UNIVERSITY OF
CAMBRIDGE

Argument mining with informal text

Yuxiao Ye

St Edmund's College

This dissertation is submitted on October, 2023 for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Yuxiao Ye
October, 2023

Abstract

Argument mining with informal text

Yuxiao Ye

The rapid growth of online discussions has led to a wealth of user-generated text, rich in arguments and diverse in nature. However, the complexity of these informal arguments presents a challenge for argument mining. Argument mining is the task of automatically analysing arguments, such that the unstructured information contained in them is converted into structured representations. Current practice in argument mining largely focuses on well-structured and edited formal text, but the annotation schemes and models developed for these simpler texts cannot account well for the phenomena found in informal text.

To capture the characteristics of informal arguments, I designed an annotation scheme which includes undercuts, a counterargument device that challenges the relationship between a premise and a claim. Other computational approaches conflate undercuts with direct attacks, a device where the truth of the claim or the premise itself is challenged. I also presented the resultant large-scale Quora dataset featuring informal arguments, complemented by a layer of annotation detailing complete argument structures.

I then proposed an end-to-end approach to argument mining based on dependency parsing. My approach uses new dependency representations for arguments and two new neural dependency parsers, one based on biaffine parsing and the other on GNNs. It comfortably beats a strong baseline on the Quora dataset. When applied to an existing benchmark dataset of formal arguments, my approach establishes a new state of the art. It is also the first automatic argument mining approach that is able to recognise undercuts.

Furthermore, I conducted a study on external knowledge integration for end-to-end argument mining, such as information from syntax, discourse, knowledge graphs, and large language models. I found that feature-based integration using GPT-3.5 is the most effective method among those I have surveyed.

Overall, I hope that my work, by providing automatic analyses of arguments in online discussions, will eventually foster better understanding among people with different opinions.

Acknowledgements

I was told many times that finishing a PhD was never meant to be easy. Now I am approaching to the end. Looking back, I would like to describe my journey to a PhD as a challenging yet rewarding expedition. I could have never made it to the destination without all the lovely and helpful people along the way.

First, I would like to express my deepest gratitude to my supervisor, Simone Teufel. You provided me with the invaluable opportunity to pursue a PhD just when I desperately needed a change in life. Your unwavering support and guidance enabled me to explore topics that truly resonated with me, and your profound academic expertise really helped with shaping my research. Also, you took care of me as a cherished friend, helping me with all the unexpected hurdles throughout the years. You made it possible.

I also want to thank Paula Buttery and Andreas Vlachos, who I honourably had as my first-year examiners. Your insightful observations and valuable advice on my PhD proposal helped put my research on the right track, and your positive feedback made me confident with my research ability.

I feel really fortunate to have so many friendly and smart colleagues in the lab. Yiwen, Guy, Josef, and Rowan, thank you for the inspiring discussions and wonderful time together.

My gratitude also goes to Toshiba Research Europe Limited for funding my research, and Tatsuya Izuha for approving this funding application. I am thankful to members of Toshiba's Cambridge Research Laboratory, including Svetlana Stoyanchev, Rama Doddipatla, Simon Keize, and Norbert Braunschweiler, for giving me very insightful suggestions in our regular meetings.

My heartfelt appreciation goes to all my friends who are always there for me. Ruiduan, You, and Yue, your empathy and consolation saved me every time when I was about to fall. Li, your virtual company always brought me laughter when I was feeling down.

Carlos, thank you for the everlasting positive energy from Sevilla. Lin, thank you for being my rock.

Lastly, I am especially grateful to my family. My mother, Jie Yang, thank you for your unconditional love and endless support, even though you have been suffering a lot. I could have never been who I am now without you.

Contents

Chapter 1

Introduction

The pervasive prevalence of the internet and social media stands as a defining feature of contemporary society. It has opened unprecedented avenues for individuals to express their opinions and engage in conversations on a wide range of subjects. A common approach for expressing these opinions, either to clarify one's stance or to persuade others, involves constructing an argument (Mercier and Sperber, 2011, Van Eemeren and Henkemans, 2016).

An example for arguments in online discussions is shown in Figure 1.1. The argument in the example is is an answer to the question "Do you support same sex marriage? Why?" It is taken from Quora, a popular question-answering platform where users can ask questions and get answers from other users. In this argument, the author starts by explaining why marriage is necessary for two adults in love. They then refute several popular reasons given by the opposing party. The author concludes by saying that they support same-sex marriage because those opposing reasons do not stand. This example gives us a taste of arguments in online discussions, which are usually less formal than those in argumentative essays and formal debates, but are often still rational and informative.

Social media allows people to express divergent opinions on the same subject and to reach many more people than was possible in earlier times. However, the ubiquity of the internet and social media also has some negative consequences. One of these is the growing polarisation between individuals holding different beliefs and opinions. This makes it increasingly important to promote productive communication and understanding among people with opposing perspectives.

To achieve this goal, we need to first analyse the arguments contained in online discussions. However, manually analysing and making sense of these arguments can be a daunting task due to the sheer volume of data. This is where argument mining comes into play. Argument mining is a computational technique that automatically identifies and extracts arguments from natural language texts (Peldszus and Stede, 2013, Green et al., 2014). It can convert unstructured textual information into structured argument

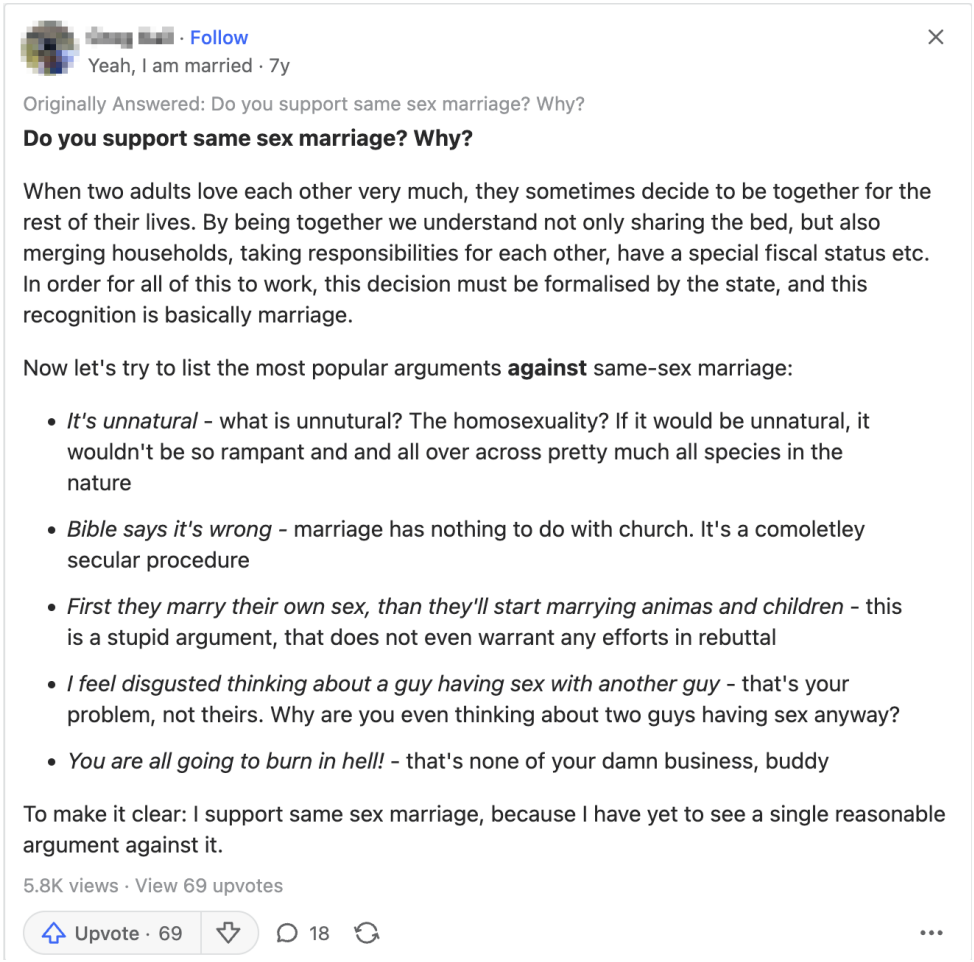


Figure 1.1: An example answer on Quora to the question "Do you support same sex marriage? Why?"

data, which not only identifies the argumentative text segments in the text but also the relations between them (Prakken and Vreeswijk, 2002, Lawrence and Reed, 2020). By convention, argumentative text segments are called "argument components" and relations between them are called "argument relations". An example of such a conversion is shown in Figure 1.2, where nodes correspond to argument components and edges to two kinds of argument relations, namely attack and support.

Argument mining has a wide range of applications in various fields, including politics, law, education, and business. In politics, argument mining can be used to analyse political speeches and debates to gain insights into the arguments made by political candidates (Menini et al., 2017). In law, argument mining can assist in legal summarisation with the automatic analysis of legal documents (Yamada et al., 2019a,b). In education, argument mining can be used to automatically classify student utterances in classroom discussions (Lugini and Litman, 2018). In business, argument mining can be used to analyse customer feedback and reviews to identify customer preferences and complaints (Park, 2016).

Argument mining can also be a valuable tool in the context of online discussions,

I think [it's not necessary for universities to provide laptops for all students]$_1$. [Not every student needs a laptop]$_2$, as [they can simply use the computers in the library for studying]$_3$. Also, [the "provided" laptops aren't really free — they are likely covered by the tuition fees paid by students themselves]$_4$. Some may argue that [universities are wealthy enough to pay using their own funds]$_5$, but I doubt it since [they can hardly offer a fair salary to their teachers]$_6$. And even if it's true, [these money should be used on something more important first]$_7$.

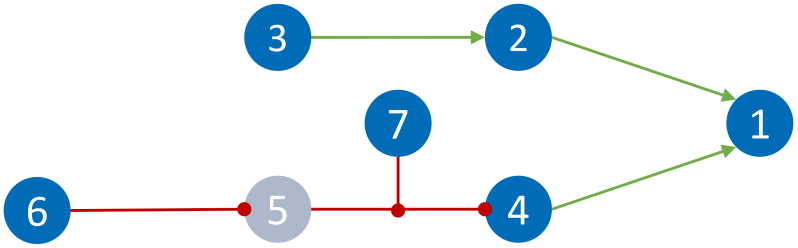


Figure 1.2: An example argument and its structure, using the annotation scheme in Peldszus and Stede (2015a). Nodes represent two different types of argument components (blue and grey). Edges represent two kinds of argument relations (arrow-head and circle-head.

allowing for the identification and analysis of arguments from diverse perspectives. Consider an online discussion forum where users are discussing a controversial topic, such as climate change. Opinions on climate change can range widely, from denial of its existence to a belief in its extreme urgency. The number of user-generated posts can be overwhelming, which makes manual analysis of the arguments in such a discussion time-consuming and error-prone. Argument mining can greatly simplify this task by automating the identification of opinions expressed by each participant, as well as their underlying reasoning. In this way, it can provide valuable insights into the underlying reasons and motivations for opinions and beliefs, leading to the discovery of common ground and the facilitation of mutual understanding.

1.1 Motivation

Many existing studies employ pipelined approaches to argument mining, first training separate models for components and relations, and subsequently reconstructing the final argument structure based on rules or model ensemble methods (Persing and Ng, 2016, Stab and Gurevych, 2017, Mirko et al., 2020, Dutta et al., 2022). As with many other pipelined approaches, these methods are susceptible to error propagation. Additionally, many of these models heavily rely on feature engineering, such as those by Moens et al. (2007), Palau and Moens (2009), and Park (2016), which requires considerable manual effort. Such approaches are also limited in terms of flexibility and robustness in cross-domain scenarios.

In contrast to arguments found in formal text such as student essays and legal docu-

ments, arguments in online discussions generally exhibit implicit language use and flexible structures (Boltužić and Šnajder, 2014). Informal argumentation does not mean that people argue badly. The informal arguments I have observed are in general logical and reasonable, but the presentation of the argument is so informal that the logic might not be immediately obvious. There is irregular syntax, incomplete sentences, and irony. Personal and emotional statements are often mixed together with more objective statements. This makes the automated analysis of such informal arguments a particularly challenging task. Efforts have been made to apply argument mining to online discussions. However, most of these attempts either only involve simple and short texts like tweets (Schaefer and Stede, 2020, Iskender et al., 2021) and customer reviews (Wachsmuth et al., 2014, Park, 2016), which are unlikely to contain structured arguments, or they focus solely on stance detection (Sridhar et al., 2015) or component identification (Habernal and Gurevych, 2017) without addressing the relations within argument structures.

The informal nature of arguments in online discussions also makes it difficult to develop annotation schemes. Annotation schemes define the structured output of argument mining. The design of the annotation scheme is a crucial factor in any argument mining datasets. While simple claim-premise based annotation schemes may fall short in capturing key attributes of informal arguments, excessively complex schemes are often less flexible and may exhibit low inter-annotator agreement. As a result, it is a considerable challenge to create an annotation scheme that balances the informativeness and reliability of annotation on informal arguments (Habernal and Gurevych, 2017). Moreover, the form of arguments in online discussions varies considerably depending on the platforms used, the topics discussed, and the individual users participating (Spatariu et al., 2007, Ruiz et al., 2011, Coe et al., 2014, Muddiman and Stroud, 2017), which adds to the complexity of the task.

In addition to being less formal, arguments in online discussions present further challenges due to the diversity of topics, the variety of linguistic expressions, and the implicit assumptions often embedded within them. Accurate interpretation of such arguments often requires a comprehensive grasp of external knowledge (Saint Dizier, 2016, Moens, 2018). Several strategies for integrating external knowledge have been explored in argument mining research, such as Lv et al. (2020), Dutta et al. (2022), and Rodrigues and Branco (2022). However, these studies often focus on specific subtasks and use distinct techniques tailored for individual model architectures. Such specificity complicates the application of these techniques. In some cases, it may even make it impossible to use them at all in end-to-end argument mining approaches.

Looking at the big picture, in this thesis I propose to apply argument mining to informal text sourced from online discussions, addressing the challenges described above. The main motivation is to help people with different perspectives better understand each other. Not being able to understand others' opinions or reasoning can often lead to

discrimination, prejudice, and stereotyping (Fiske, 1998, Fishbein, 2014). Ultimately, I aspire to contribute to the promotion of constructive dialogue and mutual understanding among individuals with varied viewpoints, potentially mitigating discrimination, prejudice, and stereotyping. By successfully applying argument mining to informal text in online discussions, we can foster a more inclusive and well-informed discourse, enabling individuals to better comprehend and appreciate the diverse opinions that shape our society.

1.2 Thesis outline

In Chapter 2, I provide the relevant background information necessary to understand the thesis. I first introduce the definitions of argumentation and argument, along with argumentation models and argumentation schemes as frameworks to analyse arguments. I also introduce the task of argument mining and its subtasks.

In Chapter 3, I provide an overview of existing research in the field of argument mining, in terms of annotation schemes, technical approaches, and external knowledge integration methods.

In Chapter 4, I present a simple, concise, and informative annotation scheme for informal arguments. The resultant dataset consists of 400 argumentative texts collected from Quora. I also conduct a human agreement study and provide an analysis of the results.

In Chapter 5, I propose an end-to-end approach to argument mining by formulating it as a dependency parsing task. My dependency representations and dependency parsers are proved to be effective for formal arguments as well as informal arguments. I conclude that argument mining as dependency parsing is a promising approach for both formal and informal arguments.

In Chapter 6, I present a comprehensive study on external knowledge integration for end-to-end argument mining. I experiment with various knowledge sources and techniques, and analyse their effects on formal and informal arguments.

In Chapter 7, I conclude the thesis by summarising the principal contributions and by suggesting potential directions for future research.

Chapter 2

Background

In this chapter I will establish the necessary background knowledge that contextualises and supports the subsequent chapters. The term "argumentation" and "argument" are often used interchangeably in the field of argument mining, but it is still necessary to distinguish them. Following the convention in Besnard and Hunter (2008), argumentation is defined in this thesis as the process of constructing and evaluating arguments, and arguments as the product of such a process. I first introduce the definitions of argumentation and argument in Section 2.1, from various perspectives including theoretical linguistics, computational linguistics and communication theories. The divergent foci in these definitions give rise to two crucial categories of frameworks for analysing arguments: those that fall under the umbrella of argumentation models, and those that focus on argumentation schemes. Argumentation models (Section 2.1.2) describe arguments in a structural point of view, while argumentation schemes (Section 2.1.3) emphasise the persuasion patterns in arguments.

In Section 2.2, I present an introduction to argument mining, a task within the field of natural language processing (NLP) that focuses on the automated extraction and analysis of argumentative structures from text. Following this, I discuss the two fundamental subtasks of argument mining, namely component identification (Section 2.2.1) and relation identification (Section 2.2.2).

2.1 Argumentation and argument

Argumentation is a multifaceted concept that has been defined and formalised from various perspectives in different fields. One of the earliest and most influential studies on argumentation was provided by Aristotle in ancient Greece. In his treatise *On Rhetoric* (translated by Kennedy, 2006), Aristotle defines argumentation as the art of communication and philosophical disputation. According to Aristotle, an inductive argument contains a particular conclusion drawn from one or more examples, while a deductive argument

consists of a series of premises that lead to a conclusion through logical deduction. In deductive arguments, the premises are statements that are accepted as true or likely to be true, while the conclusion is the statement that follows logically from the premises. Aristotle identifies three modes of persuasion that can be used in argumentation: logos, pathos, and ethos. Logos refers to the use of reasoning and logical arguments to persuade an audience, pathos refers to the use of emotions and feelings to appeal to the audience's sympathies and values, and ethos refers to the speaker's personal character, credibility, and trustworthiness. At its core, the Aristotelian theory of argumentation is concerned with the analysis and evaluation of arguments, with the aim of distinguishing between good and bad arguments.

2.1.1 Definitions

Regarding their different emphases on logos, pathos, or ethos, I categorise definitions of argumentation and argument into logos-centred or audience-centred. The logos-centred approach focuses on the logical structure and content of arguments (logos), while the audience-centred approach combines pathos and ethos, emphasising the importance of adapting arguments to the audience's values, beliefs, and emotions.

2.1.1.1 Logos-centred

One of the most fundamental frameworks under the logos-centred approach is Toulmin's model of argumentation (Toulmin, 1958). Although Toulmin does not present an abstract definition of argumentation or argument, his model provides a tool to understand the logical structure of arguments. Toulmin's model deconstructs an argument into six interconnected components, each serving a unique functional role in the reasoning process. This argument layout has become the basis for many formal approaches to argumentation. Details of this model will be introduced in Section 2.1.2.2.

Freeman (1991, 2011) defines argumentation by integrating Toulmin's model into argument diagramming techniques. In his theory of the macro-structure of argumentation, argumentation is modelled as a hypothetical dialectical exchange between a "proponent" and an "opponent". Each move in the exchange corresponding to a structural element in the argument graph.

Two other influential logos-centred definitions of argument come from Pollock (Pollock, 1987, 1992), and Besnard and Hunter (Besnard and Hunter, 2001, 2008, 2009). According to Pollock, an argument is defined as a finite sequence of propositions, where each proposition is either an epistemic basis or a result of the reasoning process of a set of earlier propositions[1]. The recursive nature of this definition makes it more flexible than

[1]This is Pollock's definition of linear arguments. More complex arguments are called indirect arguments,

Toulmin's model. Besnard and Hunter define an argument as a pair $\langle \Phi, \alpha \rangle$, where α is a claim supported by the support Φ, which is a minimal consistent subset of formulae in a database. These two definitions are similar in that they both use formal logic to conceptualise arguments, and the structure of arguments in the two definitions are almost equivalent. Details of these two definitions will be introduced in Section 2.1.2.3.

All definitions discussed above place emphasis on the structural aspect of arguments. Such an emphasis has led to the development of a category of frameworks known as argumentation models (Section 2.1.2), which is an important theoretical tool for analysing arguments.

2.1.1.2 Audience-centred

While logos-centred definitions focus on the logical structure and content of arguments, audience-centred definitions prioritise the communicative process and the persuasive effects on the audience. Despite the lack of consensus on the definitions themselves, many audience-centred definitions are based on several key elements of argumentation, including the context in which the argumentation occurs, the agents involved in the argumentation, the objectives of the argumentation, and the strategies utilised to achieve those objectives.

The definition of argumentation provided by Van Eemeren and Grootendorst (1984, 2004) addresses the communicative aspect of argumentation. They characterise argument as a dialogical exchange between a "speaker" and a "hearer". Argumentation is defined by them as "a verbal, social, and rational activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint". This activity is further seen as occurring in the context of communication, being directed towards an audience, and being intended to persuade the audience using intellectual reasoning and emotional appeals. Definitions of argumentation with similar emphases can also be found in Freeman (1991), Walton (1998), O'Keefe (2012), Freeley and Steinberg (2013), and Zarefsky (2019).

One of the key elements in audience-centred definitions is the strategies used in argumentation. Patterns of argument strategies are called "argumentation schemes", which can be applied to different situations to construct an argument (Perelman, 1971). They provide a framework for identifying the types of reasoning strategies used in an argument, which can be used to evaluate the quality of the argument based on its adherence to the underlying scheme (Nussbaum, 2011). Various studies on the classification of argumentation schemes (Hastings, 1962, Kienpointner, 1992, Grennan, 1997, Prakken, 2010) have been conducted, but the most comprehensive and systematic one is the one by Walton et al. (2008). Walton et al.'s argumentation schemes include logical reasoning, but also arguing

which allow the adoption of unestablished premises as suppositions. Considering the concern of this thesis, I only introduce Pollock's definition of linear arguments.

by appealing to emotions and by establishing speaker's credibility. Some of their schemes will be introduced in Section 2.1.3.

2.1.2 Argumentation models

Argumentation models aim to provide a structural representation of arguments, enabling a systematic analysis of their components and relations, and to capture the relationships between arguments. These models abstract away from the surface form of an argument to a conceptual level, focusing on the interconnections between the different elements of an argument and how they relate to each other (Prakken and Vreeswijk, 2002).

According to the taxonomy presented in Bentahar et al. (2010), argumentation models can be classified into three categories: monological models, which are designed to represent the internal structure of a single argument; dialogical models, which aim to capture the relationships between arguments in entire debates or dialogues; and rhetorical models, which are used to describe persuasion patterns in arguments with a special focus on the audience's perception. It should be noted that rhetorical models in this taxonomy is in fact closer to the notion of argumentation schemes rather than argumentation models, as they emphasise persuasion patterns and the audience's perception and ignore the structural aspect of arguments.

In this section, I will only introduce influential monological argumentation models, as the research target of this thesis is limited to monological arguments.

2.1.2.1 Walton's model

Walton (1996) defines several basic argument structures from a pragmatic perspective, as illustrated in Figure 2.1. He refers to the representations of argument structures as argument diagrams. An argument diagram illustrates the inference process of an argument.

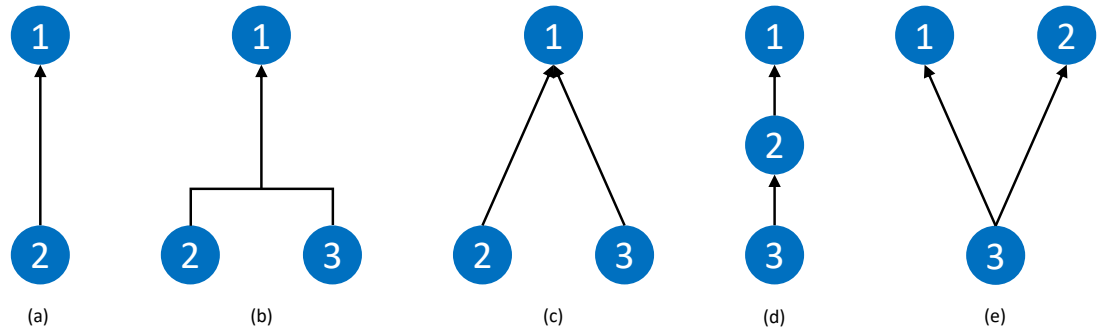


Figure 2.1: Basic argument structures in Walton's model: (a) single argument, (b) linked argument, (c) convergent argument, (d) serial argument, and (e) divergent argument.

