision trees, Mar 1986.

[149] J. Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[150] Iyad Rahwan and Chris Reed. The argument interchange format. In *Argumentation in artificial intelligence*, pages 383–402. Springer, 2009.

[151] Chris Reed. How artificial intelligence could help us to win arguments | the independent. Available at `https://www.independent.co.uk/life-style/gadgets-and-tech/artificial-intelligence-debate-argue-bbc-science-tech-research-a8118191.html` (2017-12-27).

[152] Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. Language resources for studying argument. In *Proceedings of the 6th conference on language resources and evaluation-LREC 2008*, pages 91–100, 2008.

[153] Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979, 2004.

[154] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[155] Edwina L. Rissland and Kevin D. Ashley. A case-based system for trade secrets law. In *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, ICAIL '87, pages 60–66, New York, NY, USA, 1987. ACM.

[156] Gil Rocha, Cardoso Henrique Lopes, and Jorge Teixeira. Argmine: a framework for argumentation mining. In *Computational Processing of the Portuguese Language-12th International Conference, PROPOR*, pages 13–15, 2016.

[157] Mendes M.E.S. Rodrigues and L Sacks. A scalable hierarchical fuzzy clustering algorithm for text mining. In *Proceedings of the 5th international conference on recent advances in soft computing*, pages 269–274, 2004.

[158] Haggai Roitman, Shay Hummel, Ella Rabinovich, Benjamin Sznajder, Noam Slonim, and Ehud Aharoni. On the retrieval of wikipedia articles containing claims on controversial topics. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 991–996. International World Wide Web Conferences Steering Committee, 2016.

[159] Niall Rooney, Hui Wang, and Fiona Browne. Applying kernel methods to argumentation mining. In *FLAIRS Conference*, 2012.

[160] Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning*, 3(3):237–271, 2008.

[161] Sara Rosenthal and Kathleen McKeown. Detecting opinionated claims in online discussions. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 30–37. IEEE, 2012.

[162] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[163] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[164] Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news. In *ArgMining@ HLT-NAACL*, pages 56–66, 2015.

[165] Oliver Scheuer, Frank Loll, Niels Pinkwart, and Bruce M. McLaren. Computer-supported argumentation: A review of the state of the art. *International Journal of Computer-supported collaborative learning*, 5(1):43–102, 2010.

[166] Edward Schiappa and John P Nordin. *Argumentation: Keeping faith with reason*. Pearson, 2014.

[167] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press, 2008.

[168] Rosie Shier. Statistics: 1.1 paired t-tests. *Mathematics Learning Support Centre*, 12, 2004.

[169] Guillermo R Simari and Ronald P Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial intelligence*, 53(2-3):125–157, 1992.

[170] Maria Simosi. Using toulmin's framework for the analysis of everyday argumentation: Some methodological considerations. *Argumentation*, 17(2):185–202, 2003.

[171] Benjamin Soltoff. Statistical learning: support vector machines. Available at `https://cfss.uchicago.edu/persp009_svm.html` (2018-01-01).

[172] Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics*

(COLING 2014), pages 1501–1510. Dublin City University and Association for Computational Linguistics, August 2014.

[173] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *EMNLP*, pages 46–56, 2014.

[174] Christian Stab and Iryna Gurevych. Recognizing the absence of opposing arguments in persuasive essays. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 113–118, 2016.

[175] Christian Stab and Iryna Gurevych. Recognizing insufficiently supported arguments in argumentative essays. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 980–990, 2017.

[176] Christian Stab, Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *ArgNLP*, pages 21–25, 2014.

[177] Christian Matthias Edwin Stab. *Argumentative writing support by means of natural language processing*. PhD thesis, Technische Universität Darmstadt, 2017.

[178] Karsten Stegmann, Christof Wecker, Armin Weinberger, and Frank Fischer. Collaborative argumentation and cognitive elaboration in a computer-supported collaborative learning environment. *Instructional Science*, 40(2):297–323, 2012.

[179] Stephanie. T-test (student's t-test): Definition and examples - statistics how to. Available at `http://www.statisticshowto.com/probability-and-statistics/t-test/` (2018-01-06).

[180] Leo Strauss and Joseph Cropsey. *History of political philosophy*. University of Chicago Press, 2012.

[181] Catherine A Sugar. Techniques for clustering and classification with applications to medical problems. 1999.

[182] Simone Teufel. *Argumentative zoning: Information extraction from scientific text*. PhD thesis, University of Edinburgh, 2000.

[183] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[184] Stephen E. Toulmin. *The uses of argument*. Cambridge University Press, 2003.

[185] Stephen Edelston Toulmin, Richard D Rieke, and Allan Janik. *An introduction to reasoning*. Number Sirsi) i9780024211606. 1984.

[186] Frans H Van Eemeren and Rob Grootendorst. Fallacies in pragma-dialectical perspective. *Argumentation*, 1(3):283–301, 1987.

[187] Frans H Van Eemeren and Rob Grootendorst. Rationale for a pragma-dialectical perspective. *Argumentation*, 2(2):271–291, 1988.

[188] Frans H Van Eemeren and Rob Grootendorst. *A systematic theory of argumentation: The pragma-dialectical approach*, volume 14. Cambridge University Press, 2004.

[189] Tim Van Gelder. The rationale for rationale™. *Law, Probability & Risk*, 6(1-4):23–42, 2007.

[190] Vladimir N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[191] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[192] Bart Verheij. Google scholar citations. Available at `https://scholar.google.com/citations?user=vp1HB0YAAAAJ&hl=en` (2018-03-01).

[193] Anne von der Lieth Gardner. An artificial intelligence approach to legal reasoning. 1987.

[194] James F. Voss, Rebecca Fincher-Kiefer, Jennifer Wiley, and Laurie Ney Silfies. On the processing of arguments. *Argumentation*, 7(2):165–181, 1993.

[195] Marilyn A. Walker, Pranav Anand, Jean E Fox Tree, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *LREC*, pages 812–817, 2012.

[196] Douglas Walton. Google scholar citations. Available at `https://scholar.google.ca/citations?user=iRzoJwcAAAAJ&hl=en` (2018-03-01).

[197] Douglas Walton. *Fundamentals of critical argumentation*. Cambridge University Press, 2005.

[198] Douglas Walton, Chris Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

[199] Douglas N. Walton. Argumentation schemes for presumptive reasoning, 1996.

[200] Armin Weinberger and Frank Fischer. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & education*, 46(1):71–95, 2006.

[201] John Wilkinson, Purcell Weaver, et al. The new rhetoric: A treatise on argumentation, 1969.

[202] Wolfram. Fuzzy clustering. Available at `http://reference.wolfram.com/legacy/applications/fuzzylogic/Manual/12.html` (2018-01-01).

[203] Chase B. Wrenn. Naturalistic epistemology. 2003.

[204] Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. Approaches to text mining arguments from legal cases. In *Semantic processing of legal texts*, pages 60–79. Springer, 2010.

[205] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(8):841–847, August 1991.

[206] Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, 2013.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:

Universidade de Évora

Instituto de Investigação e Formação Avançada — IIFA

Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94

7002 - 554 Évora | Portugal

Tel: (+351) 266 706 581

Fax: (+351) 266 744 677

email: iifa@uevora.pt