It consists of nodes symbolising argument components and edges connecting these nodes. Since Walton's argument diagrams are about argument structures, they can be seen as a type of argumentation model.

Walton lists five basic structures of arguments as follows:

- **Single argument** (Figure 2.1a), in which there is only one premise and one conclusion. This is the simplest form of argument where a single proposition supports another.

- **Linked argument** (Figure 2.1b), in which multiple premises jointly support a single conclusion. All the premises are necessary for the conclusion to be valid. Example 2.1 shows a linked argument: in order to compare the population density of The United States and Canada, it is necessary to know both their total area and population.

  Example 2.1
  Component 1: The population density of The United States is much higher than that of Canada.
  Component 2: The total area of The United States and Canada is similar.
  Component 3: The United States has a population about 7 times larger than that of Canada.

- **Convergent argument** (Figure 2.1c), in which multiple independent premises separately support the same conclusion. Even if one premise is removed, the conclusion can still be valid based on the remaining premises. Example 2.2 shows a convergent argument: Carlos has two good properties, each of which on their own is already sufficient to make him excel in his studies.

  Example 2.2
  Component 1: Carlos will excel in his studies.
  Component 2: Carlos is a dedicated student.
  Component 3: Carlos is smart.

- **Serial argument** (Figure 2.1d), in which a premise supports a conclusion, and that conclusion, in turn, serves as a premise for another conclusion. This creates a chain-like structure of reasoning. Example 2.3 shows a serial argument: tourism should be stopped because it is a threat to nature; tourism is a threat to nature because it has killed much marine life.

  Example 2.3
  Component 1: Tourism should be stopped.

Component 2: Tourism is a threat to nature.
Component 3: Tourism has killed much marine life.

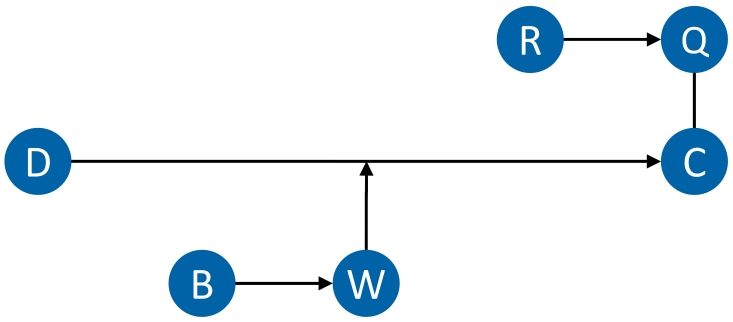
- **Divergent argument** (Figure 2.1e), in which a single premise supports multiple conclusions. Example 2.4 shows a divergent argument: holding a degree in computer science, Sean can be familiar with algorithms and data structures, and he can also be bad at thinking from human-centric perspectives.

Example 2.4
Component 1: Sean always misses out on human-centric perspectives.
Component 2: Sean is familiar with algorithms and data structures.
Component 3: Sean has a degree in computer science.

2.1.2.2 Toulmin's model

Toulmin's argumentation model (Toulmin, 1958) is based on the perspective that argumentation is a reasoned and logical attempt to convince someone of the acceptability of a particular claim.

Toulmin's model is usually illustrated as a connected set of nodes, as depicted in Figure 2.2. It deconstructs an argument into six interconnected components, each serving a unique functional role in the reasoning process. The six components of Toulmin's model are as follows:



[Harry was born in Bermuda]$_{\textbf{Data}}$. *Since* [a man born in Bermuda will generally be a British subject]$_{\textbf{Warrant}}$. *On account of* [the following legal provisions: ...]$_{\textbf{Backing}}$ *So*, [presumably]$_{\textbf{Qualifier}}$, [Harry is a British subject]$_{\textbf{Claim}}$. *Unless* [both his parents were aliens/he has become a naturalised American/ ...]$_{\textbf{Rebuttal}}$

Figure 2.2: A graphical representation of Toulmin's model, along with an example argument taken from Toulmin (1958). Nodes represent argument components: C = Claim, D = Data, W = Warrant, B = Backing, Q = Qualifier, R = Rebuttal.

1. **Claim**: the central proposition or statement that the argument is trying to establish. It is what the argument is ultimately arguing for.

2. **Data**: the evidence or information that supports the claim. The data can be factual, statistical, anecdotal, or any other kind of information that is relevant to the claim.

3. **Warrant**: the justification or reasoning that connects the data to the claim. It is the underlying principle or rule that makes the leap from the data to the claim. The warrant can be explicit or implicit, and it can be based on a variety of sources such as logic, experience, authority, or common sense.

4. **Backing**: the additional support or justification for the warrant. It provides further evidence or reasoning to support the warrant and to help strengthen the argument.

5. **Qualifier**: the acknowledgment of the limitations or scope of the argument. It sets the parameters for accepting the argument by indicating the strength or certainty of the claim.

6. **Rebuttal**: the acknowledgment of counterarguments to the claim by providing conditions or situations supporting the qualifier. It addresses potential weaknesses or objections to the argument in order to strengthen the argument, by demonstrating that it has been carefully and objectively analysed rather than reflecting a biased or one-sided perspective.

While this model does not define the relationships between its components, it can be inferred that all relationships are supportive, except for the connection between the Qualifier and the Claim, and the connection between the Rebuttal and the "qualified claim" that it attacks.

Toulmin's model has been widely used in fields such as philosophy, rhetoric, communication, and education (Verheij and Hitchcock, 2006, Tindale, 2007, Van Eemeren et al., 2013, Bailin and Battersby, 2016). However, it also has some limitations: its components are defined informally, and it is not flexible with respect to how such argument structures can be combined to represent complex arguments (Bentahar et al., 2010); the distinction between Data and Warrant is not sustainable in practice (Habernal and Gurevych, 2017).

2.1.2.3 Pollock's model, and Besnard and Hunter's model

In contrast to the informal and ambiguous nature of Toulmin's model, Pollock's model and Besnard and Hunter's model provide a formal approach to argumentation by utilising formal logic.

Pollock's argumentation model formalises defeasible reasoning, which is a form of reasoning that is tentative and subject to revision in the face of new information (Pollock, 1987, 1992). While the initial intention of this model was to address certain aspects of defeasible reasoning, its underlying logic remains grounded in classical logic. Therefore, Pollock's model is a foundational framework for logic-based argumentation.

In Pollock's model, an argument is defined as a finite sequence of propositions, where each proposition is either an epistemic basis or a result of the reasoning process of a set of earlier propositions. Pollock breaks down an argument into smaller structures, which he calls "lines of argument". A "line of argument" is an ordered triple $\langle P, R, \{m, n, \dots\}\rangle$ consisting of a proposition P that is supported by the line, a rule of inference R, and a set of indices $\{m, n, \dots\}$ that identify the previous lines that support the present line through R.

A visualisation of this model can be found in Figure 2.3. The argument in this figure can be represented as a sequence of propositions: $\{P_1, P_2, P_3, P_4\}$. Lines of argument are illustrated as dashed boxes in this figure. For example, the line from P_2 to P_4 can be written as $L_4 = \{P_4, R_2, \{L_2\}\}$. L_4 is the index of this line of argument, P_4 is the proposition supported by this line, R_2 is the inference rule used to reach P_4, and L_2 is the previous line, which supports the present line L_4 through R_2.

Unlike Pollock's model, Besnard and Hunter's argumentation model is based on deductive reasoning. In deductive reasoning, a conclusion is necessarily derived from its premises, meaning that if the premises are true, the conclusion must also be true (Besnard and Hunter, 2001, 2008, 2009). An argument in this model is defined as a pair $\langle \Phi, \alpha\rangle$ consisting of a claim α and its support Φ, where Φ is a minimal consistent subset of formulae in a database Δ. The formulae in Δ can represent certain or uncertain information, and can

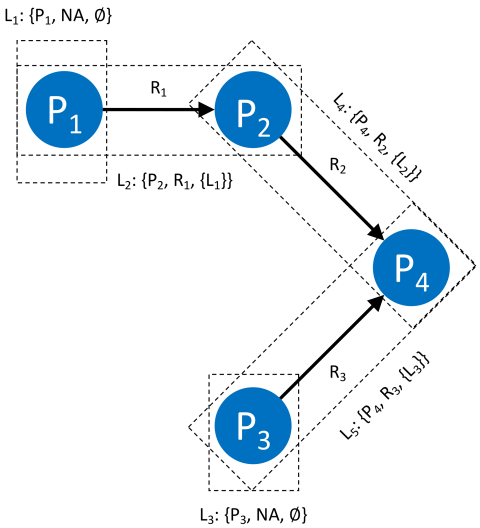


Figure 2.3: A graphical representation of Pollock's model. Nodes are propositions. Edges are inference rules. Items in dashed boxes are lines of argument.

denote objective, subjective, or hypothetical statements. Such formulae can contain logical operations including implication ($\rightarrow$), conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). The ordering of formulae in Φ is arbitrary and does not affect the argument.

The primary difference between Besnard and Hunter's model and Pollock's model lies in the nature of their inference rules. In defeasible reasoning, inference rules can be challenged given new information. In contrast, in deductive reasoning, inference rules are fixed and must always be upheld regardless of new evidence. This difference leads to two different ways of incorporating inference rules into argumentation models. Pollock's model explicitly lists the rule of inference as an element (*i.e.* R) in the triple of a line, whereas Besnard and Hunter's model embeds it into the support (*i.e.* Φ) of an argument. This is because for deductive reasoning, the rule of inference is always the logical operation implication (*i.e.* $\rightarrow$).

Nonetheless, in other aspects, Besnard and Hunter's model and Pollock's model are equivalent. In fact, the last proposition P in Pollock's model is equivalent to the claim α in Besnard and Hunter's model, and compositions of formulae in the support Φ in Besnard and Hunter's model can produce all the lines in Pollock's model. For example, using Besnard and Hunter's model, the argument in Figure 2.3 can be written as $\langle\{P_1, P_3, P_1 \rightarrow P_2, P_2 \rightarrow P_4, P_3 \rightarrow P_4\}, P_4\rangle$.

2.1.2.4 Counterargument and undercut

In argumentation, counterarguments play a critical role in challenging the soundness or validity of an argument. They provide an opportunity to anticipate and address potential objections or alternative viewpoints, making the argument more robust and convincing. In this section, I will discuss the concept of counterarguments in detail, including their types and structures.

In Toulmin's model, counterarguments are integrated into the argumentation model itself as rebuttals. A rebuttal attempts to demonstrate the weakness or invalidity of an opponent's argument by providing conditions or situations where the claim may not stand. A strong rebuttal acknowledges and addresses the reservations that the arguer or their audience might have about the argument, and is usually followed by a response to the rebuttal that demonstrates why the argument remains strong despite those reservations.

Unlike Toulmin's model, counterarguments are defined as separate entities that attack the main argument in Pollock's model and Besnard and Hunter's model. In Pollock's model, counterarguments are referred to as "defeaters". This is because in this model, counterarguments attack "prima facie reasons" (*i.e.* defeasible reasons), which are reasons that can be overturned by further information or evidence (Pollock, 1987). Pollock explains defeasibility as follows:

Example 2.5

"An observation like 'X appears red to me' may initially provide a reason to believe that X is indeed red. However, this reason is defeasible in the sense that it can be undermined by other evidence or information. For instance, if someone with a reputation for reliability informs me that X is not actually red but only appears so due to unusual lighting conditions, then the combination of these two pieces of information would no longer support my belief that X is red. " (Pollock, 1987)

Counterarguments (*i.e.* defeaters) in Pollock's model are therefore reasons that can defeat such prima facie reasons. A defeater must not conflict with the original defeasible reason. For example, "X appears black to me" conflict with "X appears red to me", so that the former is not a defeater of the latter. There are two subcategories of defeaters, namely rebutting defeaters and undercutting defeaters, which are defined as follows (Pollock, 1987):

Definition 2.1 R is a rebutting defeater for P as a prima facie reason for Q iff R is a defeater and R is a reason for believing $\neg Q$.

Definition 2.2 R is an undercutting defeater for P as a prima facie reason for Q iff R is a defeater and R is a reason for denying that P wouldn't be true unless Q were true.

The difference between a rebutting defeater and an undercutting defeater therefore lies in the way they defeat a defeasible reason. A rebutting defeater directly attacks the conclusion by providing reasons to believe the opposite of it, while an undercutting defeater challenges the relation (*i.e.* the inference rule) between the reason and the conclusion. Consider the following two examples:

Example 2.6

Reason: Carlos turned in all assignments on time.
Conclusion: Carlos should get an A in this course.
Rebutting defeater: The instructor discovered that Carlos had plagiarised in his final paper.

Example 2.7

Reason: Carlos turned in all assignments on time.
Conclusion: Carlos should get an A in this course.
Undercutting defeater: The quality of Carlos' assignments was poor.

The rebutting defeater in Example 2.6 directly attacks the conclusion by showing that Carlos did not meet the academic integrity requirements, and therefore does not deserve an A. In contrast, the undercutting defeater in Example 2.7 suggests that despite Carlos's timeliness, the poor quality of his assignments could undermine the initial reasoning process.

Counterarguments defined by Besnard and Hunter (2009) also have two subcategories, namely rebuttals and undercuts:

> **Definition 2.3** An argument $\langle \Psi, \beta \rangle$ is a rebuttal for an argument $\langle \Phi, \alpha \rangle$ iff $\beta \leftrightarrow \neg\alpha$ is a tautology.

> **Definition 2.4** An undercut for an argument $\langle \Phi, \alpha \rangle$ is an argument $\langle \Psi, \neg(\phi_1 \wedge \cdots \wedge \phi_n) \rangle$ where $\phi_1, \ldots, \phi_n \subseteq \Phi$.

We can see that a counterargument in Besnard and Hunter's model is an argument consisting of a support and a claim, while a counterargument in Pollock's model is only a reason and does not include claims. When we overlook this distinction, a rebuttal as defined by Besnard and Hunter is equivalent to that by Pollock, with both being an attack on the conclusion. However, their definitions of undercuts are different, in that Pollock defines undercuts as an attack on the relation between the support and the conclusion, while Besnard and Hunter define it as an attack on the support. Since inference rules are incorporated into the support in Besnard and Hunter's model, undercuts as defined by Besnard and Hunter include those by Pollock. This inclusiveness can be illustrated by the following example:

> **Example 2.8**
> *Support*: The new medication has been approved by the FDA.
> *Conclusion*: The new medication is safe.
> *Counterargument*$_1$: The FDA approval was based on limited clinical trials which might overlook unexpected side effects.
> *Counterargument*$_2$: That the new medication has been approved by the FDA is fake news.

In this example, the first counterargument attacks the relation between the support and the conclusion, and the second counterargument attacks the support. As a result, they are both regarded as undercuts according to Besnard and Hunter's definition, but only the first is considered as an undercut by Pollock's definition.

In this thesis, I aim to distinguish the attacks on relations from those on supports or conclusions. Therefore, throughout the rest of this thesis, I will adhere to Pollock's definition in terms of the concept of undercuts.

2.1.3 Argumentation schemes

In addition to proposing structural models of argumentation, another approach to analysing arguments is through the use of argumentation schemes.

Walton et al. (2008) propose a set of argumentation schemes which are particularly frequent in informal arguments, such as "argument from sign", "argument from example", and "argument from consequences". Each argumentation scheme in Walton et al.'s framework is represented as an abstract description of the general inference process involved in that type of argumentation. For instance, "argument from consequences" is a form of reasoning where the truth of a proposition is argued based on the perceived consequences that would result from its acceptance or rejection (*e.g.* "If we do not water this plant, it will die. Therefore, we should water the plant."). The inference process for this scheme is described as follows:

> If A is brought about, then good (bad) consequences will (may plausibly) occur.
>
> Therefore, A should (not) be brought about.

In this framework, an argumentation scheme is always accompanied by a set of critical questions, which serve as a tool to evaluate the strength and soundness of an argument. Critical questions aim to test the validity and coherence of an argument and identify any weaknesses or gaps in the reasoning (Hastings, 1962). For example, critical questions for "argument from consequences" are (Walton et al., 2008):

1. How strong is the likelihood that these cited consequences will (may, must, *etc.*) occur?
2. If A is brought about, will (or might) these consequences occur, and what evidence supports this claim?
3. Are there other consequences of the opposite value that should be taken into account?

There are typically two types of critical questions in this framework: presumptive and conclusive. Presumptive critical questions are used to evaluate the plausibility of the premises of an argument, while conclusive critical questions are used to evaluate the validity of the inference from the premises to the conclusion. By asking critical questions, interlocutors in an argument can challenge the arguments put forward by the other party and encourage them to provide further evidence or reasoning to support their claims.

In what follows, I will introduce some of the argumentation schemes by Walton et al. (2008) that are closely related to the design of the annotation scheme proposed in this thesis.

2.1.3.1 Argument from position to know

The argumentation scheme "argument from position to know" involves using a person's position or expertise to support a claim or conclusion they put forward. In this type of argumentation, the opinion or assertion made by the interlocutor with presumed access to information is given greater weight and is considered more credible than if it were made by someone without such access. This is because the interlocutor's presumed privileged access to information is seen as providing additional support for the claim being made. For example, a company's CFO may be more trusted to provide accurate information about the company's financial performance than an external financial analyst, as the CFO has privileged access to internal financial data. The inference process for this scheme and its accompanying critical questions are listed in Table 2.1.

"Argument from position to know" is a type of argumentation scheme that is frequently used in everyday argumentation. While it can be effective in shifting the responsibility between interlocutors to prove a claim, it is also subject to weaknesses such as the credibility of the source, the potential for misquotation or misinterpretation, and the lack of transparency in identifying sources, making it vulnerable to potential biases and inaccuracies. Careful evaluation of the source and the exact wording of the assertion is crucial in assessing the value and reliability of this type of argumentation.

2.1.3.2 Argument from expert opinion

As a special type of "argument from position to know", "argument from expert opinion" relies on the presumed knowledge or expertise of an individual in a particular domain to support a claim or conclusion they put forward. In this type of argumentation, the opinion or assertion made by an expert is given greater weight and is considered more credible than if it were made by someone without the same level of expertise. For example, a doctor's opinion on a medical issue may carry more weight than that of a non-expert. The inference process for this scheme and its accompanying critical questions are listed in Table 2.2.

Argument from position to know	
Inference process	a is in a position to know whether A is true. a asserts that A is true. Therefore, A is true.
Critical questions	Is a in a position to know whether A is true? Is a an honest (trustworthy, reliable) source? Did a assert that A is true?

Table 2.1: The inference process for "argument from position to know", and its accompanying critical questions (Walton et al., 2008).

Argument from expert opinion	
Inference process	E is an expert in domain D. E asserts that A is known to be true. A is within D. Therefore, A may (plausibly) be taken to be true.
Critical questions	Is E a genuine expert in D? Did E really assert A? Is A relevant to domain D? Is A consistent with what other experts in D say? Is A consistent with known evidence in D?

Table 2.2: The inference process for "argument from expert opinion", and its accompanying critical questions (Walton et al., 2008).

"Argument from expert opinion" is a powerful type of argumentation that can be particularly useful in discussions on specialised topics where expert knowledge is needed. Besides the problems in argument from position to know, this argumentation scheme also suffers from the potential for conflicting opinions among experts, and the inappropriate use of expert opinions in an unrelated domain. Therefore, it is important to evaluate the credibility of the expert, and whether the assertion is relevant to the domain at hand.

2.1.3.3 Argument from example

The argumentation scheme "argument from example" involves using one or more examples to support a generalisation or claim. The examples are usually presented as specific instances of the generalisation or claim, and are meant to illustrate the claim or provide evidence in favour of it. For instance, in order to support the opinion that dropping out of college is not detrimental to success, it is common to use examples of college dropout founders of successful companies such as Apple and Microsoft. The inference process for this scheme and its accompanying critical questions are listed in Table 2.3.

"Argument from example" is a common form of argumentation used in everyday discourse, and is often used to support claims in fields such as science, law, and politics. However, the strength of the argument depends on the quality and representativeness of the examples provided, the nature of the generalisation being supported, whether there are any counter-examples to the generalisation, and whether the number of examples is sufficient to support the generalisation. A single example may be sufficient to support a plausible or defeasible generalisation, but would be insufficient to support a strict or universal generalisation.

Argument from example	
Inference process	In this particular case, the individual a has property F and also property G. a is typical of things that have F and may or may not also have G. Therefore, generally, if x has property F, then x also has property G.
Critical questions	Is the proposition presented by the example in fact true? Does the example support the general claim it is supposed to be an instance of? Is the example typical of the kinds of cases that the generalisation ranges over? How strong is the generalisation? Were there special circumstances present in the example that would impair its generalisability?

Table 2.3: The inference process for "argument from example", and its accompanying critical questions (Walton et al., 2008).

2.1.3.4 Argument from analogy

"Argument from analogy" is another common type of argumentation scheme, in which an argument is made by drawing a comparison between two cases that are said to be similar in a certain respect. For instance, when a baseball coach is teaching a golfer, they may instruct the golfer to swing a bat with a similar technique to a golf swing. The inference process for this scheme and its accompanying critical questions are listed in Table 2.4.

"Argument from analogy" can be a powerful way to make an argument when the target subject is hard to explain or unfamiliar to the audience, but it can be flawed if the differences between the two cases are overlooked. Critical evaluation of the similarities and differences between the two cases and the exact nature of the analogy being made is crucial in assessing the strength of this type of argumentation.

Argument from analogy	
Inference process	Generally, case C_1 is similar to case C_2. A is true in case C_1. Therefore, A is true in case C_2.
Critical questions	Are C_1 and C_2 similar, in the respect cited? Is A true in C_1? Are there differences between C_1 and C_2 that would tend to undermine the force of the similarity cited? Is there some other case C_3 that is also similar to C_1, but in which A is false?

Table 2.4: The inference process for "argument from analogy", and its accompanying critical questions (Walton et al., 2008).

2.1.4 Argument interchange format

Both argumentation models and schemes provide theoretical frameworks for analysing arguments. However, the heterogeneity among various argumentation models and schemes makes it challenging to establish a standardized or universally applicable method for argument analysis. In order to address the challenges posed by such heterogeneity, Chesnevar et al. develop the argument interchange format (AIF). AIF uses an abstract model with multiple possible "reifications" (*i.e.* concrete realisations), providing a standardised framework for representing arguments among various argumentation tools and applications.

The AIF ontology consists of three main aspects:

- Arguments and argument networks. This involves defining the core ontology for argument components and their relationships, essential for AIF reification. It assumes that argument components and their relations can be represented as nodes in a directed graph, forming an argument network (AN).

- Communication. This group focuses on elements related to the interchange of arguments, including locutions (*i.e.* individual words or expressions) and interaction protocols.

- Context. This includes elements associated with the environments in which argumentation takes place, such as participants in argument exchanges and theories used for argumentation.

In terms of monological arguments, the notion of AN in the AIF is especially useful, as ANs are capable of representing monological arguments annotated with a wide range of existing annotation schemes. An AN is a directed graph consisting of nodes and edges, as illustrated in Figure 2.4. There are two kinds of node in ANs, namely information nodes (I-nodes) and scheme nodes (S-nodes). I-nodes represent textual content in an argument, and S-nodes are applications of annotation schemes. S-nodes can be further categorised into rule of inference application nodes (RA-nodes), conflict application nodes (CA-nodes), and preference application nodes (PA-nodes). RA-nodes are close to the "support relations" we have seen in most argumentation models, and CA-nodes are similar to the "attacking relations". Edges in ANs usually are not explicitly labelled or supplied with semantic pointers, because their types can be easily inferred from the types of nodes they connect.

With those abstract node types, besides usual argument structures including single, linked, convergent, serial and divergent argument, ANs can also represent undercuts as distinct from rebuts. For instance, in Figure 2.4, I-node 12 undercuts the relation from I-node3 to I-node1.

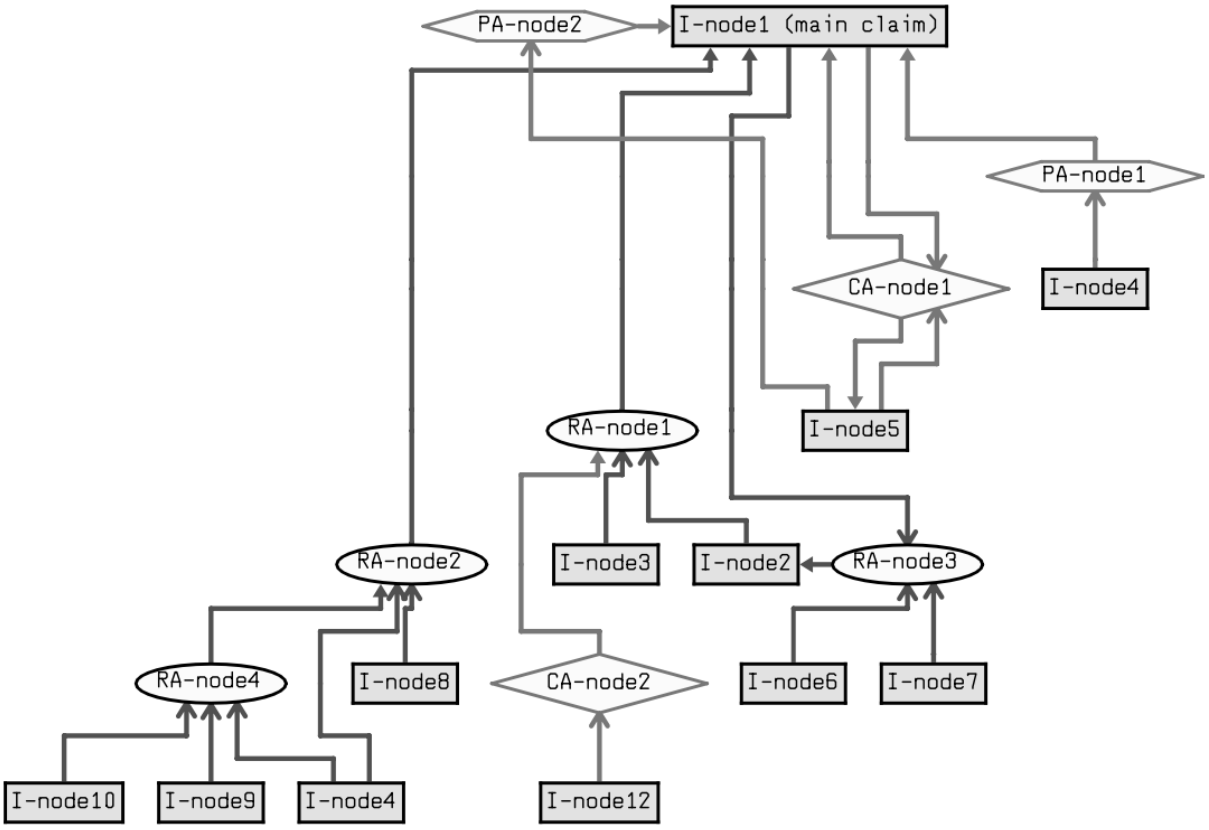


Figure 2.4: An example argument network. Reproduced from Chesnevar et al. (2006).

The AIF has been used as a basis of many annotation schemes, such as those in the studies by Peldszus and Stede (2015a), Lawrence and Reed (2017), Visser et al. (2020), Hautli-Janisz et al. (2022). It is also the basis of the annotation scheme that I will propose in this thesis.

2.2 Argument mining

Based on the frameworks of argumentation models and schemes, argument mining aims to analyse arguments automatically (Peldszus and Stede, 2013, Green et al., 2014). It typically includes three subtasks (Persing and Ng, 2016, Eger et al., 2017, Habernal and Gurevych, 2017, Stab and Gurevych, 2017, Lawrence and Reed, 2020):

1. **Segmentation**: Cutting a raw sequence into minimal units of analysis that are either argumentative or non-argumentative (only argumentative segments are called argument components). Argument components usually cover clauses or proposition-like units of text.

2. **Component classification**: Labelling each argument component with a tag in a pre-defined annotation scheme (*e.g.* Premise or Claim).

3. **Relation identification**: Deciding if each argument component directly relates to other components or relations (for undercuts), and categorising a detected relation into a class in a pre-defined annotation scheme (*e.g.* ATTACK or SUPPORT).

The first two subtasks are often referred to together as component identification.

2.2.1 Component identification

Component identification involves identifying the boundaries of argumentative segments in a text (*i.e.* segmentation) and classifying them into different types of argument components (*i.e.* component classification). The goal is to separate argumentative segments from non-argumentative ones, and to identify different types of argument components.

Segments that are argumentative are referred to as argument components (or just components) throughout this thesis. In other works on argument mining, they are also referred to as argumentative discourse units (ADUs) (Peldszus and Stede, 2013, Al Khatib et al., 2016b, Wachsmuth et al., 2018). Boundaries of argument components are not always sentence boundaries. Instead, they can be at any point in a sentence. As a result, an argument component can be either a subpart of a sentence, a complete sentence, or a text span covering several consecutive sentences or even paragraphs. The choice of component boundaries usually depends on the objectives of the research and the nature of the target text.

Pre-defined categories of argument components are often based on their functional roles in the argument (*e.g.* PREMISE and CLAIM) (Stab and Gurevych, 2017), their dialogical or pragmatic roles (*e.g.* PROPONENT, OPPONENT, and CALLOUT) (Ghosh et al., 2014, Peldszus and Stede, 2015a), and the nature of their content (*e.g.* POLICY, VALUE and FACT) (Wagemans, 2016).

Consider the following text which we have already encountered in Figure 1.2:

> **Example 2.9**
> I think [it's not necessary for universities to provide laptops for all students]$_1$. [Not every student needs a laptop]$_2$, as [they can simply use the computers in the library for studying]$_3$. Also, [the "provided" laptops aren't really free — they are likely covered by the tuition fees paid by students themselves]$_4$. Some may argue that [universities are wealthy enough to pay using their own funds]$_5$, but I doubt it since [they can hardly offer a fair salary to their teachers]$_6$. And even if it's true, [these money should be used on something more important first]$_7$.

Here, segments in brackets are identified as argument components; the remaining text spans are considered non-argumentative. The labels of argument components depend on

the specific annotation scheme used. For example, possible labels of the first component in the example text can be PREMISE, PROPONENT, or POLICY.

2.2.2 Relation identification

Relation identification involves determining if a relation exists between two argument components or between a component and another relation and, if so, also determining the type of this relation.

Generally, in relation identification, two argument components are considered directly related if there is an inferential connection between them, which can either be supportive or opposing. The most commonly used types of argument relations are SUPPORT and ATTACK (Mochales and Moens, 2011, Stab and Gurevych, 2017). These two categories are based on the intuitive understanding that one argument component can either support or challenge another argument component. Relation types can be also defined based on other frameworks. For example, following the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), an argument relation can be classified as ANTITHESIS, CONCESSION, or EVIDENCE (Green, 2010).

By Pollock's definition given in Section 2.1.2.4, an undercut attacks the inferential connection between two components. This leads to a special kind of relation, which is always an opposing relation between a component and another relation.

In Example 2.9, component 2 and component 1 are considered to be directly related, and the relation between them could can be categorised as SUPPORT or EVIDENCE, depending on the specific annotation scheme used. Similarly, the relation between component 5 and component 4 can be categorised as ATTACK, and component 7 is an undercut for that relation. The structure of the argument in Example 2.9 are illustrated in Figure 1.2.

2.3 Chapter summary

I began this chapter by examining the definitions of argumentation and argument, including logos-centred and audience-centred definitions. Logos-centred definitions emphasise the logical structure and content of arguments, while audience-centred definitions emphasise the importance of adapting arguments to the audience's values, beliefs, and emotions.

Such different emphases corresponded to two crucial categories of argument analysis frameworks: argumentation models and argumentation schemes. The focus of argumentation models is on the structure of arguments, while argumentation schemes concern the reasoning strategies employed in arguments.

Besides introducing important argumentation models and argumentation schemes, I also discussed counterarguments as a means to challenge the soundness or validity of an argument, as well as the concept of undercuts as a specific type of counterargument.

Furthermore, I introduced argument mining as a task that aims to automatically extract and analyse arguments from natural language texts. I described two critical subtasks within argument mining: component identification and relation identification. Component identification involves segmentation of raw text and classification of argument components. Relation identification determines if a relation exists between two components or between a component and another relation, as well as the type of that possible relation.

Having described the prior theoretical research pertinent to my thesis, I will introduce in the next chapter relevant practical work in the field of argument mining.

Chapter 3

Related work

In this chapter, I begin by reviewing some existing annotation schemes and their resultant datasets in Section 3.1, followed by some influential approaches to argument mining in Section 3.2. I also introduce how existing approaches integrate external knowledge into argument mining models in Section 3.3.

After introducing the theoretical background in Chapter 2 and related work in this chapter, I then establish my vision for argument mining with informal arguments in online discussions (Section 3.4.1), with a special interest in undercuts (Section 3.4.2). This vision directly leads to the research objectives of this thesis in Section 3.4.3.

3.1 Annotation schemes and datasets

For successful argument mining model development, it is essential to have an annotation scheme that accurately captures the characteristics of arguments in concern. Meanwhile, it is also important for such a scheme to ensure satisfactory inter-annotator agreement. Typically, annotation schemes are presented and evaluated in conjunction with their resultant datasets. Therefore, in this section, I review various existing annotation schemes alongside their resultant datasets, scrutinising their respective features and limitations. Table 3.1 presents a list of recent argument mining datasets and their annotation schemes, with a preference for those that include informal text.

Annotation schemes used to create these datasets vary a lot in terms of the information extracted from the source text. For example, the annotation scheme used for the dataset by Al Khatib et al. (2016b) only contains the fine-grained categories of premises in an argument, while the one used for the QT30 dataset by Hautli-Janisz et al. (2022) describes the hierarchical structure among all argument components and their relations to the structure of dialogues. In this thesis, the desired output of argument mining models is the complete structure of monological arguments, which should include both labelled components and relations. Only some of the datasets listed in Table 3.1 contain such

Dataset	Source	Scheme	Domain	Genre	Size (token)
Walker et al. (2012)	Debate posts on political issues on 4forums.com	Stance and relation on the post level	Politics	Forum post	1,031,398
Peldszus and Stede (2015a)	Manually created short texts in a controlled experiment	Proponent-opponent macro-structure	Politics, moral, daily life	Argumentative short text	7,846
Rinott et al. (2015)	Wikipedia articles concerning controversial topics randomly selected on idebate.org	Typed claim-premise pairs	Mixed	Wikipedia article	2,376,785
Abbott et al. (2016)	Conversational posts on forums and debate portals	Typed quote-response pairs	Mixed	Forum post	~63.8M
Al Khatib et al. (2016b)	Editorials on news portals	Typed premises	Mixed	Editorial	287,364
Habernal and Gurevych (2017)	User comments, forum posts, blogs and newspaper articles online	Modified Toulmin	Education	Online user-generated text	84,673
Lawrence and Reed (2017)	Transcripts of the BBC Radio 4 programme Moral Maze	Claim-premise pairs	Banking	Live debate	5,768
Niculae et al. (2017)	Online user comments about rule proposals regarding Consumer Debt Collection Practices	Directed graph with typed claims and premises	Finance	Forum post	~88,000
Stab and Gurevych (2017)	Essays written with given writing prompts on essayforum.com	Majorclaim-claim-premise	Mixed	Essay	147,271
Visser et al. (2020)	Transcripts of the 2016 US presidential debates, and reaction posts on Reddit	Inference Anchoring Theory	Politics	Live debate, forum post	97,099
Bhatti et al. (2021)	Posts about Planned Parenthood on Twitter	Typed premises	Abortion	Social network post	24,100 tweets
Hautli-Janisz et al. (2022)	Transcripts of political debates on BBC's "Question Time"	Inference Anchoring Theory	Politics	Live debate	280,000
Mim et al. (2022)	Initial argument and counterargument pairs written by experts	Template-based directed graph including node-edge relations	Education, law	Parliamentary debate	250 pairs

Table 3.1: Existing datasets for argument mining.

annotation, namely Peldszus and Stede (2015a), Habernal and Gurevych (2017), Stab and Gurevych (2017), Mim et al. (2022), and Niculae et al. (2017). The first four are closely

related to my research, and will be introduced in more detail in the following.

3.1.1 Persuasive Essays

The Persuasive Essays dataset introduced in Stab and Gurevych (2017) consists of 402 persuasive essays. These essays were randomly selected from an online forum where users upload their papers and essays for feedback.

The annotation scheme used to create this dataset results in a connected tree structure, illustrated in Figure 3.1. In this scheme, a MajorClaim represents the author's viewpoint on the topic. The text expressing this viewpoint is typically located in the introduction and conclusion of an essay. Each body paragraph comprises one or more Claims. A Claim has a stance attribute (for or against), indicating whether it supports or attacks the MajorClaim. Each Claim is supported or attacked by one or more Premise components, which in turn may be supported or attacked by additional Premises. The label of a relation between a Premise and a Claim, or between two Premises, is either Support or Attack.

Some notable characteristics of this dataset are as follows:

- Each essay may contain one or more MajorClaims. When multiple MajorClaims are present, they must express equivalent meaning and are treated as equivalent components. In such cases, the argument structure deviates from a tree structure.
- Claims without Premises are allowed.

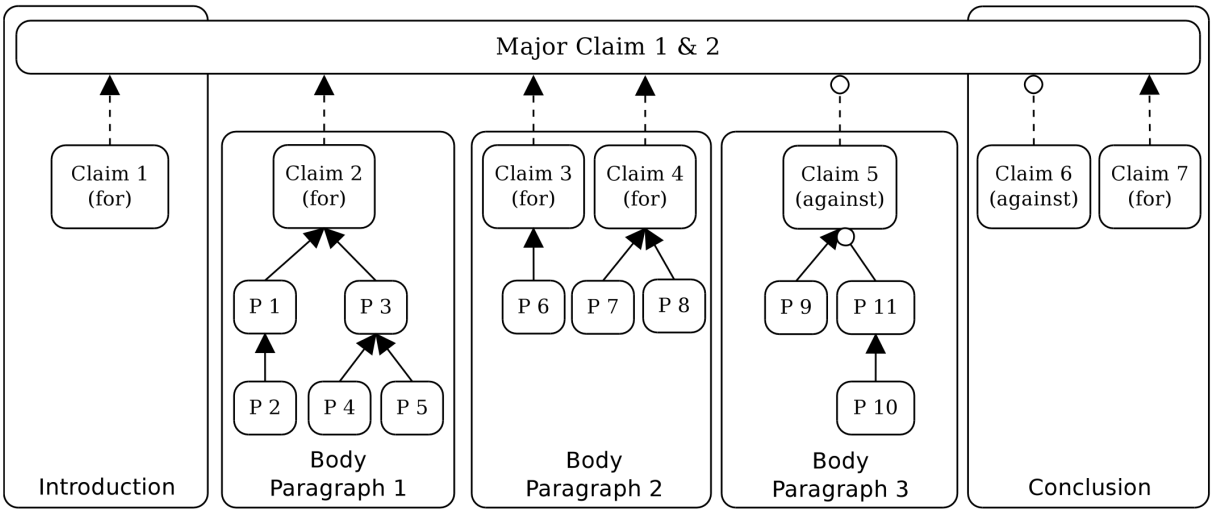


Figure 3.1: An example argument structure of the Persuasive Essays dataset. Edges represent argument relations: arrow-head = Support, circle-head = Attack. Dashed-edges indicate component stance: arrow-head = For, circle-head = Against. Reproduced from Stab and Gurevych (2017).

- Relations including Premise→Claim and Premise→Premise must occur within the same paragraph, while Claim→MajorClaim relations can reach across paragraph boundaries.

The annotation task for creating the Persuasive Essays dataset consists of text segmentation, component classification, and relation identification. The inter-annotator agreement for this dataset was reported as Krippendorff's unitised alpha (Krippendorff, 2004), which is an agreement measure that can take into account segmentation and categorisation simultaneously. For component identification, the scores are $\alpha = 0.81$, $\alpha = 0.52$, and $\alpha = 0.82$ for MajorClaim, Claim, and Premise, respectively. In terms of relation identification, the scores for Support and Attack are $\alpha = 0.71$ and $\alpha = 0.74$, respectively. The component categories most prone to confusion are Claim and Premise, which might be due to one annotator systematically never splitting sentences containing a Claim and a Premise.

While the simplicity of the annotation scheme of the Persuasive Essays dataset might have contributed to the relatively high inter-annotator agreement, it simultaneously constrains the depth of information that can be extracted from the source text. The predefined component categories — MajorClaim, Claim, and Premise — only denote the tier of components in the argument's overall hierarchical structure. For instance, a component placed at the top of a tree invariably falls under the MajorClaim category. As a result, this scheme does not provide any information about the content of components or the persuasion patterns involved. This drawback rules it out for the research in this thesis.

3.1.2 Gold Standard Toulmin

The Gold Standard Toulmin dataset presented in Habernal and Gurevych (2017) includes 340 texts selected from user comments, forum and blog posts, and newswire articles. The source text is considered informal due to its origin in user-generated web discourse. Texts in this dataset were selected arbitrarily from six controversial topics in the domain of education, such as homeschooling, redshirting, and single-sex education.

The prototype of the annotation scheme used for this dataset is Toulmin's model. Compared to the six component categories in Toulmin's model, there are only five component categories in the Gold Standard Toulmin scheme: claim, premise, backing, rebuttal, and refutation. We can see that Habernal and Gurevych omitted warrant from their scheme. This is because they observed that warrants were often expressed implicitly in arguments. Also, it is usually difficult to distinguish warrants from other categories in practice (Van Eemeren et al., 2019). During annotation, annotators were asked to first segment the raw text and then to label each component as one of the five categories.

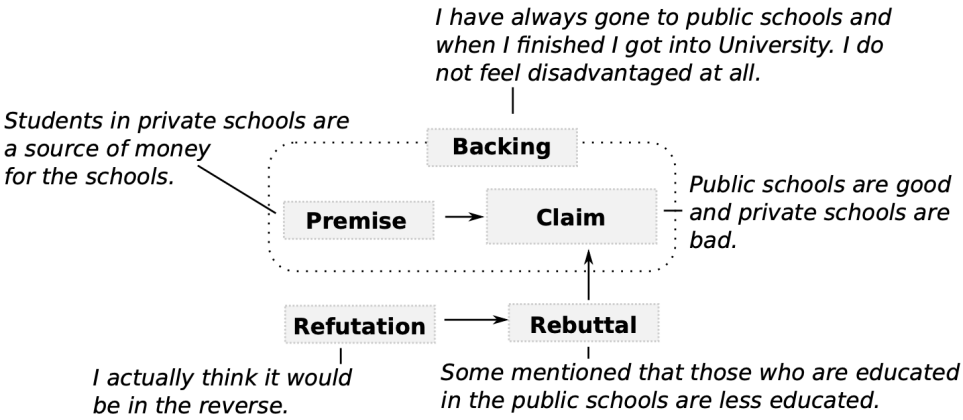


Figure 3.2: An example annotation of the Gold Standard Toulmin dataset. Reproduced from Habernal and Gurevych (2017).

The CLAIM component is required to exist in every argument, and can be either explicit or implicit. The other four components are optional and may or may not be present in the argument. An example of the annotated text in this dataset is shown in Figure 3.2. Although relations are not explicitly labelled, they can be inferred according to the nature of each type of component: a PREMISE always supports a CLAIM; a REBUTTAL always attacks a CLAIM; and a REFUTATION always attacks a REBUTTAL. A BACKING is not directly related to any component, but instead supports the entire argument.

The reported inter-annotator agreement score of the Gold Standard Toulmin dataset is $\alpha = 0.48$ in Krippendorff's unitised alpha. According to the authors, possible reasons for this low agreement include the long average length (around 11.5 sentences per text) and the low readability of texts, difficulties in deciding segment boundaries, the confusion between some certain types of components, and the perplexity brought by rhetorical questions, sarcasms, and fallacies which are frequent in online discussions.

Apart from those reasons, the overly complex annotation scheme could also be a factor that leads to the low agreement. This scheme categorises argument components according to their functional roles in an argument. However, those categories are not formally defined in Toulmin's model, as discussed in Section 2.1.2.2. Although the creators of the Gold Standard Toulmin dataset have modified Toulmin's model and reduced the number of component categories to five, this scheme might still be too complex. The Gold Standard Toulmin scheme is therefore also not a suitable scheme for the research in this thesis.

3.1.3 Microtext

The Microtext dataset introduced by Peldszus and Stede (2015a) consists of 112 German texts, along with professional English translations that preserve linearisation and argumentative structure. The source text was produced in a highly controlled text generation experiment. Participants were instructed to write a short argumentative text for a given

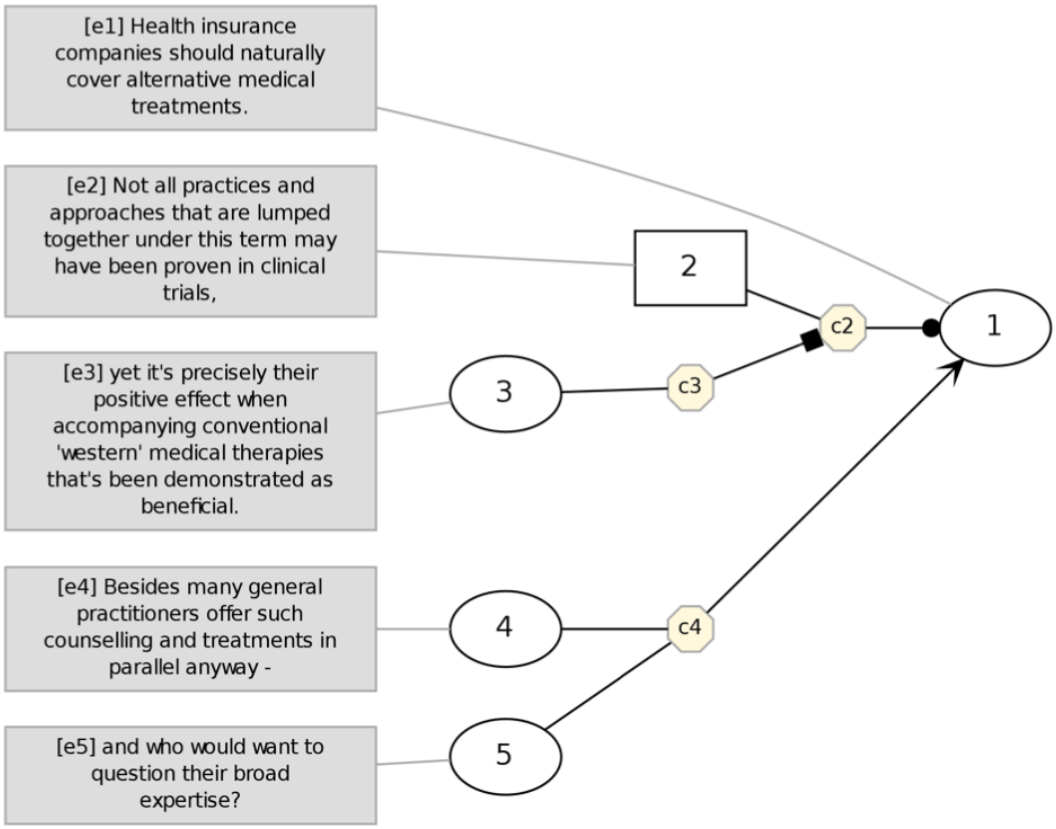


Figure 3.3: An example annotation in the Microtext dataset. Nodes represent argument components: round = PROPONENT, square = OPPONENT. Edges represent argument relations: arrow-head = SUPPORT, circle-head = REBUT, square-head = UNDERCUT. Reproduced from Peldszus and Stede (2015a).

question and were required to include certain elements in the text.

The Microtext annotation scheme is based on Freeman's theory of the macro-structure of argumentation (Freeman, 1991, 2011). The pre-defined categories for components in this annotation scheme are PROPONENT and OPPONENT. Relations in this scheme are SUPPORT, REBUT, and UNDERCUT. Additionally, this annotation scheme allows for combining multiple components in one move to form a linked argument, either for supporting or attacking moves. Figure 3.3 shows an example of the annotated text in this dataset. In this figure, component 3 undercuts the relation between component 2 and component 1. Notably, to the best of my knowledge, the Microtext dataset is the first and only one that contains explicitly annotated undercuts ($N = 62$), where undercuts are distinguished from direct attacks.

There are two types of elementary structures in the Microtext annotation scheme:

- **Component-component pair**: This elementary structure includes two components, where one component supports or attacks another, as well as the relation between them.

- **Component-relation pair**: This elementary structure includes a component and a relation that it attacks[1] (or rebuts, according to the Microtext nomenclature), and the relationship between them. The relation between the source component and the target relation in this elementary structure is labelled as UNDERCUT.

The concept of undercutting in this scheme is similar to Pollock's definition of undercutting defeaters in Definition 2.2, in that they both describe the undercutting move as an attack on the inference process. However, in Pollock's definition, an undercutting move is shown through an undercutting defeater, which is a component, while in this scheme, it is shown through an argument relation called undercut. Formalising undercuts as a relation is a more straightforward and intuitive solution, because the undercutting move is concerned with how a component relates to its target, rather than with the attribute of this component itself.

In the Microtext dataset, sentence boundaries are used as segment boundaries. The annotation task therefore only consists of component classification and relation identification. The inter-annotator agreement was measured at $\kappa = 0.83$ in Fleiss' kappa (Fleiss, 1971). This agreement is relatively high considering the fact that undercuts are included in the annotation scheme. This result indicates that the explicit annotation of undercuts might be feasible.

However, the source text of the Microtext is relatively too simple. On average, each document only contains about five segments. Additionally, the highly controlled text generation process leads to some degree of homogeneity among texts in the dataset. The relatively small size of this dataset and the homogeneity among texts may also lead to overfitting for some machine learning models. There is also a problem with the categorisation of argument components in its annotation scheme. The two component categories, namely PROPONENT and OPPONENT, are designed based on the hypothetical dialectical exchange in arguments. As a result, the focus of such a design is the dialectical role of each component. In contrast, what I want to explore instead is the persuasion patterns contained in the components, such as "argument from example" and "argument from analogy". It would allow for a more nuanced understanding of the rhetorical strategies employed in arguments, focusing on how different components contribute to the persuasive power of a text. The Microtext scheme therefore also does not fulfil my requirements.

3.1.4 LPAttack

The LPAttack dataset by Mim et al. (2022) consists of 250 pairs of argument and counterargument. The source text (Naito et al., 2022) was written to simulate parliamentary

[1]Similar to the Gold Standard Toulmin scheme, the Microtext scheme also does not include WARRANTS, in which a component supports a relation. No discussions on this decision are provided by Peldszus and Stede (2015a).

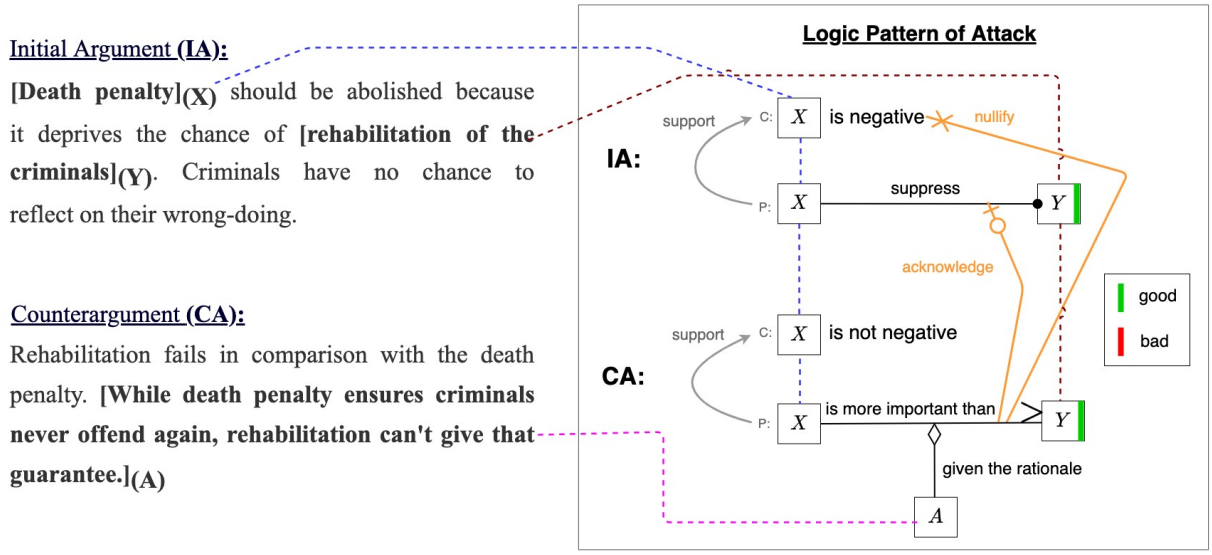


Figure 3.4: An example annotation of the LPAttack dataset. C = Conclusion, P = Premise. Reproduced from Mim et al. (2022).

debates. For each topic, there are two rival teams: the "prime minister" (PM) and the "leader of the opposition" (LO). The PM advocates for the topic, while the LO opposes it. The PM's speech serves as the initial argument (IA), with the LO's speech responding as the counterargument (CA).

The primary goal of the annotation scheme was to capture the underlying logic patterns that characterise attacks in arguments. Each document in this dataset includes an IA and a CA, each of which features only one Conclusion and one Premise, as shown in Figure 3.4. The LPAttack annotation scheme redefines the smallest annotation unit from traditional clause-sized argument components to concepts (*e.g.* "death penalty" and "rehabilitation of the criminals" in Figure 3.4). Each of these concepts has pre-defined attributes, mainly based on value judgments such as Good and Bad. These concepts are then interconnected by relations based on rhetorical moves including Promote, Suppress, and Nullify. Such concepts and relations together form the components in the dataset — Conclusions and Premises. These same relations are employed to connect components in the initial argument and the counterargument, thereby revealing the logic patterns of attacks between them.

Although undercuts are not explicitly represented in this annotation scheme, they can be inferred from the annotated logic patterns of attacks. For example, in Figure 3.4, the CA agrees with the premise in the IA (*i.e.* "death penalty deprives the chance of rehabilitation of the criminals"), but it disagrees with the conclusion that "death penalty should be abolished" because "death penalty is more important than rehabilitation of the criminals". Such an attacking relation is in fact an undercut, in which neither the premise nor the conclusion is directly attacked, but the reasoning process is challenged instead.

The inter-annotator agreement for the LPAttack dataset was reported in Cohen's kappa (Cohen, 1960). When IA-pattern, CA-pattern and attack pattern are considered a combined representation of a debate, the inter-annotator agreement score is $\kappa = 0.49$. The score increases to $\kappa = 0.63$ when each type of pattern is considered separately. The annotation task is structured as a template-choosing and slot-filling task. Typically, this format simplifies the process and leads to higher agreement scores. However, the agreement for this dataset is moderate at best. This could signal certain inherent complexities or ambiguities within the annotation scheme: it includes nine relation types and four attribute types, resulting in twenty unique nuanced logic patterns of attacks within the dataset.

Although the aim and the depth of the LPAttack annotation scheme are different from what I envision for my research, the distribution of logic patterns within this dataset provides an interesting empirical finding regarding undercuts in arguments. Approximately 40% of all logic patterns of attacks in the LPAttack dataset can be interpreted as undercuts. This finding suggests that undercuts frequently serve as a device for attacks and that human annotators are generally proficient at detecting them.

3.2 Argument mining approaches

In this section, I will present an overview of recent argument mining approaches, both pipelined and end-to-end ones. These approaches are listed in Table 3.2. The last three columns in the table indicate whether an approach performs the following subtasks: SEG = segmentation, CC = component classification, and RI = relation identification. Approaches listed above the double line are pipelined, and those below are end-to-end. I will discuss the strengths and limitations of these approaches, which constructs the foundation of my own research in this thesis.

3.2.1 Pipelined approaches

Pipelined approaches to argument mining break down argument mining into several subtasks, typically including segmentation, component classification and relation identification. These subtasks are processed sequentially, and the intermediate outputs are then combined in some way so that the full argument structure is generated. This section also covers approaches that focus only on some specific subtasks to provide a comprehensive review.

In recent pipelined approaches, segmentation is often jointly modelled with component classification as a token-level sequence labelling problem, using the BIO scheme to encode segment boundaries and component labels to encode component categories (Schulz et al., 2018, Mayer et al., 2020). Recurrent neural networks (RNNs) (Medsker and Jain, 2001) and long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are frequently used for these sequence labelling tasks.

Approach	Description	Task		
		SEG	CC	RI
Sardianos et al. (2015)	Component classification using CRFs with linguistic features and word embeddings	✗	✓	✗
Persing and Ng (2016)	Segmentation using handcrafted rules, component classification and relation identification using feature-based maximum entropy, ILP for joint inference	✓	✓	✓
Hou and Jochim (2017)	Sentence pair classification for relation identification using feature-based logistic regression within a multi-task framework using Markov logic networks	✗	✗	✓
Stab and Gurevych (2017)	Component classification and relation identification using feature-based SVMs, ILP for joint inference	✗	✓	✓
Schulz et al. (2018)	Sequence labelling using LSTMs for component identification	✓	✓	✗
Gemechu and Reed (2019)	Decomposing components into four new functional components, classification these new components and identifying the relations between them	✗	✓	✓
Reimers et al. (2019)	Component classification using RNNs with and without topic information	✗	✓	✗
Chakrabarty et al. (2020)	Component classification using BERT, relation identification using BERT and XGBoost	✗	✓	✓
Mayer et al. (2020)	Sequence labelling using RNNs for component identification, sentence pair classification and multiple choice using BERT for relation identification	✓	✓	✓
Ruiz-Dolz et al. (2021)	Sentence pair classification for relation identification using various Transformer-based pre-trained models	✗	✗	✓
Peldszus and Stede (2015b)	Generating argument graphs using feature-based evidence graph with MST decoding	✗	✓	✓
Eger et al. (2017)	Dependency parsing and sequence labelling using various neural networks	✓	✓	✓
Niculae et al. (2017)	Generating argument graphs using a factor graph model with SVMs and LSTMs	✗	✓	✓
Morio et al. (2020)	Dependency parsing using biaffine networks	✗	✓	✓
Bao et al. (2021)	Dependency parsing using a neural transition-based model	✗	✓	✓
Galassi et al. (2021)	Joint component classification and relation identification using multi-task attentive residual networks	✗	✓	✓

Table 3.2: Existing approaches to argument mining.

Component classification is sometimes formulated as a text classification problem which is solved by feature-based machine learning algorithms (Sardianos et al., 2015, Persing and Ng, 2016, Stab and Gurevych, 2017). Methods such as conditional random fields (CRFs) (Lafferty et al., 2001), maximum entropy classification (Nigam et al., 1999), and support vector machines (SVMs) (Hearst et al., 1998) are often used. Features used in these approaches include n-grams, verbs, discourse indicators, and word embeddings. Some other approaches employ feature-free neural networks (Reimers et al., 2019, Chakrabarty et al., 2020, Mayer et al., 2020). These methods use RNNs and LSTMs, and pre-trained Transformer-based models (Vaswani et al., 2017) like BERT (Devlin et al., 2019).

Relation identification is often formulated as a sentence-pair classification problem. The features of two components are fed into feature-based machine learning algorithms (Persing and Ng, 2016, Hou and Jochim, 2017, Stab and Gurevych, 2017), or the two components are concatenated into one piece of text so that text classification techniques can apply (Chakrabarty et al., 2020, Mayer et al., 2020, Ruiz-Dolz et al., 2021). Pre-trained Transformer-based models such as BERT, ALBERT (Lan et al., 2019), and RoBERTa (Liu et al., 2019b) are often used because sentence-pair classification is a built-in ability of these models.

During the combination of each subtask's output, global optima can be achieved by joint learning or multi-task learning, where the dependencies between subtasks are utilised. Both Persing and Ng (2016) and Stab and Gurevych (2017) use integer linear programming (ILP) (Schrijver, 1998) for a joint inference on component classification and relation identification. Hou and Jochim (2017) use Markov logic networks to jointly model stance detection and component classification in order to improve the performance on component classification. These approaches allow for better information sharing between subtasks, and have been shown to improve performance over traditional pipelined approaches.

There are some limitations to pipelined approaches. One is error propagation, which can occur when the performance of one subtask is dependent on the quality of the output from the previous one. For example, if the segmentation task is not accurate, then the component classification task may classify the wrong component, which can then cause the relation identification task to produce incorrect results. This may lead to a decrease in the overall performance of the system.

Another limitation of such approaches is that they often require human experts to carefully design and handcraft features for each subtask, which can be time-consuming and requires extensive domain knowledge.

Despite these limitations, pipelined approaches to argument mining are still widely used in the field, and they have their place. They are useful for investigating the relationship between different subtasks and for identifying areas where performance can be improved. Furthermore, pipelined approaches can be combined with end-to-end approaches to achieve a better balance between accuracy and efficiency. Overall, the success of pipelined approaches depends on the careful design of each subtask and the effective management of error propagation.

3.2.2 End-to-end approaches

End-to-end approaches to argument mining allow for the prediction of the full argument structure with a single model, and have been gaining popularity due to their advantages over pipelined approaches, including avoiding error propagation and eliminating the need for designing different models for different subtasks. However, such approaches also pose

a challenge due to the difference in granularity among the subtasks of argument mining: segmentation is typically a token-level task, where each token is labelled separately; component classification can be solved at either the token-level or segment-level, where each segment is assigned a label; relation identification is typically a segment-level task, where the relationship between two or more segments is identified. As a result, many existing end-to-end approaches avoid the segmentation subtask by assuming that the input text for their models is already segmented (Peldszus and Stede, 2015b, Niculae et al., 2017, Morio et al., 2020, Bao et al., 2021, Galassi et al., 2021).

If the segmentation subtask is ignored, one way to approach end-to-end argument mining is to use a multi-task framework with neural networks. For instance, Galassi et al. (2021) utilise a shared dense layer and a shared LSTM layer to encode a pair of segments, along with an attention mechanism (Vaswani et al., 2017) and a residual connection (He et al., 2016), and jointly predict segment types and relation types using three dense classifiers at the bottom of the neural model. However, this approach predicts each segment pair independently. Each time the classifiers only see the text in the segment pair, without the access to other text in the argument. It is therefore less likely to reach a global optimum.

Some end-to-end approaches directly predict the full argument structure so that global information is always available. As the full argument structure is often represented as a tree or a graph, some end-to-end approaches choose graph models to learn argument structures. For example, Peldszus and Stede (2015b) propose using evidence graphs (Wang and Daniels, 2005) to learn segment types and relations between segments. Each node in the evidence graph represents a segment in the text, and the score of each edge is a combination of various probabilities in terms of relations predicted by feature-based linear log-loss models. From each evidence graph, a minimum spanning tree (MST) (Graham and Hell, 1985) can then be generated by the Chu-Liu-Edmonds algorithm (Chu and Liu, 1965, Edmonds, 1967). Similarly, Niculae et al. (2017) adopt factor graphs (Kschischang et al., 2001) to learn full argument structures. A factor graph decompose a full structure into smaller units and dependencies between them, which are called "factors" of the graph. Niculae et al. use SVMs and LSTMs to parameterise factors of the graph. For inference, they use the alternating directions dual decomposition algorithm (Martins et al., 2015). These two approaches ignore the segmentation subtask, assuming that input text is already segmented.

The tree or graph structure of arguments also enables some end-to-end approaches to formulate argument mining as a dependency parsing problem. For example, Morio et al. (2020) use bidirectional LSTMs (BiLSTMs) (He et al., 2016) to encode argument components, and a biaffine dependency parser (Dozat and Manning, 2018) to classify components and their relations. Bao et al. (2021) propose a neural transition-based model

to predict the dependency structure of arguments. They use BERT to encode components, LSTM to encode parser states, and a dense classifier to predict transition actions. These two approaches also assume that the input text is already segmented.

In contrast, Eger et al. (2017) address the segmentation subtask in their proposed approach based on dependency parsing. They develop a dependency representation for arguments, which is designed to convert segment-level dependencies to token-level dependencies for relations. In Section 5.1, this dependency representation will be explained further and will be compared with my own dependency representations. Eger et al. explore several feature-based parsers (McDonald et al., 2005, Bohnet and Nivre, 2012) and neural parsers (Dyer et al., 2015, Kiperwasser and Goldberg, 2016), but neither type of parsers achieves satisfactory performance. In addition to dependency parsing, Eger et al. (2017) also investigate other end-to-end neural argument mining approaches, including sequence labelling (Ma and Hovy, 2016), multi-task tagging (Søgaard and Goldberg, 2016), and relation extraction (Miwa and Bansal, 2016). Among all the approaches tested, relation extraction performs best.

Eger et al. (2017) experiment with two input variations: the document-level approach, which means the parsers process one entire document at a time; and the paragraph-level approach, which means the parsers process one paragraph of a document at a time. The paragraph-level approach requires a post-processing step where the paragraph-level outputs are merged to form the final argument tree. In their experiments, the paragraph-level approach always outperforms the document-level approach.

Formulating argument mining as dependency parsing is intuitively a promising approach because argument structures and dependency parsing structures are similar in some ways. However, Eger et al. (2017) tested one such approach and then discarded it due to its unsatisfactory performance. In this thesis, I also frame argument mining as dependency parsing, and address the segmentation subtask as well. By developing a more efficient dependency representation for arguments and a more powerful dependency parser, I demonstrate that dependency parsing is indeed a promising approach for argument mining, a fact that was overlooked in the past.

3.3 External knowledge integration into argument mining

External knowledge integration is a crucial component in many NLP tasks, including argument mining. External knowledge can refer to various information sources beyond the available training data, such as domain-specific ontologies, curated knowledge graphs, or general world knowledge. By incorporating external knowledge into argument mining models, additional information that may not be explicitly stated in the text can be

accessed, thereby potentially enhancing the accuracy and robustness of the models. In this section, I present an overview of two types of external knowledge integration techniques in argument mining systems: feature-based knowledge integration and transfer-based knowledge integration.

3.3.1 Feature-based knowledge integration

Feature-based knowledge integration is a technique where external knowledge is incorporated into argument mining models as additional features. These features can capture domain-specific information or background knowledge that may not be directly present in the text.

One type of feature-based knowledge integration is the use of pre-trained word embeddings (Ando et al., 2005, Mikolov et al., 2013, Peters et al., 2017). These embeddings represent general linguistic characteristics learned from unsupervised natural language text. They have become an essential component of contemporary NLP systems, providing enhancements over embeddings trained from scratch (Turian et al., 2010, Peters et al., 2018). Many neural-based argument mining models (Eger et al., 2017, Schulz et al., 2018, Reimers et al., 2019, Morio et al., 2020, Galassi et al., 2021) incorporate pre-trained word embeddings as an integrated component without specifically recognising them as external knowledge. However, Fromm et al. (2019) use pre-trained word embeddings as a feature that incorporates external knowledge, and explicitly acknowledge the improvement brought by the embeddings.

In addition to pre-trained word embeddings, another approach for using knowledge from unsupervised natural language text as features is the clusters-as-features method, as named by Søgaard (2013). This method is adapted by Habernal and Gurevych (2015) in their work on component classification, which involves collecting unlabelled sentences and posts from online debate portals, clustering them using word embeddings, and then using the distance between a target segment and all cluster centroids as real-valued features in the classification algorithm.

Specific linguistic features, either crafted by humans or extracted by NLP tools, can also be used as external knowledge and integrated into argument mining models. These features can include syntactic information and sentiment information, among others. For example, syntactic information such as syntactic parse trees and part-of-speech tags are frequently used in feature-based argument mining approaches (Persing and Ng, 2016, Niculae et al., 2017, Stab and Gurevych, 2017), and are sometimes used in neural-based approaches (Eger et al., 2017). Sentiment information, such as sentiment word count and emotion expression count, is used by Park (2016) as external knowledge to improve component classification. These features can be directly fed into feature-based algorithms or encoded as vectors for neural models.

Curated knowledge graphs are another type of external knowledge source. They are believed to be more reliable than databases that are purely automatically generated, as they often undergo review and validation by humans, or contain knowledge from expert-created databases or human input (Nickel et al., 2015). Nodes in a knowledge graph typically represent entities, concepts, and events, while edges represent relationships and attributes. DBpedia (Auer et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) are two influential curated knowledge graphs, both containing structured information extracted from Wikipedia. ConceptNet (Liu and Singh, 2004, Speer et al., 2012, 2017) is also often used in NLP research, whose sources include the Open Mind Common Sense project (Singh et al., 2002), Open Multilingual WordNet (Bond and Foster, 2013), and DBpedia. Researchers have explored various ways to incorporate curated knowledge graphs into argument mining models. For instance, Fromm et al. (2019) integrate the knowledge in DBpedia into their component classification model by generating embeddings for entities in the knowledge graph using TransE (Bordes et al., 2013), which are then used in the same way as word embeddings in the neural classification model. Meanwhile, Abels et al. (2021) use a more complex method to incorporate knowledge graphs into their component classification model. They first select subgraphs containing relevant entities in the sentence from Wikidata using latent Dirichlet allocation (Blei et al., 2003). They then prune the subgraphs based on the cosine similarity between word embeddings of each entity and the sentence vector using a dynamic breadth-first search algorithm. Finally, they encode the pruned subgraphs into a single vector using a BiLSTM layer, which generates a final sentence representation through an attention mechanism (Hermann et al., 2015).

Feature-based knowledge integration has several advantages, including the ability to integrate diverse external knowledge sources and flexibility in adapting to various model structures. However, a drawback is the increase in computational complexity when a large number of features are added, leading to longer training time and potential overfitting due to the noise in the input features.

3.3.2 Transfer-based knowledge integration

Transfer-based knowledge integration involves transferring knowledge from other source domains, tasks, and languages to improve the performance of the target task (Ruder, 2019). In recent years, transfer learning has been closely linked to neural networks, and as a result, most transfer-based knowledge integration techniques in argument mining are employed in the context of neural-based approaches.

Knowledge transfer across domains and languages are similar. They both pre-train models on data from other sources other than the target data. The pre-trained models are either used to directly predict on the target domain or language, or are first fine-tuned with the training data in the target domain or language. In the case of domain adaptation for

argument mining, Al Khatib et al. (2016a) first map unlabelled source data from an online debate portal to a set of pre-defined class labels, and then directly use a feature-based classifier trained on the mapped source data to predict component classes on the target data. For cross-lingual knowledge transfer, Rocha et al. (2018) train several neural models for the relation identification subtask on an English dataset and then fine-tune these models on a Portuguese dataset. Ajjour et al. (2017) evaluate knowledge transfer across domains on the segmentation subtask using datasets from three different domains, including essays, editorials, and web discourse, by training sequence labelling models on two of the three datasets and testing on the remaining one.

Knowledge transfer across tasks involves pre-training a model on a related source task and then using it for the target task. Usually, the source task and the target task should be based on the same text. For example, in the work on argument mining by Accuosto and Saggion (2019), a BiLSTM sequence labelling model is first trained on the SciDTB dataset (Yang and Li, 2018) for a discourse parsing task. Then, the encoder in the trained BiLSTM model is used to generate token representations for various combinations of argument mining subtasks on the same dataset.

With the rise of powerful pre-trained Transformer-based language models such as BERT and RoBERTa, a new hybrid method for transfer-based knowledge integration has been employed in many studies in argument mining. As these models are pre-trained on general natural language text with training tasks including masked language modelling and next sentence prediction, fine-tuning them for argument mining can be seen as a hybrid knowledge transfer technique across domains and tasks. Additionally, sequence labelling and sentence-pair classification are built-in functions of such pre-trained models, allowing for their direct use in all argument mining subtasks without the need to alter model structures. For example, Fromm et al. (2019) adopt an off-the-shelf pre-trained BERT model for component classification. Rodrigues and Branco (2022) pre-train a RoBERTa model on a wide range of source data for masked language modelling and then fine-tune it on the Persuasive Essays dataset for all argument mining subtasks. Dutta et al. (2022) fine-tune a pre-trained Longformer (Beltagy et al., 2020) with unsupervised data from an online discussion forum called ChangMyView on Reddit, using a modified masked language modelling training objective, and use it for component classification. For relation identification, they adapt a prompt-based strategy (Liu et al., 2023a) to predict the conjunctions between two components in a prompt, and then classify the relation according to the predicted conjunctions.

Transfer-based knowledge integration offers several advantages, such as being computationally efficient and requiring less labelled data for the target task. The new hybrid transfer strategy, in particular, is free from issues related to the quality of source data and task similarity. This is because pre-trained Transformer-based language models are

trained on a massive amount of unsupervised natural language data. However, existing approaches using this strategy can only perform argument mining in a pipelined manner, as the built-in functions of such models do not support direct end-to-end argument mining. Therefore, in this thesis, I will explore new methods for knowledge integration using powerful pre-trained language models, with the goal of supporting end-to-end argument mining with informal text.

3.4 Goal of this thesis

The discussion above concludes my examination of how prior research approaches arguments. The main difference between previous studies and my research objectives lies in the informal nature of online discussion texts, as contrasted with the more formal texts traditionally addressed. This difference in text implies that I need to find a new approach on my part. At the same time, it is essential to incorporate the successful ideas evident in the existing state of the art.

3.4.1 Vision

Informal arguments in online discussions often involve a multitude of contributors. My vision is to establish a form of argument mining that can effectively summarise multiple arguments on the same topic, thereby highlighting their similarities and differences. What is implemented in this thesis is only the analysis of individual arguments at a single-document level. However, to effectively conceptualise what such an analysis should encompass, we must also consider how arguments from several documents could be brought together. In the following, I will sketch my envisioned objectives for an approach to multi-document argument mining.

Consider Figure 3.5. It represents ten individual arguments on the same topic "private schools *vs.* public schools", manually annotated by myself[2]. The ten arguments are represented in boxes of ten different colours (blue, red, green, *etc.*). Please note that the text in the figure is very small, and it is not necessary for the reader to be able to read the text. The purpose of the image is to convey the intuition behind how complex multi-party arguments could be displayed and interpreted.

There are clusters of subtopics (represented by grey boxes) that span across different coloured boxes, indicating shared subtopics among various arguments. For example, the large grey box at the bottom right of Figure 3.5 addresses the comparison of teacher quality in private and public schools (detail shown in Figure 3.6), whereas the large grey

[2]This graph was produced before the finalisation of the annotation scheme in Chapter 4, so the annotation in this graph is not necessarily in line with the annotation scheme proposed in this thesis. For illustration purposes, such detail should not matter.

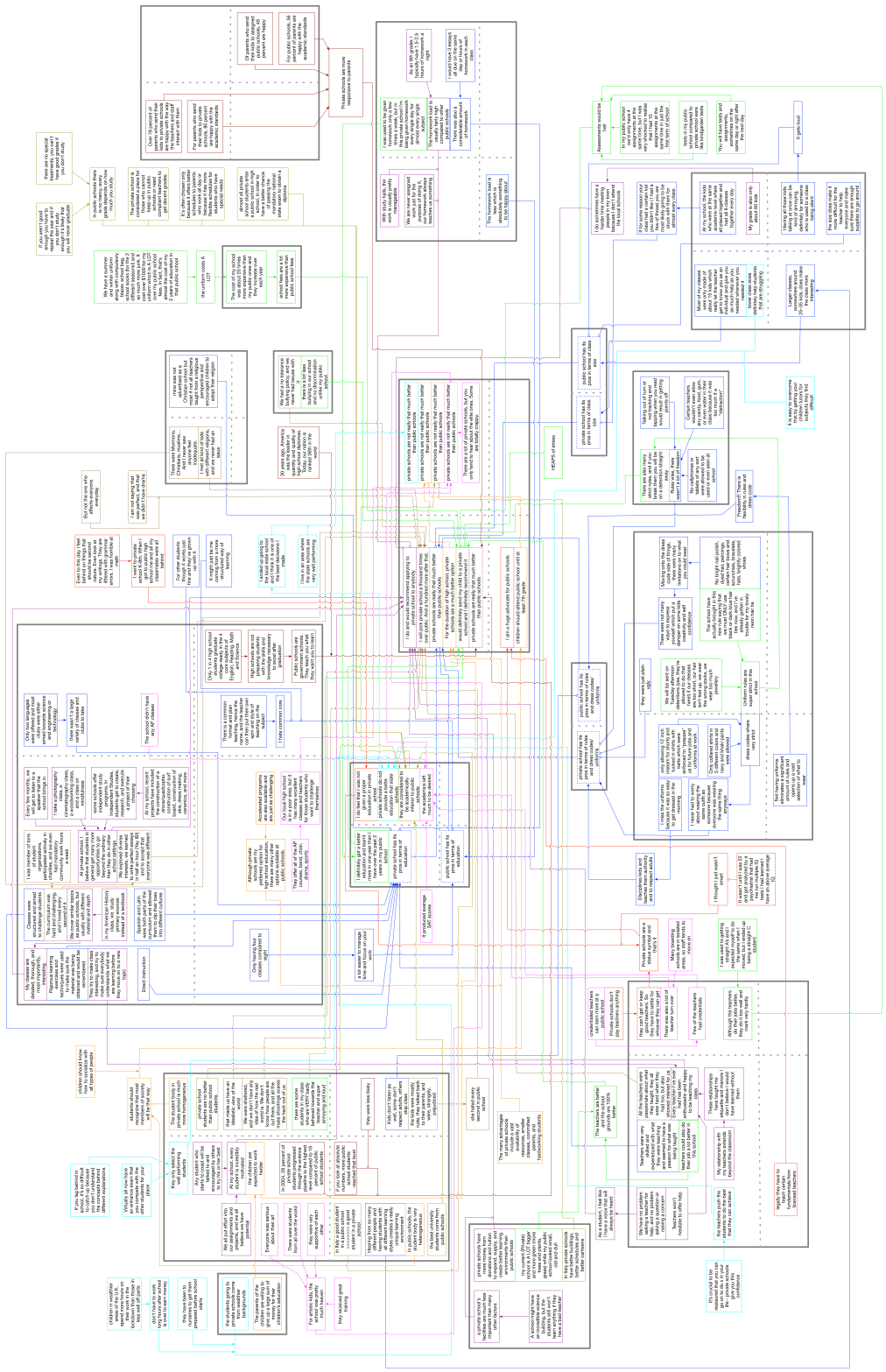


Figure 3.5: A graph summarising ten arguments on the topic "private schools *vs.* public schools". Coloured boxes represent argument components. Edges represent argument relations: arrow-head = SUPPORT, circle-head = ATTACK.

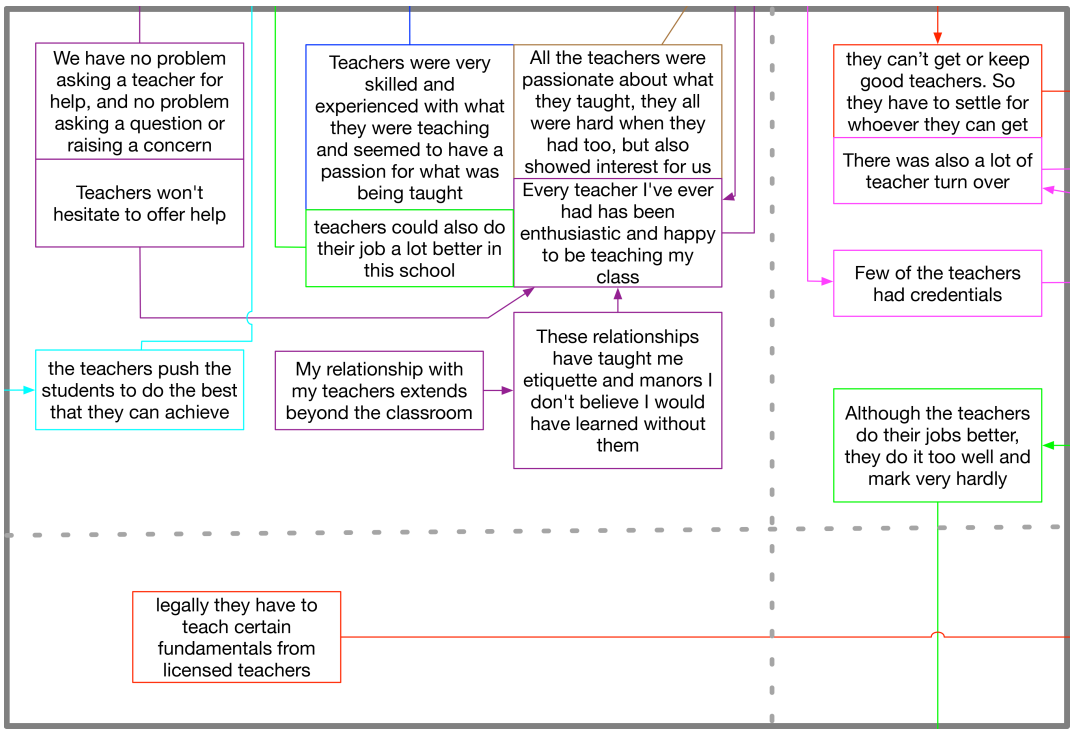


Figure 3.6: Detailed view of the grey box at the bottom right of Figure 3.5.

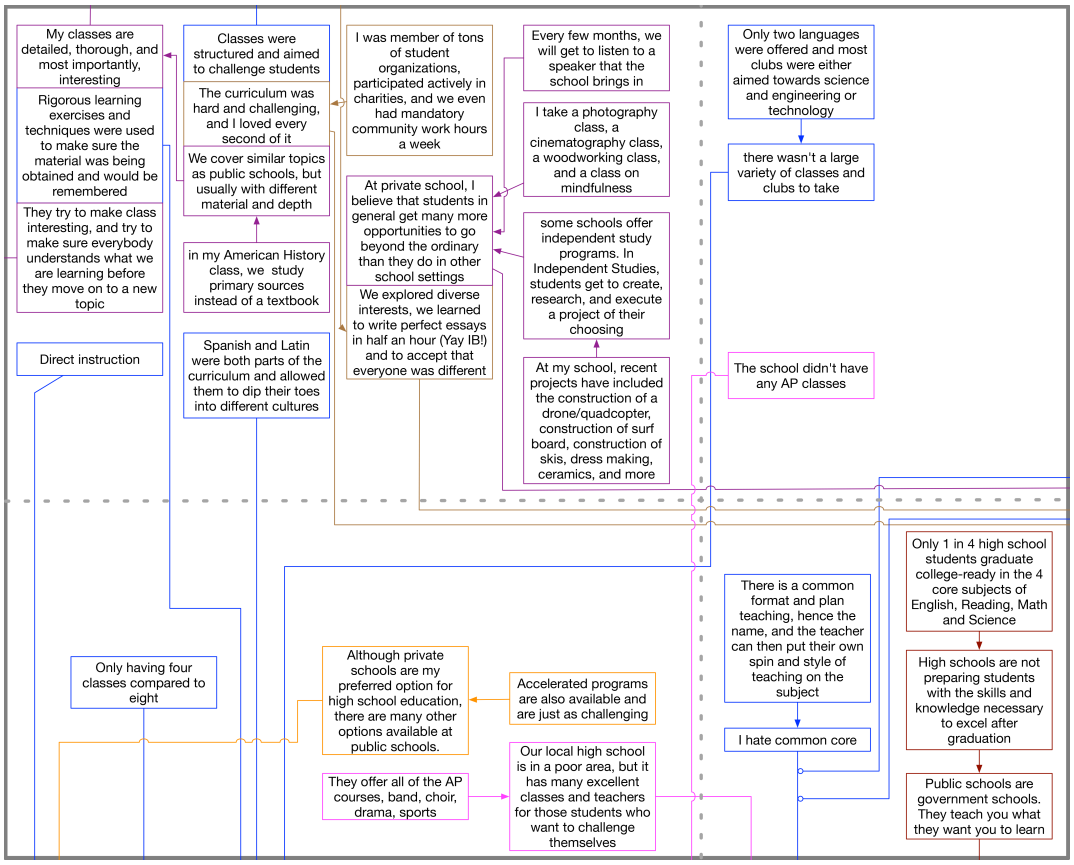


Figure 3.7: Detailed view of the grey box at the centre top of Figure 3.5.

box at the centre top discusses opportunities and special classes available at each type of school (detail shown in Figure 3.7). The size of these boxes could be interpreted as the relative importance of each subtopic for a certain topic in online discussions. This is because the more contributors and components engage with a specific subtopic, the more relevant or prominent it is in its community. Such insights could be utilised for real-time automated statistics, providing a dynamic overview of how an argument is being deliberated in online discussions.

Inside each large grey box, components are strategically organised into four quadrants. Each quadrant indicates the overall stance behind the opposing subjects (private schools *vs.* public schools), each with a positive or a negative stance. This segmentation yields four comprehensive quadrants in a left-to-right and top-to-bottom order: private school is good, private school is bad, public school is good, public school is bad. By systematically arranging components according to their inherent stance, we can formulate logically-grouped summaries of arguments. This can be beneficial for individuals interested in surveying diverse viewpoints on a specific topic. It can also be useful for those trying to broaden their understanding by examining arguments from opposing sides. My hope is that this might enable people to see beyond their own bubble.

If we were able to create such a structure automatically, we would be able to see much more:

- We can zoom in and see similarities in subarguments. Interestingly, the same facts are sometimes interpreted positively by one group and negatively by another. Consider, for instance, the box at the centre bottom that discusses the subtopic of school uniforms (detail shown in Figure 3.8). Both proponents and opponents of private schools recognise that these institutions tend to have stricter dress codes. Despite sharing the same premise, they arrive at very different conclusions. The blue argument favouring private schools regards uniforms as beneficial, positing that a uniform policy alleviates concerns about clothes-related competition. Conversely, the green argument against private schools views the strict uniform rules as a restriction of individual freedom.

- Note that this analysis explicitly express undercuts. One example, depicted in detail in Figure 3.9, involves the premise that private schools assign a large amount of homework, which seemingly attacks the conclusion "I would recommend private school". However, the author undercuts this connection with two reasons: one stating the existence of study groups and the other asserting that the homework is relevant. These two undercuts justify why a significant homework load is not necessarily bad in this context.

- When analysing text in online arguments, I found that people often argue from

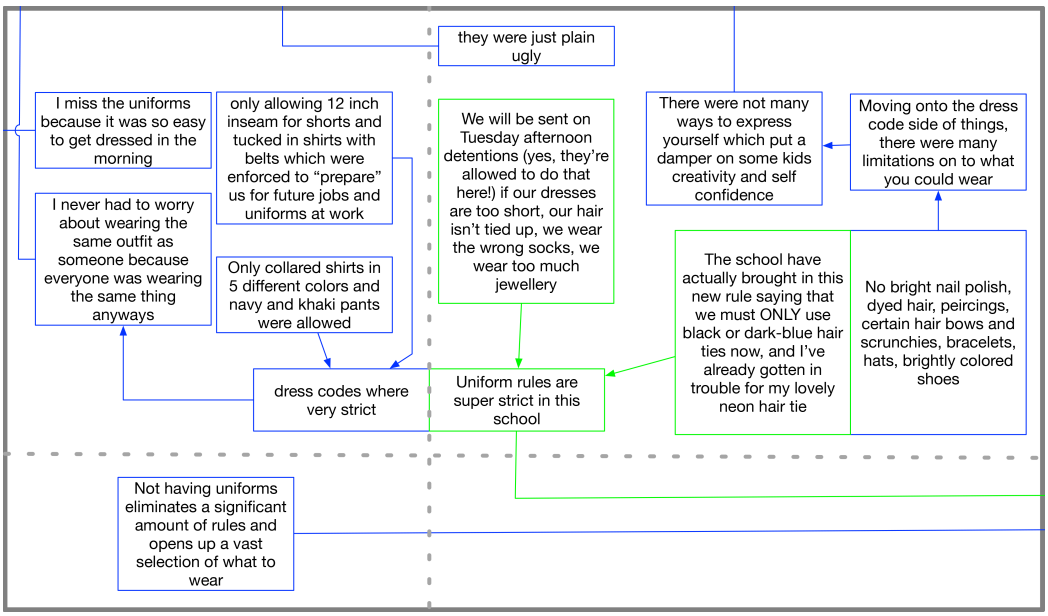


Figure 3.8: Detailed view of the grey box at the centre bottom of Figure 3.5.

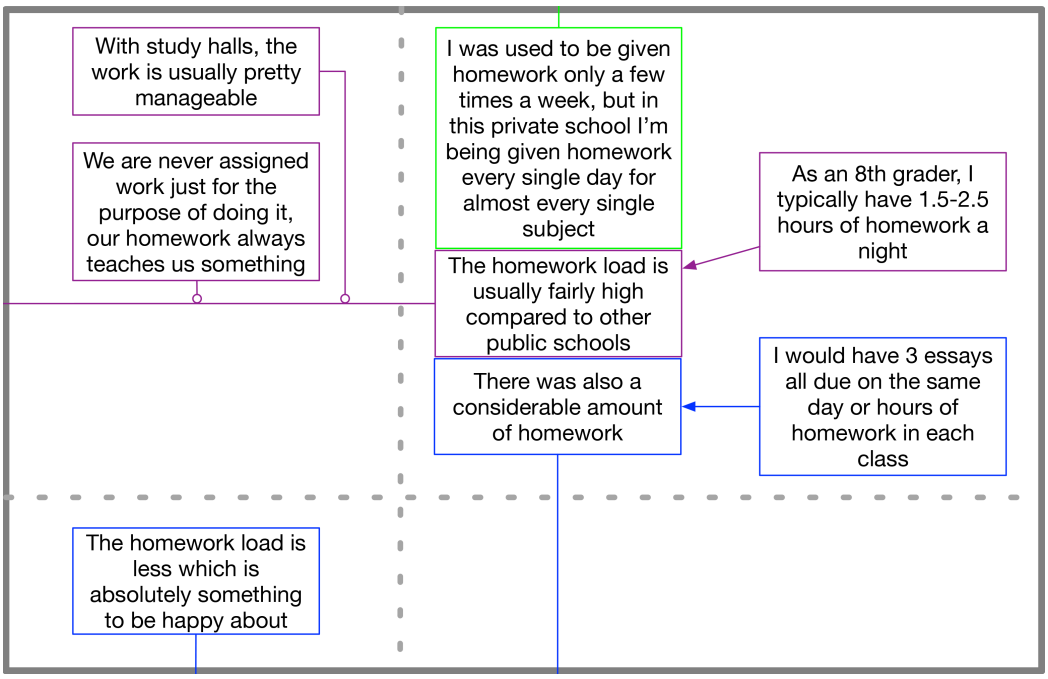


Figure 3.9: Detailed view of two example undercuts in Figure 3.5.

personal experience. This is often taken to be a weak argumentation strategy because such experiences do not generalise. However, isolating such experiential narratives from the more factual portions of an argument can be an effective means of spotlighting personal experiences. The following examples can be found in Figure 3.5:

– "I hate common core"

– "I thought I just wasn't smart. It wasn't until I was 22 . . . that I learned I had an above average IQ"

Such experiences could also serve as the basis of experience-based statistics. This analysis thus presents numerous opportunities for facilitating a more personalised perspective within the broader context of an argument.

- In the source text (not shown in the figure), I also found declarations of authors' personal backgrounds, such as the following:
  - "As someone who went to private school from k-8 and switched to public for high school ..."
  - "I was a public school teacher who attended a parochial elementary and a private secondary school, so I have divided loyalties"
  - "I want to start off this answer by saying that I have never attended a public school. I've been at the same private elementary school from pre-kindergarten until this year, my eighth grade year of high school"

  These examples illustrate that there are different roles, *i.e.* different stakeholders in such arguments: parents, pupils and teachers. This information can possibly either enhance or limit the strength of the argument, and can therefore be useful for others to interpret their arguments.

In summary, I have sketched a structure made up out of multiple arguments. I consider this structure as a potential basis for a new task of large-scale argument summarisation, which I call "multi-document argument mining". This task could go in many different directions. In the rest of my thesis, I will consider how the underlying argument structure (*i.e.* the annotation scheme for single arguments) should look like and how it can be automatically extracted. The aim is to enable such summarisation tasks in the future. In this thesis, I will present an annotation scheme for single arguments designed by myself, and will present automatic methods of recognising the structures of single arguments. I will leave the realisation of large-scale argument summarisation for future work.

3.4.2 Undercuts

Undercuts have not received the attention in the NLP literature they should have. As we have seen earlier in this chapter, Peldszus and Stede (2015a) include undercuts in the Microtext dataset, but they do not subcategorise them or provide any automatic method to recognise them in their follow-up experiments. As we have also seen, Mim et al. (2022) also present some ideas that touch upon the phenomenon of undercuts, although they do not explicitly mention undercuts when doing so. But nobody has studied which kinds of undercut strategies exist, nor is there large corpus data about them available.

I included undercuts as a subcategory of relations in my annotation, because I think they are not only frequent in online discussions, but also important for other reasons. For instance, undercuts have been connected to the values involved in arguments (Mim et al., 2022). Having access to an automatic, large-scale analysis of undercuts, as opposed to direct attacks, could be interesting for several disciplines and applications:

- Psychologists and sociologists could learn more about people's convictions related to values.
- Linguists could computationally analyse the specific language patterns used in undercuts.
- Researchers could test their hypotheses on social media using structures such as these, populated with thousands of structured arguments.

I have manually examined all the undercuts in my data, and found that there are different ways how a component can attack a relation in an undercut. I classified these undercuts into three categories that I formulated. These categories are:

- **Rejection**: rejecting the relation by denying the relevance between the source component and the target component. In Example 3.1, component 3 undercuts the relation between component 2 and component 1. It directly rejects this relation by saying that other people's feelings are irrelevant to same-sex marriage.

  Example 3.1
  Question:
  Do you support same-sex marriage?
  Answer:
  "[**I literally cannot think of one coherent reason to not support same sex marriage**][1].
  [...]
  [**Some people are disgusted by same sex relationships**][2].
  [**You feelings don't matter here**][3].
  [...]"

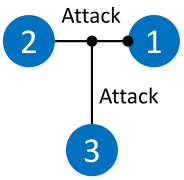


Figure 3.10: The structure of the argument in Example 3.1.

- **Low importance**: questioning the importance of the relation, or providing more important reasons. In Example 3.2, component 2 undercuts the relation between component 1 and component 3. It does not completely deny the relation. Instead, it weakens this relation by saying that bravery and distinction in battle are less important than what people fight for.

  Example 3.2
  Question:
  Do you think all Confederate monuments statues should be removed from public areas in the South?
  Answer:
  "[. . .]
  The point here is that although [**bravery and distinction in battle are important**][1], [**WHAT people fight for is ultimately of far greater importance**][2].
  [. . .]
  For this reason, [**they do not deserve to be honored**][3]".

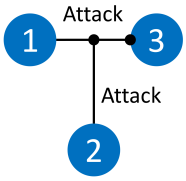


Figure 3.11: The structure of the argument in Example 3.2.

- **Alternative option**: stating that the current solution is not the only option, or providing alternative options. This kind of undercuts often appears in arguments about policies. In Example 3.3, component 3 undercuts the relation between component 2 and component 1. Component 3 attacks this relation by pointing out that a firm and explicit dress code does not necessarily require uniforms — there are many other clothing options that conform to this dress code.

  Example 3.3
  Question:
  Is it necessary for children to wear school uniforms?
  Answer:
  "[. . .]
  Every, and I mean [**every argument in favor of school uniforms doesn't hold water**][1].
  [. . .]
  [**There's absolutely nothing wrong with a firm, explicit dress code**

requiring coverage from shoulders to knees, banning see-through fabrics, and limiting obscene slogans on shirts][2]. [**That doesn't require uniforms**][3]. "

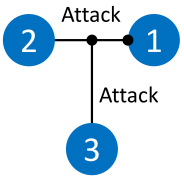


Figure 3.12: The structure of the argument in Example 3.3.

As readers may have noticed, all the relations being undercut in the examples above are attacks. This is because in a monological argument, undercuts are usually used to attack the possible opposing opinions that are raised by the author themselves. Such "attacks on attacks" are meant to support the author's own argument.

3.4.3 Research objectives

In this thesis, I propose to apply argument mining techniques to large-scale informal text in online discussions. I will not only explore approaches to argument mining in general but also address specific challenges that arise when dealing with arguments in online discussions. My research objectives in this thesis are as follows:

1. **Developing an annotation scheme for informal text in online discussions.** In this chapter, I have found that existing annotation schemes are either too simple or too complex for argument mining with informal text in online discussions. Therefore, I will now set out to create my own annotation scheme, in accordance to the envisaged large-scale summarisation task I just introduced. This annotation scheme should capture logical structures and persuasion patterns in informal arguments in online discussions while still maintaining satisfactory inter-annotator agreement. Since I consider undercuts a unique and important argument structure, the desired scheme should include undercuts. The scheme is described in Section 4.1, applied on informal text collected from an online question-answering platform (Sections 4.2, 4.3 and 4.4). I also conduct an annotation study to evaluate the resultant dataset in Section 4.5.

2. **Applying argument mining to informal arguments in an end-to-end approach.** Considering the limitations of existing approaches to argument mining, there is a need for effective end-to-end approaches that can generate argument structures directly from raw text. To achieve this objective, in Chapter 5 I formulate end-to-end argument mining as dependency parsing. I do this by designing two dependency representations for arguments (Section 5.1) and two neural dependency parsers (Section 5.2). My approach includes undercuts, something that no existing

approach is able to model (neither in informal or formal texts). I evaluate my approach on formal arguments without undercuts (Section 5.3) and informal arguments with undercuts (Section 5.4).

3. **Improving argument mining with external knowledge integration.** External knowledge should help argument mining with informal text in online discussions, but to the best of my knowledge the integration of external knowledge for model-agnostic end-to-end argument mining has not been extensively studied. I propose feature-based and transfer-based approaches that explore various knowledge sources and integration techniques in Section 6.1, and evaluate and compare these approaches in Section 6.2.

3.5 Chapter summary

In this chapter, I provided an overview of existing work on argument mining, laying the foundation for subsequent chapters.

I introduced relevant existing annotation schemes and their resultant datasets. Existing schemes are either too simple or too complicated to be directly used on informal arguments that I studied in this thesis. The source text for those datasets are also not ideal for my research interest in terms of genre or size. These factors became the reason for me to designed my own annotation scheme and to create my own argument mining dataset.

I reviewed different approaches to argument mining, including pipelined and end-to-end approaches. I also discussed their respective strengths and limitations. Pipelined approaches support investigation into specific subtasks, but they often suffer from error propagation and heavy feature-engineering. The amount of existing approaches that are truly end-to-end is limited, because segmentation is a token-level subtask, while the remaining subtasks are at the segment-level. The end-to-end approach based on dependency parsing by Eger et al. (2017) is especially relevant to my work, although this approach did not yield satisfactory results at that time.

Additionally, I reviewed existing methods for external knowledge integration into argument mining models. I grouped them into two categories: feature-based and transfer-based. In feature-based knowledge integration, external knowledge is incorporated as additional features. Advantages of feature-based knowledge integration include its inclusiveness for diverse external knowledge sources and its flexibility towards various model structures. Its potential limitations include longer training time and overfitting. Transfer-based knowledge integration transfers knowledge from other source domains, tasks, and languages. Recent work on knowledge transfer with pre-trained language models have been shown to be effective, but most existing methods of this kind do not support end-to-end argument mining. These observations motivated me to experiment with various methods for knowledge

integration into end-to-end models.

Having reviewed how existing research approaches argument mining theoretically and practically, I then presented my vision for graph-based representation of multiple arguments in online discussions. Based on my observations and vision, I listed my research objectives of this thesis, and each of the following chapters, except the last one, is dedicated to one of them.

Chapter 4

The annotation scheme and the Quora dataset

In this chapter, I propose an annotation scheme that aims to capture major logical structures and persuasion patterns in informal online arguments. Particularly, this annotation scheme includes undercuts.

I collected 400 arguments from Quora, the question-answering platform I talked about in the introduction. I manually pre-segmented the Quora dataset, and then trained two annotators to annotate this dataset using my annotation scheme. In chapters 5 and 6 this data will be used for my automatic experiments. To the best of my knowledge, this is the biggest argument mining dataset that contains annotation of undercuts.

I present the annotation scheme in Section 4.1, and describe the creation of the Quora dataset in Sections 4.2, 4.3 and 4.4. To evaluate the quality of the annotation, I conduct an annotation study and present it in Section 4.5.

4.1 The annotation scheme

My scheme includes four argument component categories and two argument relation types. The design of this scheme mirrors some of the facets of the established argumentation schemes and argumentation models I introduced in Chapter 2. It can be seen as an extension to the AIF reviewed in Section 2.1.4.

My scheme is illustrated in Figure 4.1. The unit of annotation for components in my scheme is a segment that can be part of a sentence, a sentence or a sequence of sentences. Besides "supports" (*e.g.* component 2 supporting component 1) and "attacks" (*e.g.* component 7 attacking component 3), my schemes also includes undercuts (*e.g.* component 4 undercutting the relation between component 5 and component 2). Component 9 is a stand-alone component. It is a STAKE, a component category I will explain in subsequent sections.

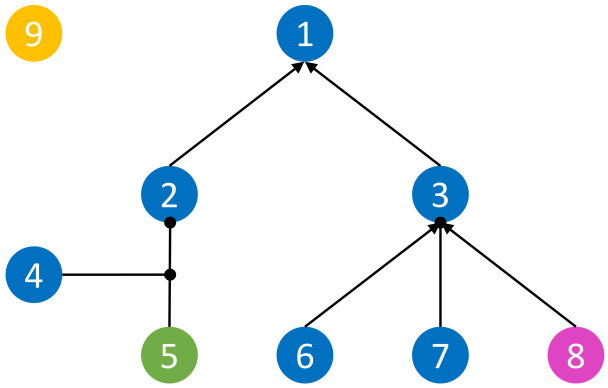


Figure 4.1: An example argument graph using my annotation scheme. Nodes in different colours represent components of different categories. Edges represent relations: arrow-head = SUPPORT, circle-head = ATTACK.

4.1.1 Component categories

The component categories in my annotation scheme are designed to reflect persuasion patterns embedded in components. In contrast, the simple claim-premise component categorisation in most annotation schemes only reflects the structural function of components. Each of the categories I define is related to one or more persuasion patterns in Walton et al.'s argumentation schemes (Section 2.1.3). The four component categories in my annotation scheme are:

- **PROPOSITION (PRP)**: Regular argumentative statements.
- **STAKE (STA)**: Declarations of the author's personal circumstances.
- **ANECDOTE (ANE)**: Descriptions of personal episodes the author experienced which are connected to their stance in the argument.
- **ANALOGY (ANA)**: Metaphorical explanations of the argument that can make the argument more understandable and relatable.

If a text segment does not fit into any of the pre-defined categories, it is considered as non-argumentative. I will now define the component categories.

4.1.1.1 Proposition

A PROPOSITION is a statement expressing one's beliefs and opinions (*e.g.* "I do support same-sex marriage"), or a statement about factual evidence that supports or opposes those beliefs and opinions (*e.g.* "Marriage is defined as the legally or formally recognised union of two people"). PROPOSITIONs are the most common components in most arguments, whether formal or informal. They are similar to the general components in most annotation schemes.

4.1.1.2 Stake

In informal online arguments, authors sometimes provide personal background information related to the topic at hand. They may do so to indicate that they are biased, which could decrease the credibility of their answers. Or they want to signal their expertise on the topic which comes from their personal experience; in that case, their credibility would be increased (Example 4.1). Such components are categorised as STAKEs.

> **Example 4.1**
> **Question**:
> Is China a good place to live in?
> **Answer**:
> [**I have been living in China for 12 years**]$_{STA}$. I'd say it's a perfect place to live in. For people who love big cities, you'll find [...]

This component category is closely related to some of Walton et al.'s argumentation schemes, namely "argument from position to know" (Section 2.1.3.1) and "argument from expert opinion" (Section 2.1.3.2). In my opinion, such components are not part of the argumentation *per se*, because they cannot function as an independent component either to support or attack another component or relation. However, they give information which can be essential for the interpretation and evaluation of the entire argument. I therefore treat these as meta-information and make them stand-alone components in argument graphs.

4.1.1.3 Anecdote

An ANECDOTE is a short story about the author or individuals personally known to them. Authors in online discussions sometimes use ANECDOTEs as evidence to support their arguments, as shown in Example 4.2. This component category is closely related to Walton's argumentation scheme "argument from example" (Section 2.1.3.3).

> **Example 4.2**
> **Question**:
> "Which group is better for the whole US? Republicans or Democrats?"
> **Answer**:
> Most Democrats are good, and the bad Republicans are recognisably awful.
> [**I've never had a Democrat look me in the eye and say "I think you deserve to be waterboarded for having a boyfriend." I have heard that from a Republican before, an unpleasant elderly relative who identified as a Republican because she agreed with their economic policies**]$_{ANE}$.

4.1.1.4 Analogy

Analogies are sometimes used in online discussions to imply that some conclusions that apply to A also apply to B, because B is like A, where B is the subject of the argument at hand (Example 4.3). Using analogies in an argument can help make complex concepts or ideas more understandable to the audience, by comparing them to something more familiar. This can increase the persuasiveness of the argument. The argument component category ANALOGY is reserved for such cases. This component category is closely related to Walton et al.'s argumentation scheme "argument from analogy" (Section 2.1.3.4).

Example 4.3
Question:
Do you support same-sex marriage? Why?
Answer:
I very much support equal marriage. **[Sexual orientations are just like various flavours of ice-cream. You may prefer vanilla but you should not prevent others choosing chocolate. Equal marriage is like that]**$_{ANA}$.

ANECDOTEs and ANALOGYs are included in my annotation scheme because I observed many of these in my initial study of the material in online discussions. The persuasion patterns behind them are usually considered to be relatively weak (Walton et al., 2008), so I think they are more likely to appear in informal arguments than in formal arguments. Therefore, in future studies, they might be useful for distinguishing between formal and informal arguments, and for evaluating the quality of arguments.

4.1.2 Relation types

Relations in my annotation scheme closely follow those in the Microtext annotation scheme (Section 3.1.3). The two types of elementary structures I inferred from the Microtext

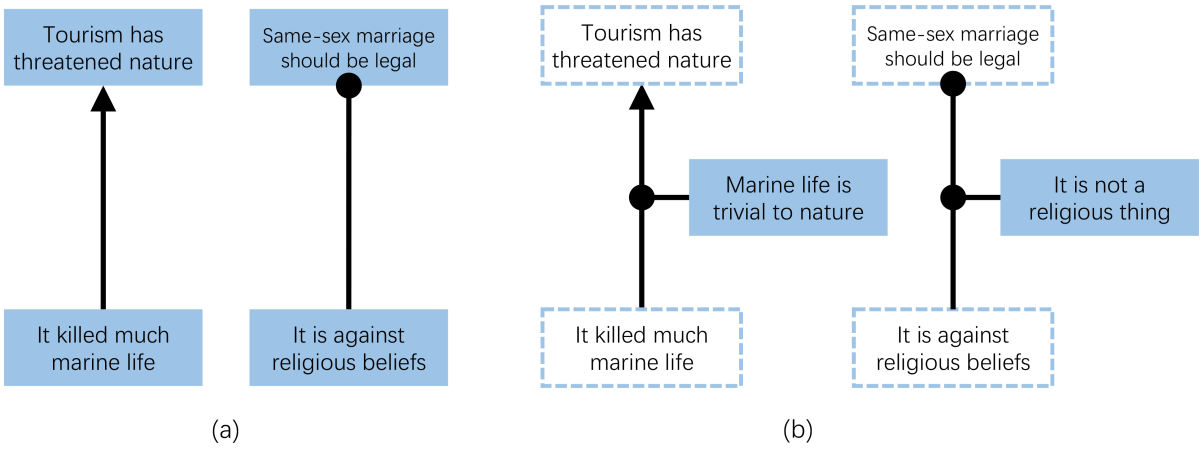


Figure 4.2: Examples of (a) component-component pairs and (b) component-relation pairs.

annotation scheme, namely component-component pairs and component-relation pairs, are kept in my scheme. If we ignore the edge labels, a component-component pair (Figure 4.2a) is equivalent to the basic argument structure that Walton (1996) refers to as a "single argument" (Section 2.1.2.1). A component-relation pair (Figure 4.2b) includes a component-component pair and one extra undercutting component. This component always attacks the relation in the component-component pair. Following Habernal and Gurevych (2017) and Peldszus and Stede (2015a), I also omitted warrants from my scheme.

In my scheme, the label of a relation is either SUPPORT or ATTACK. ATTACKs can be further categorised into direct ATTACKs and undercuts. This is the major difference in terms of relations between my scheme and the Microtext scheme. In the Microtext scheme, labels of relations include SUPPORT, REBUT and UNDERCUT. The Microtext scheme distinguishes between direct ATTACKs and undercuts through assigning them different labels. My distinction between them does not rely on relation labels, but on the elementary structure they occur in: ATTACKs always appear in component-component pairs, and undercuts in component-relation pairs.

4.1.2.1 Intermediate structures and final argument graphs

The final structure of my scheme is built by repeatedly applying the elementary structures. When doing so, there are some constraints:

- A source component can have only one target component or relation.
- A target component or relation can have multiple source components.
- Undercuts are limited to the first-order level and cannot be applied to other undercuts, meaning that an undercut cannot itself be the target of another undercut.

When multiple elementary structures are combined, I call the result an "intermediate structure", which is a structure between the elementary structures and the final argument graph. With the constraints above, two types of intermediate structures can result:

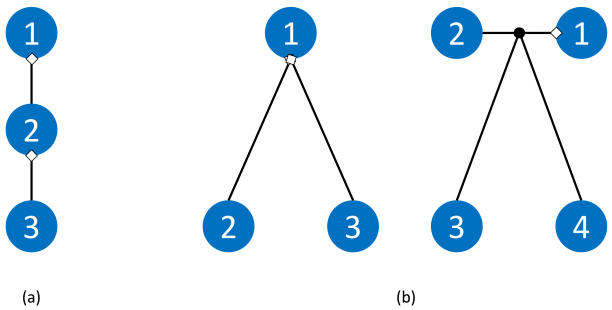


Figure 4.3: Examples of (a) serial and (b) convergent intermediate structures. Square-headed edges can represent either SUPPORT or ATTACK relations.

- Serial intermediate structure: in this intermediate structure, multiple components are related one after another, as illustrated in Figure 4.3a.
- Convergent intermediate structure: in this intermediate structure, a target component or relation is related to multiple source components, as illustrated in Figure 4.3b.

If we ignore the edge labels, the serial intermediate structure in my scheme is identical to the "serial argument" in Walton's argumentation model (Walton, 1996) that I have reviewed in Section 2.1.2.1. The convergent intermediate structure is similar to Walton's "convergent argument", except that the target in the convergent intermediate structure can be either a component or a relation.

In Walton's argumentation model, two other structures, namely "linked argument" and "divergent argument", are presented. However, they are not incorporated into my annotation scheme. I excluded linked arguments in order to avoid confusions because they closely resemble convergent intermediate structures. This resemblance is also noted in Walton (1996). In my scheme, all linked arguments are transformed into convergent intermediate structures. Divergent arguments are omitted because I fear that they might complicate the annotation process. I have imposed a rule to avoid possible divergent arguments: a source component can only target the component or relation that is most directly related to it. With all this in place, a final graph consisting of multiple intermediate structures can be constructed for an argument, such as the one in Figure 4.1.

4.2 Arguments on Quora

I use my annotation scheme described above to annotate arguments collected on Quora, where controversial topics usually receive a significant amount of attention and engagement from users.

Compared to user-generated content on other online discussion platforms such as ChangeMyView, Kialo, idebate.org, and Twitter, arguments on Quora are more in line with my research interest. On Quora, users can present detailed and well-reasoned points of view in their answers to a question. As a result, each answer on Quora can be seen as containing a stand-alone cogently structured complete argument, often supplemented with explanations and supporting evidence. Platforms including ChangeMyView, Kialo and idebate.org are specifically designed for interactive debates and discussions, with a focus on the process of changing one's mind through constructive conversations. Each post on such platforms may only contain an incomplete argument, and intertextual referencing is frequent among posts, making the analysis of such posts more difficult. Twitter, unlike those previous platforms, is a more open and informal platform, where arguments can take on a more combative tone. As a result, arguments there may not always be rational

and often lack the depth and nuance seen on the other platforms. Additionally, due to Twitter's character limit, arguments in tweets tend to be very short, with structures that may be too simplistic to warrant a detailed analysis. Therefore, I chose Quora over other online discussion platforms for my research in this thesis.

4.3 Collection method

I used a systematic method to collect data from Quora, in order to reduce bias in the distribution of topics. My aim was to collect 400 answers on Quora, evenly distributed across 20 topics. For each topic, I first chose one question, and then selected 20 answers to that question.

Ideally, the chosen questions should be controversial, have a diverse coverage, and be representative of the concerns of many users in online discussions. However, Quora does not provide a direct way to search for questions that meet these criteria. Therefore, in order to select candidate questions that I would later apply to Quora, I first turned to Kialo, which provides a comprehensive catalogue of argument topics, such as "Politics", "Society", "Science", and "Technology", as shown in Figure 4.4. These topics represent general domains of arguments on Kialo.

The data collection process consists of three steps, 1) topic and candidate question selection on Kialo, 2) question selection on Quora, and 3) answer selection on Quora.

In the first step, I chose the first 20 topics in the topic catalogue on Kialo. Under

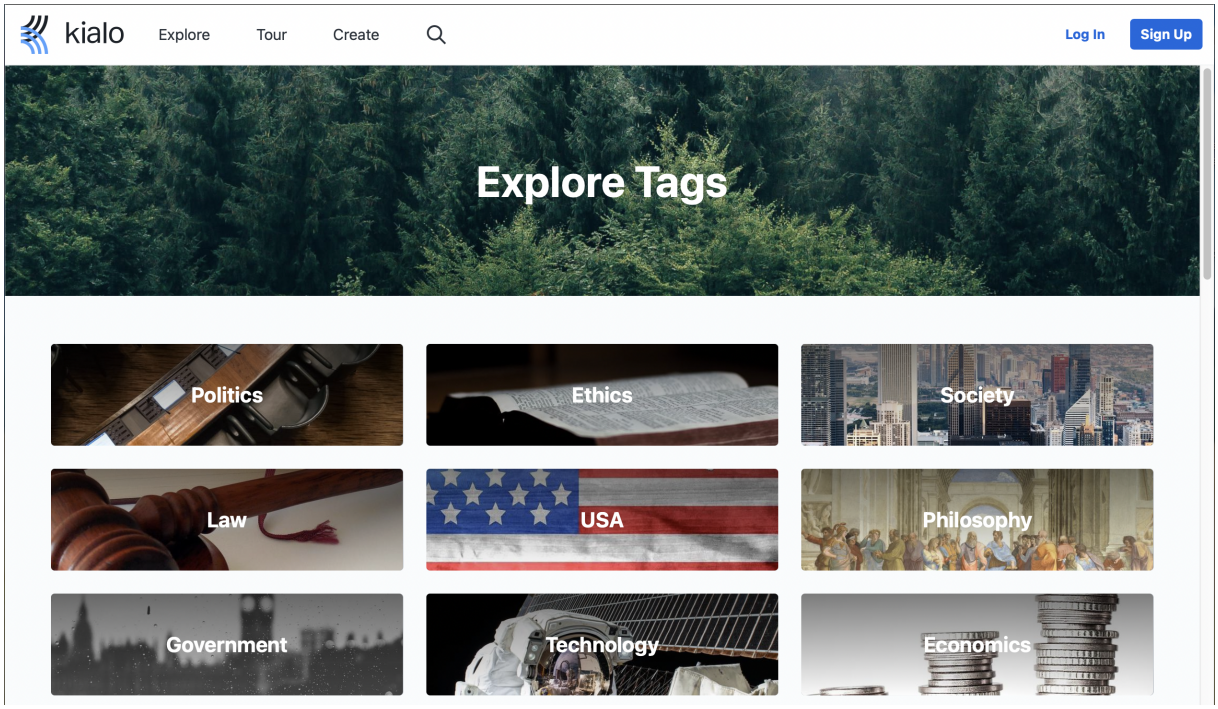


Figure 4.4: Topics on Kialo.

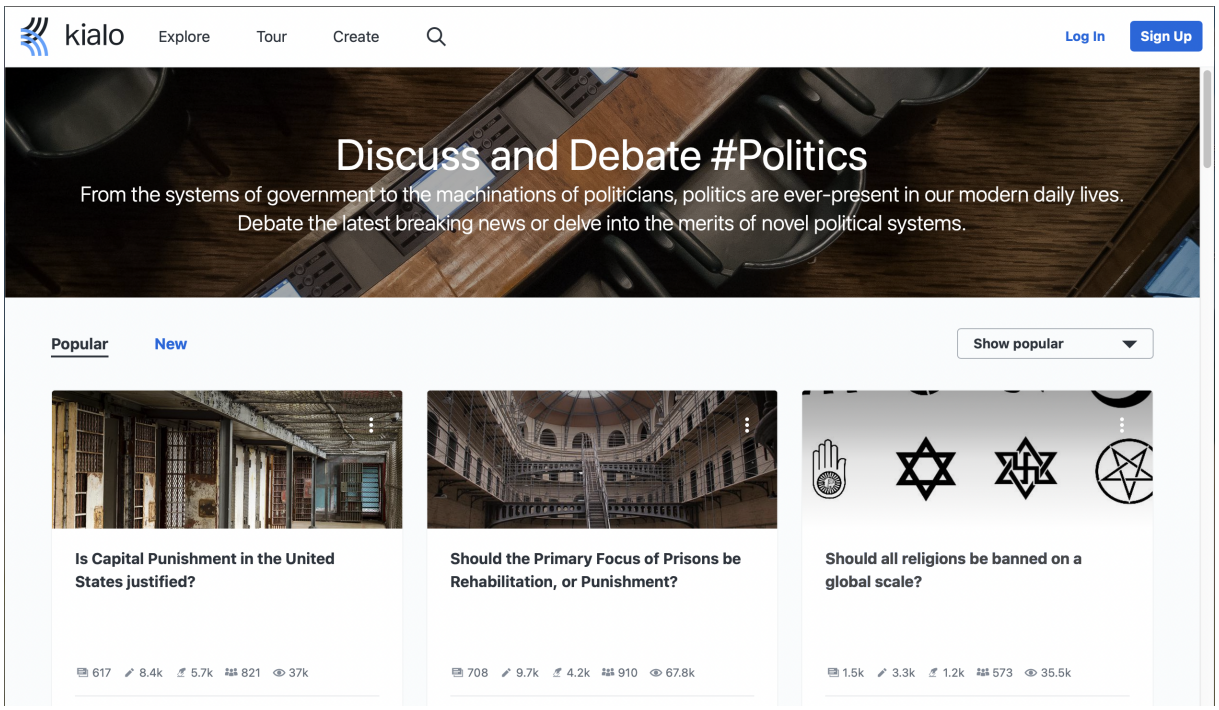


Figure 4.5: Controversial questions under the topic "Politics" on Kialo.

each topic (*e.g.* Politics), there are many specific controversial questions that people argue about (*e.g.* "Is Capital Punishment in the United States justified?"), as can be seen from Figure 4.5. I selected the first five questions by popularity under each of the 20 selected topic as candidate questions. This step resulted in 20 topics and 5×20 candidate questions. These candidate questions taken from Kialo would be next used to search for relevant questions on Quora.

In the second step, my aim was to select only one question on Quora for each of the 20 topics. This was achieved by searching on Quora using the Kialo questions from the first step, using its full string as a query. Among the relevant Quora questions returned by this search, I selected a suitable one. Suitability is defined as relevance to the Kialo question and availability of a sufficient amount of answers (at least 50). If a suitable Quora question was found, the remaining Kialo questions under this topic would no longer be processed. If not, I would manually choose key words (*e.g.* "Capital Punishment") from the current Kialo question for another search. If I still was unable to find a suitable Quora question, I would proceed to the next Kialo question. This step resulted in 20 selected Quora questions. The complete list of topics and candidate questions on Kialo, and selected questions on Quora can be found in Appendix A.

In the third step, I selected 20 answers to each selected Quora question in the second step. To be considered a qualified answer, the following criteria need to apply:

- **Relevance**. A qualified answer must directly address the topic raised in the corresponding question. Irrelevant answers, such as those that deviate from the

question (Example 4.4), or deny the necessity of the question (Example 4.5), are excluded.

> **Example 4.4**
> **Question**:
> Do you support Trump banning TikTok from the USA?
> **Answer**:
> Most people here are not fond of TikTok it seems. It ain't that bad if you use the search feature. The stuff that I get at the homepage is mostly cancer. The algorithm shows me what most people that use the app like, instead of my own preferences. Basically the algorithm only works if you're a normie lol. You just gotta use the search feature to find the content that you like.

> **Example 4.5**
> **Question**:
> What is better, democracy or communism?
> **Answer**:
> No we can't compare the both systems. It clearly meant for different geographical, economical and social location.
> [. . .]
> The solution for the best is considered as Mixture of Democracy and Socialist like India.

- **Argumentativeness**. I will manually judge the argumentativeness of an answer. A qualified answer must contain at least 30% material that is reasonably and objectively argumentative, rather than declarations of opinion without any reasoning or overly emotional statements.

- **Length**. The length of a qualified answer must be between 60 and 800 wordpieces after being tokenised by the BERT WordPiece tokeniser (Wolf et al., 2020). Based on my observation, answers shorter than 60 wordpieces often fail to express meaningful or well-formed arguments. The upper limit of 800 wordpieces is chosen to prevent answers from needing to be split into more than two parts when fed into a pre-trained BERT encoder, as BERT can only process a maximum length of 512 wordpieces at a time. In practice, only a few answers exceed this upper limit.

Using this data collection method, I selected 400 (20×20) answers that cover various topics such as politics, environment, education, equality, and so on. Since arguments on Quora happen in a question-answering context, I have appended each answer to its

corresponding question, placing a paragraph break after the question. To refer to this merged text consisting of the question and the answer, I will from now on use the term "document" when talking about the Quora dataset. The 400 documents were then manually annotated with my annotation scheme.

4.4 Annotation of the dataset

I first manually segmented the raw text in the Quora dataset, and then employed two annotators to annotate the pre-segmented dataset using my annotation scheme.

4.4.1 Pre-segmentation

Pre-segmentation can make the annotation process more efficient, as it reduces the cognitive load on the annotators, and can lead to higher consistency across annotations. However, pre-segmentation could introduce bias, because the decisions made during pre-segmentation may influence the way annotators perceive and label the data. In order to reduce bias, I devised segmentation rules before pre-segmentation and tried to adhere to them during pre-segmentation as well as I could. These rules are:

- **Conciseness rule**: A component should include all words that are relevant to the main idea of the component. Removing such words would change the core meaning of the component. In contrast, irrelevant tokens at either ends should be trimmed. Punctuation marks are considered as irrelevant. In Example 4.6, several words at the beginning of the sentence and the full stop at the end are considered irrelevant.

  Example 4.6
  "There is evidence of this from the fact that [**children in wealthier areas of the U.K. spend more hours on their work in the lockdown than those in less well off parts**]$_{PRP}$".

- **Splitting rule**: A sentence containing multiple statements should be split into multiple components only if one of the statements supports or attacks another (Example 4.7), or if these statements are not involved in the same argument relation (Example 4.8). In Example 4.7, the sentence is split because component 2 supports component 1. In Example 4.8, component 1 and component 2 in the first sentence are not involved in the same argument relation, because component 1 is supported by component 3 but component 2 is not. As a result, the first sentence is split, even though there is no supporting or attacking relation between the two components in it.

Example 4.7
"[**We should not develop Artificial Intelligence**]$^1_{PRP}$, because [**losing control of AI could be catastrophe for humans**]$^2_{PRP}$".

Example 4.8
"[**Develop Artificial Intelligence is too expensive**]$^1_{PRP}$, and [**losing control of AI could be catastrophe for humans**]$^2_{PRP}$. According to some relevant studies, [**a major company in America has invested 2 billion in the past 6 years**]$^3_{PRP}$".

- **Combination rule**: If multiple neighbouring statements about the same subject are used together to make a point, but neither of them on its own can serve an independent role as a stand-alone component in the argument, those multiple statements should be combined to form one argumentative component.

These pre-segmentation rules cannot cover all phenomena and do not always seem reasonable. The pre-segmentation rules, as currently defined, encounter limitations in capturing some linguistic phenomena and may not always align with our intuitive reasoning. Consider the sentence "I do not think those developing countries, which are consuming a big share of natural resources on the planet, should take less responsibility in reducing carbon emission". In this complex sentence, there exist two distinct components: an overarching claim "I do not think those developing countries should take less responsibility in reducing carbon emission" and a supporting premise "which are consuming a big share of natural resources on the planet". However, the first component is dislocated, being interrupted by the second component. My pre-segmentation rules do not support the combination of non-consecutive segments into a single component. Consequently, the example sentence would be annotated as a single component. This decision is a compromise between minimising annotation complexity and ensuring a reasonable level of representativeness in the annotation.

4.4.2 Annotation guideline and annotation process

I wrote a 25-page guideline for prospective annotators of the Quora dataset. This guideline contains a detailed introduction to the Quora dataset and my argumentation scheme, and a step-by-step guide on how to annotate a document in the dataset.

The step-by-step annotation procedure is as follows:

1. Annotators first read through the document.

2. They then decide which of the segments in the document best qualify as the main components (*i.e.* the components expressing the author's overall stance). This is

important because they first need to understand what the author is arguing for or against.

3. Next, annotators label all segments they consider argumentative as either PROPOSITION, STAKE, ANECDOTE or ANALOGY (these are explained elsewhere in guidelines). They leave all segments they consider non-argumentative unlabelled.

4. Starting from the first component to the last, for each component except STAKE, annotators decide whether it supports or attacks something — either an already identified component or an already identified relation. This relationship between the two entities is then marked and labelled as either SUPPORT or ATTACK. Sometimes the target relation of a component might appear later in the text. Annotators are therefore reminded that if they cannot find a target of a component at the moment, they can move on to the next component. After finishing all components in this step, they can go back to the remaining unlinked components and try to find their targets. This process can be repeated until all PROPOSITIONS, ANALOGYS and ANECDOTES are now connected to the argument graph.

5. If annotators are considering a potential ANALOGY or ANECDOTE candidate segment in step 4, but fail to find its target component or relation, this probably indicates that the candidate is not sufficiently closely related to the argument. As a result, they remove the current label of the candidate ANALOGY or ANECDOTE and leave it unlabelled.

A sample document annotated following the steps above is shown in Figure 4.6. Square

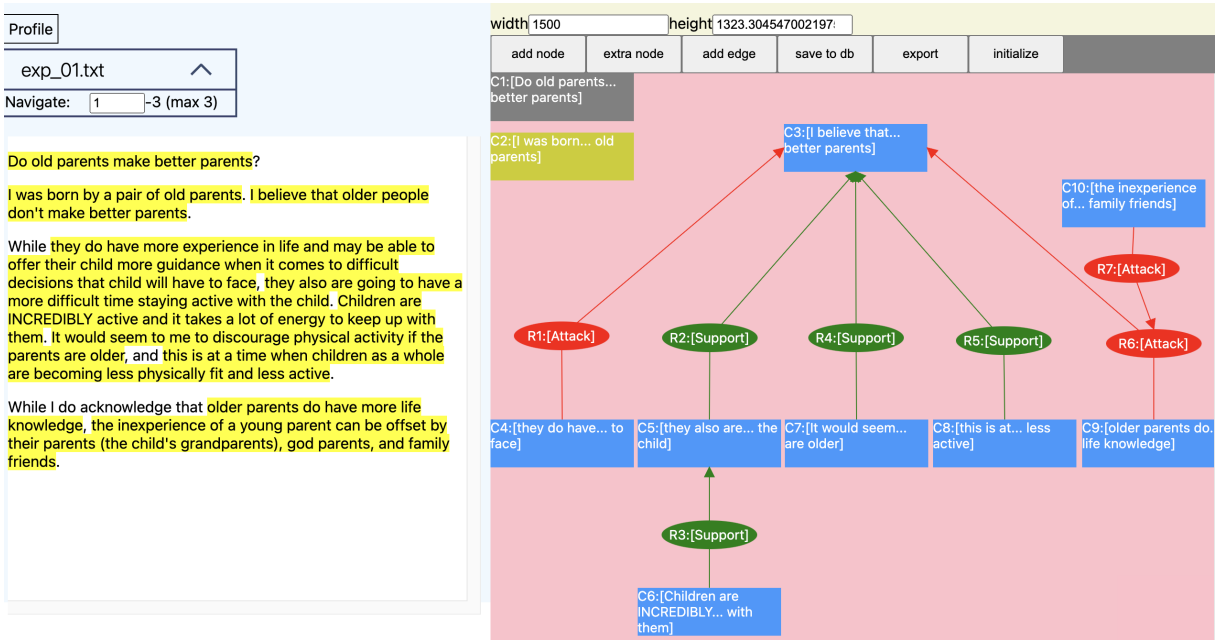


Figure 4.6: An example of an annotated document in the annotation tool interface.

nodes represent segments: grey = unlabelled (non-argumentative), blue = PROPOSITION, olive = STAKE. Oval nodes represent relations: green = SUPPORT, red = ATTACK. Note that in this figure, stand-alone nodes are non-argumentative segments (shown as a grey box) and a STAKE component (shown as an olive box).

I then hired two annotators and trained them to annotate the Quora dataset. One annotator is a researcher in NLP, and the other in linguistics. While the annotators are not native English, they have lived in the UK for several years and are very familiar with reading and writing scientific papers in English as part of their day-to-day work.

Before the training session, the annotators were asked to read through the annotation guideline. The training session took place as a video conference with three participants (myself and the annotators), where I explained my scheme and the annotation procedure using 27 slides. I also demonstrated the annotation of a sample Quora document using a web-based annotation tool. This training session lasted approximately 50 minutes and was recorded for reproducibility. Immediately following the training, I asked each annotator to annotate two identical sample documents from Quora. After reviewing their work, I offered feedback on any errors, aiming to clarify potential misunderstandings regarding the guideline.

Once these preliminary steps were completed, the annotators commenced the official annotation of the Quora dataset, proceeding as follows:

1. Three documents from each of the 20 topics were randomly selected from the Quora dataset. Both annotators annotated the 60 documents, which served as the basis for my annotation study in Section 4.5.

2. The remaining 340 documents were randomly split into two equal subsets. Each subset was assigned to one annotator for annotation. To compile the final collection of 400 annotated documents, I randomly selected one of the doubly annotated documents from the first step and combined it with the 340 documents from this step.

3. Upon completing the first two steps, I initiated a quality control process for the final dataset. In order to identify any violations of the annotation scheme, an automated check was performed on each annotated document in the set. Documents with invalid annotations were sent back to the respective annotator for corrections, with the invalid parts explicitly stated. This process was reiterated until all annotations met the validity criteria.

Overall, this annotation process took about 80 hours for each annotator.

4.4.3 Dataset statistics

The Quora dataset as distributed is randomly divided into three subsets: 280 documents for training, 40 for development, and 80 for testing. Table 4.1 shows the statistics of the Quora dataset. The dataset contains a total of 118,573 tokens, which is comparable in size to the Persuasive Essays dataset (147,271 tokens), but much larger than the Microtext dataset (7,846 tokens) and the Gold Standard Toulmin dataset (84,673 tokens). There are over 7,800 segments in the dataset, and approximately 56% of them are argumentative.

Table 4.2 shows the distribution of components and relations. Looking at the distribution of components, the Quora dataset includes over 4,000 PROPOSITIONs, which are the core building blocks of arguments in the dataset. There are approximately 200 instances of ANALOGY and 79 instances of ANECDOTEs in the dataset, which may provide useful information on how analogies and anecdotes are used in argumentation on Quora. The instances of STAKE are relatively rare, with only 28 instances in the dataset. This small number may cause some problems for quantitative analysis or machine learning algorithms. Therefore, in my experiments on the Quora dataset in Sections 5.4 and 6.2, I will merge all instances of STAKE and ANECDOTE and re-label them as STAKE+ANECDOTE. I made this decision for two reasons. On the one hand, there are only three STAKEs in the test set, which may not have too much influence on the overall results of the experiments. But if I report the results on the three STAKEs, the variance caused by the extremely small test sample may lead to imprecise results. I assume the influence on the overall results is less detrimental compared to the uninterpretability of the results on STAKEs. On the other hand, I chose ANECDOTE as the category to merge STAKE with because STAKEs and ANECDOTEs share some similarities, as they both provide information about the authors, either about their personal background or personal experiences. However, when releasing the Quora dataset, I have preserved these two original categories, STAKE and ANECDOTE, in the hope that future studies may find them useful.

In terms of relations, the Quora dataset includes 2,752 instances of SUPPORT and 1,190 instances of ATTACK. There are 326 undercuts in the dataset, which constitutes approximately 27% of all ATTACKs. This result confirms the importance of undercuts in Quora arguments.

	All	Per document
Token	118,573	296.4
Segment	7,883	19.7
Component	4,381	11.0
Sentence	6,398	16.0
Paragraph	2,826	7.1

Table 4.1: Statistics of the Quora dataset.

		All	Per document
Component	Proposition	4,075 (51.7%)	10.2
	Stake	28 (0.4%)	0.1
	Anecdote	79 (1.0%)	0.2
	Analogy	199 (2.5%)	0.5
	Total	4,381	11.0
	Non-argumentative	3,502 (44.4%)	8.8
Relation	Support	2,752 (69.8%)	6.9
	Attack	1,190 (30.2%)	3.0
	Normal attack	864 (72.6%)	2.2
	Undercut	326 (27.4%)	0.8
	Total	3,942	9.9

Table 4.2: Component and relation distribution of the Quora dataset.

I manually counted the number of undercuts in the Quora dataset under each category from Section 3.4.2. Figure 4.7 shows this distribution. The most frequent type is "Low importance", which constitutes about 41% of all undercuts, followed by "Alternative option" (36%), "Rejection"(15%), and others (8%) — those cannot be categorised into any of those previously mentioned types. According to this result, we can see that even when using undercuts rather than direct Attacks, authors on Quora prefer the less direct methods over straight rejection, which is relatively rare at only 15% of all cases. Instead, when undercutting a relation, they are more likely to point out the weaknesses of that relation, or to provide alternative reasons or options.

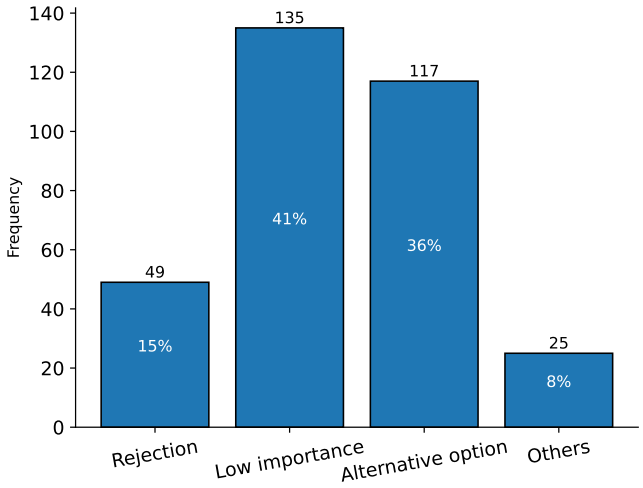


Figure 4.7: Distribution of undercuts in the Quora dataset.

4.5 Annotation study

To evaluate the quality and consistency of the annotation in the Quora dataset, I conducted an annotation study. The study involved the doubly annotated 60 documents previously mentioned. I used the original version of the dataset with four categories, *i.e.* STAKEs and ANECDOTEs are distinguished.

4.5.1 Evaluation metrics for inter-annotator agreement

If an annotation scheme is reliable, we should be able to observe high inter-annotator agreement. In the annotation study for the Quora dataset, I measure inter-annotator agreement for components and relations separately, following most existing studies on annotating argument mining datasets.

For components, frequently used measures include observed agreement, Cohen's kappa (Cohen, 1960), Fleiss' kappa (Fleiss, 1971), and Krippendorff's alpha (Krippendorff, 2018). These measures have been used in many argument mining related annotation tasks, such as in Peldszus and Stede (2015a), Habernal and Gurevych (2017), and Stab and Gurevych (2017). I chose to report only two of them, namely observed agreement and Krippendorff's alpha. Conceptually, Cohen's kappa, Fleiss' kappa and Krippendorff's alpha are similar. They all measure the extent to which the observed agreement exceeds the expected agreement purely by chance. But mathematically, kappa coefficients may overestimate agreement when annotators have "unequal preferences for coding categories" (Krippendorff, 2018). So I abandoned kappa measures and only report Krippendorff's alpha among the chance-corrected measures. When calculating these metrics for component classification, each segment is regarded as a markable item.

Kappa and alpha measures can only be applied when each markable item is independent of each other (Cohen, 1960, Fleiss, 1971, Krippendorff, 2018). Component classification meets this requirement well, but for relation identification the independence assumption does not fully hold. When calculating kappa and alpha metrics for relation identification, each possible component-component pair is usually regarded as a markable item. This approach is problematic because there could be dependencies between annotations of component-component pairs. For instance, if a source component is already linked to a target, it can no longer be linked to other targets. As a result, in the calculation of kappas and alphas, there will be many markable items where no relation exists. Such "unrelated" markable items will lead to an overestimation of agreement. For example, in an argument containing N components, for a source component, even if two annotators disagree on its target components, they still agree on the remaining $N-3$ possible component-component pairs containing this source component. Despite this, several argument mining studies have applied kappa and alpha measures to relations, including Peldszus and Stede (2015a)

and Stab and Gurevych (2017). I refrained from doing so in my annotation study.

Therefore, for relations, I only report observed agreement. A markable item for this measure is a triple consisting of a source segment, its target, and the relation between them. For a source segment having no target, its target and the relation are both marked as $\emptyset$. Agreement is reached only when all three elements in a relation are identical for both annotators, including the source and the target, their labels, and the label of the relation. The total number of markable items for an argument equals the number of the union of markable items by the two annotators, instead of all possible component-component and component-relation pairs in that argument. This is much preferable as it does not have the problem of excessively many unrelated cases and thus should avoid overestimation of agreement.

I also use another agreement measure when reporting the results for binary relation identification which is less strict. I need this metric because I wanted to measure the confusion between the three distinct relation types: SUPPORT, direct ATTACK, and undercut. It is also necessary to measure the confusion that arises when one annotator identifies one of these three relations and the other does not (*i.e.* "unrelated"). For a source segment, agreement is reached when 1) the relation is labelled identically by the two annotators, and 2) the type of its target chosen by the two annotators is the same, either a component or a relation, or $\emptyset$. Criterion 1 distinguishes between SUPPORTs, ATTACKs, and "unrelated" cases. Criterion 2 tells direct ATTACKs and undercuts apart. For example, if a relation is labelled as an ATTACK by both annotators, but one annotator puts it in a component-component pair while the other in a component-relation pair, this agreement measure distinguishes between these two ATTACKs by counting one as a direct ATTACK and the other as an undercut.

To add more perspectives to the measurement of agreement in terms of relations, I use four graph-based metrics to measure the structural similarity between two argument graphs, ignoring all component and relation labels. These metrics assess the holistic agreement on relations across two graphs, rather than focusing on individual relations in isolation. The first graph-based metric is the one proposed by Kirschner et al. (2015). It measures the extent to which one graph is included in the other, doing so in both directions. Equation 4.1 shows the calculation of $I_{A \subset B}$, the extent to which graph A is included in graph B.

$$I_{A \subset B} = \frac{1}{|E_A|} \sum_{(x,y) \in E_A} \frac{1}{SP_B(x,y)} \tag{4.1}$$

In this equation, E_A is the set of edges in A, x is the source node, y is the target node for an edge, and $SP_B(x,y)$ is the length of the shortest path from x to y in B. When

adapting this metric to graphs containing undercuts, I declare a path from an undercutting component to the target component in the component-component pair (length = 0.5), and no path between that undercutting component to the source component in the component-component pair.

$I_{B \subset A}$ is computed analogously. According to the equation, this metric detects if an edge in one graph also exists in another graph, either as an edge or as a path with multiple hops. The final similarity score between A and B can be either the mean of $I_{A \subset B}$ and $I_{B \subset A}$, or their F_1 score. In this annotation study I report both versions, referring to them as Kirschner$^{\text{mean}}$ and Kirschner$^{\text{f}}$.

Other graph-based metrics I use have been proposed by Putra et al. (2022). Compared to Kirschner et al.'s, their metrics are more comprehensive in that they evaluate the mutual inclusiveness between two graphs from different dimensions, namely links (*i.e.* edges), paths, and sets of descendant nodes. The corresponding metrics are referred to as MAR$^{\text{link}}$, MAR$^{\text{path}}$, and MAR$^{\text{dSet}}$ respectively. The calculation of these metrics are as follows:

$$\text{MAR}^{\text{link}} = \frac{1}{2}\left(\frac{|E_A \cap E_B|}{|E_A|} + \frac{|E_A \cap E_B|}{|E_B|}\right) \tag{4.2}$$

$$\text{MAR}^{\text{path}} = \frac{1}{2}\left(\frac{|P_A \cap P_B|}{|P_A|} + \frac{|P_A \cap P_B|}{|P_B|}\right) \tag{4.3}$$

$$\text{MAR}^{\text{dSet}} = \frac{1}{2}\left(\frac{\sum_{x \in N_A} \frac{|dSet^A(x) \cap dSet^B(x)|}{|dSet^B(x)|}}{|N_B|} + \frac{\sum_{x \in N_B} \frac{|dSet^A(x) \cap dSet^B(x)|}{|dSet^A(x)|}}{|N_A|}\right) \tag{4.4}$$

In Equation 4.2, E_A and E_B are sets of edges in graph A and graph B. In Equation 4.3, P_A and P_B are sets of paths in A and B. A path here is represented as an ordered tuple consisting of nodes on the path. In Equation 4.4, N_A and N_B are sets of nodes in A and B (including stand-alone nodes), and $dSet^A(x)$ denotes the set of descendant nodes of the node x in A. When adapting these metrics to graphs containing undercuts, I treat undercuts as nodes. Stand-alone nodes, which usually represent non-argumentative segments, do not affect MAR$^{\text{link}}$ or MAR$^{\text{path}}$, but they do affect MAR$^{\text{dSet}}$. In the calculation of MAR$^{\text{dSet}}$, stand-alone nodes are counted as a match if they are deemed non-argumentative in both annotations. In my case, stand-alone nodes also include STAKEs.

Since Kirschner et al.'s and Putra et al.'s' measures are relatively new, I wanted to provide a reasonable reference for interpreting their results. For this purpose, I compare these metrics against a tailored baseline that I devised. In this context, a baseline system is a method that automatically generates argument graphs, omitting labels of either components or relations. A simple approach might be to draw random links between any two segments within an argument. I did not choose it because I consider it a rather weak

baseline system. My method is more sophisticated in that it always generates valid graph structures that align with my annotation scheme, and it allows for undercuts. Further more, it incorporates the distributions of component and relation categories from the annotation study. Given a pre-segmented argument consisting of s segments, my baseline system generates a graph in the following way:

1. Mirroring the proportion of stand-alone nodes amongst all nodes from the annotation study, the system first randomly selects $s \times \frac{|NA|+|STA|}{S}$ segments from the argument and add them to the graph as stand-alone nodes. $|NA|$ is the number of non-argumentative segments in all arguments in the annotation study. $|STA|$ is the number of STAKEs. S is the total amount of segments.

2. The system then uses the remaining segments in the argument to generate a tree. It starts by randomly choosing one segment from the remaining segments as the root of the tree. From the root, the tree is iteratively constructed by appending random segments as child nodes to each parent node, progressing in a breadth-first manner. The number of child nodes generated at each step is decided by a random process where an integer is sampled from a normal distribution $\mathcal{N}(\mu, \sigma)$, bound between 1 and the amount of segments currently available. This distribution is also informed by the observed annotation. The tree is finished when when no segments are left.

3. Mirroring the proportion of undercuts within all relations, the system randomly selects $r \times \frac{|UC|}{R}$ tree edges from the prior step and transforms them into undercuts. Here r is the number of edges in the tree. $|UC|$ is the number of undercuts in all arguments. R is the number of relations. To transform a selected edge which connects a child node and a parent node into an undercut, the system reroutes the edge from the child to the parent's outgoing edge. The only exception arises when the parent is the root; here, the edge remains unaltered.

With this strong baseline system, I can generate a distinct version of annotation for each annotator. The Kirschner and MAR scores derived from this baseline can then be used as a lower bound to compare against the actually observed scores. I ran the baseline system ten times and report the average scores.

4.5.2 Results and disagreement analysis

The overall inter-annotator agreement scores are presented in Table 4.3. For component classification, the observed raw agreement is $P(A) = 0.88$. The score of Krippendorff's alpha is $\alpha = 0.78$ ($N = 7,883, n = 2, k = 5$). According to the interpretation scale in Krippendorff (2018), this score is acceptable for "drawing tentative conclusions" ($\alpha \geq 0.67$), and is close to the threshold ($\alpha \geq 0.80$) for being considered "reliable".

		Human	Baseline
Component	Observed	0.88	-
	Krippendorff's alpha	0.78	-
Relation	Observed	0.73	
	$\text{Kirschner}^{\text{mean}}$	0.69	0.04
	$\text{Kirschner}^{\text{f}}$	0.67	0.03
	MAR^{link}	0.64	0.02
	MAR^{path}	0.54	0.01
	MAR^{dSet}	0.84	0.41

Table 4.3: Inter-annotator agreement for components and relations.

For relation identification, the observed agreement is $P(A) = 0.73$. In terms of the graph-based measures, scores achieved by the baseline are close to zero, except $\text{MAR}^{\text{dSet}} = 0.41$. The relatively high MAR^{dSet} score for the baseline might be a result of the measure's inherent nature, because stand-alone nodes agreed by both annotators always positively contribute to MAR^{dSet}. Human agreement easily beats the baseline ($p < 0.01$ in all paired t-tests, in which agreement for each document is regarded as a sample): $\text{Kirschner}^{\text{mean}} = 0.69$, $\text{Kirschner}^{\text{f}} = 0.67$, $\text{MAR}^{\text{link}} = 0.64$, $\text{MAR}^{\text{path}} = 0.54$, and $\text{MAR}^{\text{dSet}} = 0.84$. The same graph-based metrics are also reported by Putra et al. (2022) in their annotation work: $\text{Kirschner}^{\text{mean}} = 0.63$, $\text{Kirschner}^{\text{f}} = 0.63$, $\text{MAR}^{\text{link}} = 0.56$, $\text{MAR}^{\text{path}} = 0.39$, and $\text{MAR}^{\text{dSet}} = 0.85$, but of course no comparison against my scores is possible because of different tasks, schemes, and data.

In order to examine to which degree annotators agree per component category, I calculate inter-annotator agreement on binary classification per category and present the alpha scores ($N = 7,883, n = 2, k = 2$) in Table 4.4. We can see that the two annotators completely agree on STAKE and ANECDOTE, which is a remarkable item result. This could be because components under these two categories are characteristic enough to be easily distinguished from other categories. Among the remaining component categories, agreement on recognising ANALOGYs ($\alpha = 0.82$) is slightly higher than that on non-argumentative segments ($\alpha = 0.77$) and PROPOSITIONs ($\alpha = 0.77$).

I do not report observed agreement for binary classification per category. Since observed agreement is not chance-corrected, it is largely effected by the distribution of categories. For example, if the number of instances in a category is extremely small, the number of true negatives is likely to be large, which will lead to a high observed agreement for that

	Non-arg	Proposition	Stake	Anecdote	Analogy
Krippendorff's alpha	0.77	0.77	1.00	1.00	0.82

Table 4.4: Inter-annotator agreement for binary component classification.

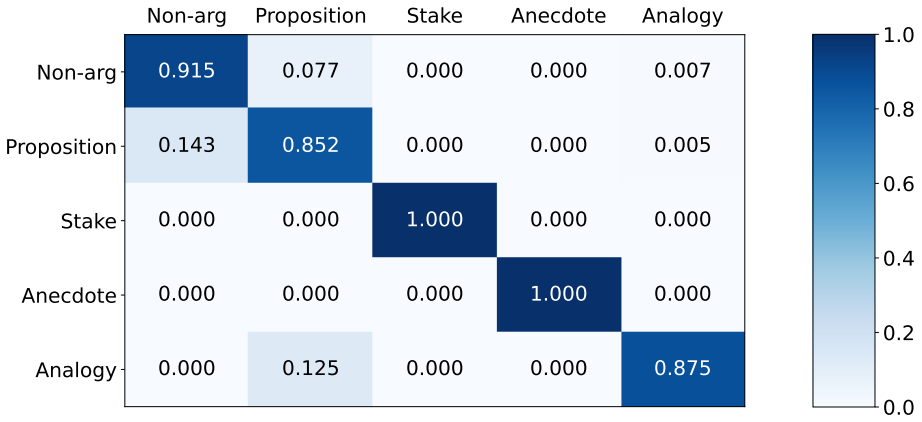


Figure 4.8: The confusion matrix for component classification, normalised by row.

category.

I present the confusion matrix for component classification in Figure 4.8. In the matrix, the number on the i-th row and j-th column is the amount of instances (normalised by row) that is classified as the i-th component category by one annotator while the j-th category by the other. Results show that the two categories most frequently confused with each other are non-argumentative segments and PROPOSITIONs. This is reflected by the relatively high off-diagonal values in the corresponding cells (0.143 and 0.077). One reason for this confusion might be the difficulty in labelling components containing rhetorical questions. The annotation guideline requires that rhetorical questions should be considered argumentative only if an explicit answer is present in the text, or if the answer to that question is obvious enough for most readers to reach an agreement (*e.g.* "Is the Earth flat?"). However, this criterion is subjective and proved to be a problem.

Example 4.9
Question:
Should animal testing be banned?
Answer:
[...] [What is to be learned from the body of animals, except for how the animals work]$_{PRP}$? [What is to be learned from humans, except for how humans work]$_{PRP}$? [...]

In Example 4.9, for instance, one annotator considers the two rhetorical questions argumentative because they believe that the answers are obvious — what are learned from animals cannot be applied to humans, so that animal testing is meaningless. However, the other annotator regards them as information-seeking questions and categorises them as non-argumentative. This provides another example of the challenges of argument mining

with informal text in online discussions, where irony, rhetorical devices, and ellipsis are frequent.

The confusion between PROPOSITION and ANALOGY is also relatively high (0.125 and 0.005). One possible reason for this confusion is that sometimes the key entity being discussed in an ANALOGY is too semantically close to that in the argument. Recognising the distinction between the two key entities is crucial for identifying ANALOGYs. As a result, failing to notice this distinction can lead to missing ANALOGYs. For instance, in Example 4.10, one annotator correctly recognises the component as an ANALOGY, while the other annotator labels it as a PROPOSITION, as the latter thinks that this component is still talking about healthcare because it is also a kind of "product".

Example 4.10
Question:
Would a completely free market healthcare system with zero government involvement and no subsidies for anyone work better in the long run?
Answer:
[...] [Just like with shoes or kitchen cutlery sets or televisions, I can evaluate how much my money is worth to me, how much the product is worth to me, and then make a choice to buy or not buy]$_{ANA}$. [...]

Moving on to relation identification, I present the confusion matrix in Figure 4.9. Results show that the "Unrelated" instances are frequently confused with SUPPORTs (0.137 and 0.056) and direct ATTACKs (0.158 and 0.013). Most cases of such confusion are due to a prior disagreement in component classification. If a segment is considered argumentative by one annotator but non-argumentative by the other, confusion with "Unrelated" in relation identification necessarily follows.

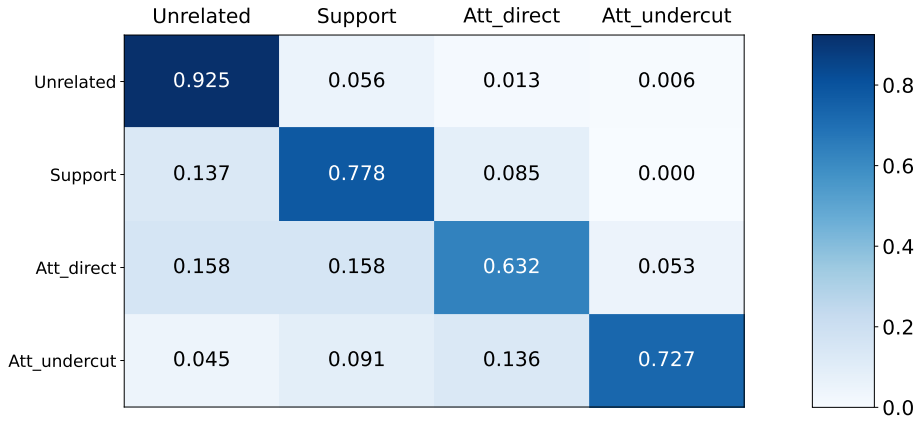


Figure 4.9: The confusion matrix for relation identification, normalised by row.

Among the remaining relation categories, the confusion between SUPPORT and direct ATTACK is the most frequent (0.158 and 0.085). This happens mostly because a component is linked to two different targets that contradict each other, so that this component supports one target and attacks the other (*e.g.* "private schools have much more funding" supports "private schools are better" and attacks "private schools are worse"). For such cases, the guideline instructs annotators to link the source component to the target that requires the least inference. But again, this criterion is subjective and proved to be a problem.

The confusion between direct ATTACK and undercut is slightly lower (0.136 and 0.053). This is good because it suggests that undercut recognition for humans is potentially objectively achievable, despite the nuanced nature of undercuts. When an annotator fails to recognise an undercut, they often incorrectly associate the source component with a target component, rather than a relation. In these cases, an undercut is mistakenly identified as a direct ATTACK. As reported by the annotators, it was difficult for them to consider the entire search space necessary when recognising an undercut, because they needed to consider not only the direct relation between two components, but also the potential existence of an undercut and its impact on the relation between the two components. However, discussions with the annotators during training and the quality control process indicated that they were able to correctly identify most mislabelled undercuts when prompted to "think twice"[1].

In conclusion, results of the annotation study suggest that the annotation of components in the Quora dataset is acceptable for "drawing tentative conclusions", and that of relations is better than chance.

4.6 Chapter summary

In this chapter, I introduced my annotation scheme, which is tailored to informal online arguments. Component categories in this scheme include PROPOSITION, STAKE, ANECDOTE, and ANALOGY. Different from simple claim-premise categorisation, these categories were designed to reflect persuasion patterns in arguments. My annotation scheme includes undercuts, allowing for investigation into how arguments are undermined or weakened.

I then presented the Quora dataset, which was annotated with this scheme. It consists of a collection of 400 informal arguments collected from online discussions. It is by far the biggest dataset that contains explicitly annotated undercuts. I designed a systematic collection method to achieve good coverage and balance of topics in this dataset. I also trained two annotators with my detailed annotation guideline. The dataset contains 118,573 tokens and over 7,800 segments, and approximately 56% of them are argumentative. Among

[1]Please note that during the annotation study, all agreement was measured on annotation that was performed independently. No such discussion or prompting took place.

the component categories, STAKE is relatively small in number, making it challenging to conduct statistical analysis. When it comes to argument relations, out of the 1190 Attacks in the dataset, 326 are undercuts. This suggests that undercuts are indeed a frequent counterargument device in online argumentation.

I also conducted an annotation study and showed that the annotation of my dataset is acceptable. For component classification, the inter-annotator agreement is acceptable for "drawing tentative conclusions" at Krippendorff's alpha $\alpha = 0.78$ ($N = 7,883, n = 2, k = 5$). For relation identification, I reported observed agreement and several graph-based metrics, namely the Kirschner and MAR metrics. On these graph-based measures, human annotation substantially beats a strong random baseline.

With the Quora dataset, I now have at my disposal material that can be used for my planned experiments in undercut-aware argument mining. These experiments will be described in the next two chapters.

Chapter 5

End-to-end argument mining as dependency parsing

In this chapter, I present an approach to end-to-end argument mining based on dependency parsing. I name my approach as AMDP (Argument Mining as Dependency Parsing). Eger et al. (2017) and Morio et al. (2020) are the first to formulate argument mining as dependency parsing. However, Eger et al.'s implementation of such an approach does not yield satisfactory results. The approach by Morio et al. (2020), on the other hand, is not a full end-to-end approach because it assumes that the input text is already segmented. In contrast, my approach achieves a new state of the art on a benchmark argument mining dataset and is truly end-to-end.

I create two versions of dependency representations (Section 5.1), one that does not model undercuts, and one that does. To the best of my knowledge, I am the first to model undercuts computationally. Undercuts appear in the Microtext dataset, but they have never been processed by argument mining models. Even in the study by Peldszus and Stede (2015b), who are also creators of the Microtext dataset (3.1.3) that includes explicitly annotated undercuts, undercuts are not treated in any way by their argument mining models. I suspect this could be due to their structure, being a relation between a component and anther relation. This is very different from the traditional structure of relations in argument mining, which is between two components. Through a redesign of existing dependency representations of arguments, I am able to directly use existing neural dependency parsers to computationally model undercuts.

I introduce two neural models as dependency parsers: a biaffine model in Section 5.2.1 and a model based on Graph Neural Networks (GNN) (Veličković et al., 2017) in Section 5.2.2.

I also evaluate AMDP on two datasets: firstly a popular argument mining dataset (Section 5.3), and secondly my newly introduced Quora dataset (Section 5.4).

5.1 Dependency representations for arguments

Many annotation schemes represent arguments as trees or directed acyclic graphs (Toulmin, 1958, Peldszus and Stede, 2013, Habernal and Gurevych, 2017, Visser et al., 2020). Such structures bear resemblance to those used in syntactic and semantic parsing, which can be analysed using dependency parsers (Dozat and Manning, 2017, 2018, Qi et al., 2018). However, unlike syntactic or semantic parsing, which operates at the token-level (Figure 5.1a), the dependency structures in argument mining operate at the segment-level (Figure 5.1b). Therefore, in order to utilise the current dependency parsing techniques for argument mining, it is necessary to transform the dependencies at the segment-level into those at the token-level.

I have created two versions of dependency representation for arguments, one that does not represent undercuts, and one that does. I will first introduce the undercut-exclusive representation here. The undercut-exclusive representation contains information about segment boundaries and types of relations that could potentially hold between segments, as illustrated in Figure 5.2. Please note the difference in arrow conventions between argument mining and dependency parsing. This figure shows dependencies pointing from

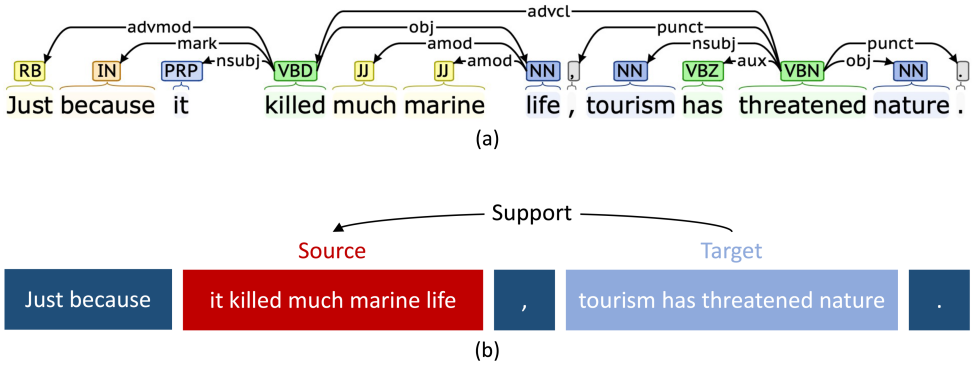


Figure 5.1: The dependency structure of the sentence "Just because it killed much marine life, tourism has threatened nature." in (a) syntactic parsing and (b) argument mining. The syntactic parse tree is produced using Stanford CoreNLP (Manning et al., 2014).

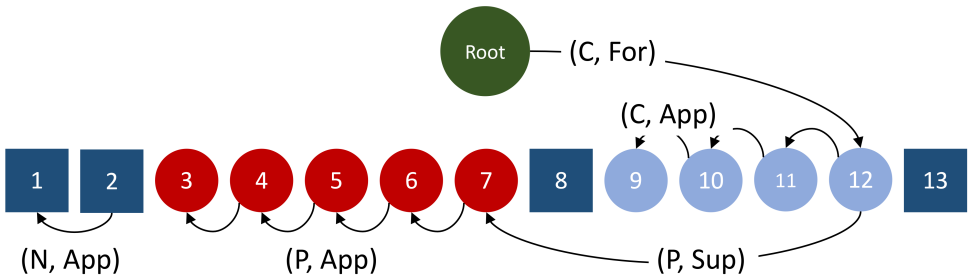


Figure 5.2: My undercut-exclusive dependency representation of the argument "Just because it killed much marine life, tourism has threatened nature." C = Claim, P = Premise, Sup = Support, N = non-argumentative, App = append.

head to dependant, as is common in dependency parsing. In contrast, in argument mining, relations are shown in the opposite direction (*i.e.* pointing from source to target), as can be seen in Figure 1.2. In the rest of this paper, A $\rightarrow$ B means A pointing to B in terms of argument structures, while A $\Leftarrow$ B means B pointing to A with B being the head of A in terms of dependency structures. Please also note that this figure uses the category labels in the Persuasive Essays dataset, which I will later use for evaluation. My dependency representation is not limited to a certain set of pre-defined labels. This flexibility means that it can be adapted to various annotation schemes.

My undercut-exclusive representation is as follows:

- Each token is allowed to have zero, one, or more heads, which makes the representation a dependency graph.

- Information on segment boundaries and segment labels is captured through within-segment labelled edges. Within a segment (*e.g.* tokens 3-7 in Figure 5.2), either argumentative or non-argumentative, each token except the last one is headed by its succeeding token (token 4 $\Leftarrow$ token 5). Labels of within-segment edges are written as ($segment\_label \in [component\_label, N]$, App), where $component\_label$ is the component category (*e.g.* Claim and Premise), N represents "non-argumentative", and App denotes "append". In this way, segment boundaries are encoded topologically — if the label of an edge does not contain App, we know that a segment boundary has occurred. For example, the edge between token 4 and token 5 is labelled as (P, App), indicating that they are two neighbouring tokens within a Premise.

- Information on relations and their labels is captured through inter-segment labelled edges. If a relation exists between two components, it is expressed as a labelled edge between the last token of the head component and the last token of the dependant component. Labels of inter-segment edges are written as ($component\_label$ of dependant node, $relation\_label$), where $relation\_label$ is the relation category (*e.g.* Attack and Support). For instance, the label of the edge between token 7 and token 12 is (P, Sup), which means that token 7 is the last token of a Premise, and this Premise supports the component that includes token 12. The last token in a non-argumentative segment does not have any head node.

- A pseudo-token $ROOT$ is added to the beginning of each argument. In syntactic parsing, a hypothetical root acts as the head of each sentence. Similarly, in my representation the pseudo-token $ROOT$ acts as the head of each argument[1].

[1]Theoretically, the $ROOT$ token could represent the topic or the main point of the entire argument, such as "uniforms" in my multi-argument example in section 3.4.1. I do not model such $ROOT$-represented topics or points in this thesis, but such a design can be a placeholder for future studies.

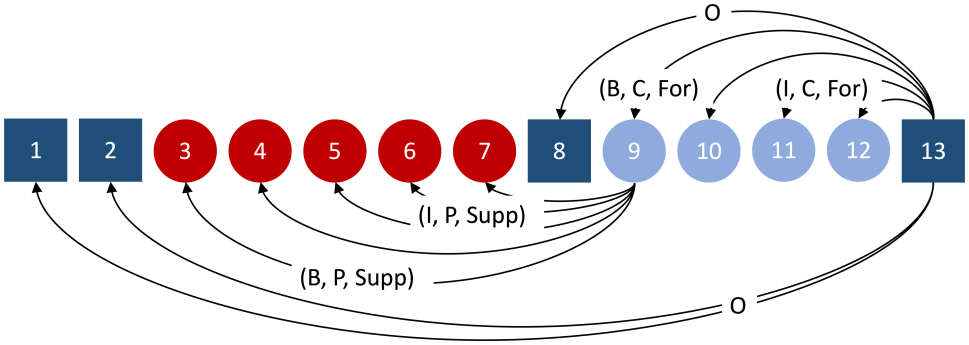


Figure 5.3: The dependency representation in Eger et al. (2017).

My dependency representation differs from the one in Eger et al. (2017), as illustrated in Figure 5.3, in several ways:

- The dependencies among tokens in the same segment are represented differently by Eger et al. and myself. In Eger et al.'s representation, each token within a segment is headed by the first token in its head component. This means that their representation does not directly express within-segment dependencies. In my representation, each token within a segment is headed by its succeeding token within the same component, and only the last token is linked to its head component. As a result, most dependencies in my representation will be within-segment relations instead of inter-segment relations. This difference is important because, according to my linguistic intuition, relations between neighbouring tokens within a segment are more likely to be meaningful than relations between a token and the first token in another component. Therefore, I would expect that my representation should model within-segment and long-distance inter-segment dependencies better.

- Segment boundaries are also encoded differently. Eger et al. (2017) use the BIO scheme to encode segment boundaries, which results in a larger prediction space than the topological encoding in my representation. My encoding method is thus computationally more efficient and might result in better results because segment boundaries can be inferred without BIO tags.

- We also conceptualise the root node of an argument differently. Eger et al. (2017) declare the terminating token in an argument as root. All non-argumentative tokens are also headed by this root. Their conceptualisation of the root node is problematic for two reasons. First, the relations between all non-argumentative segments and the terminating token seem arbitrary. Second, in some annotation schemes, the terminating token could be a part of a component. Linking non-argumentative segments to the terminating token would then spuriously include them into the argument graph. In contrast, my representation uses the pseudo-node $ROOT$ as

the root node, which is outside all textual segments and thus avoids those problems. Additionally, non-argumentative segments are not linked to any other segments in my representation.

My dependency representation in Figure 5.2 cannot be used for arguments containing undercuts, as undercuts involve relations between a component and a relation, and a relation cannot be a dependent or a head in typical dependency representations. Therefore, I have designed a specialised undercut-inclusive dependency representation for arguments that allows existing neural dependency parsers to process undercuts directly.

The undercut-inclusive representation is illustrated in Figure 5.4. It is a modification of the undercut-exclusive representation in Figure 5.2. New features of the undercut-inclusive representation are as follows:

- A relation node (shown as dashed-line nodes in Figure 5.4) for each token in the argument is added. Each relation node is indexed with the token number of its corresponding token, followed by a prime. The relation nodes are meant to represent relations. This is very different from the undercut-exclusive representation, where relations are represented by edges.

- The relation node (*e.g.* relation node $7'$) of the last token (token 7) in a component is always the head of that token, and represents the relation from that component to its target, or the other way around. The edge label between the last token in a component and its corresponding relation node is written as (*segment_label*, REL),

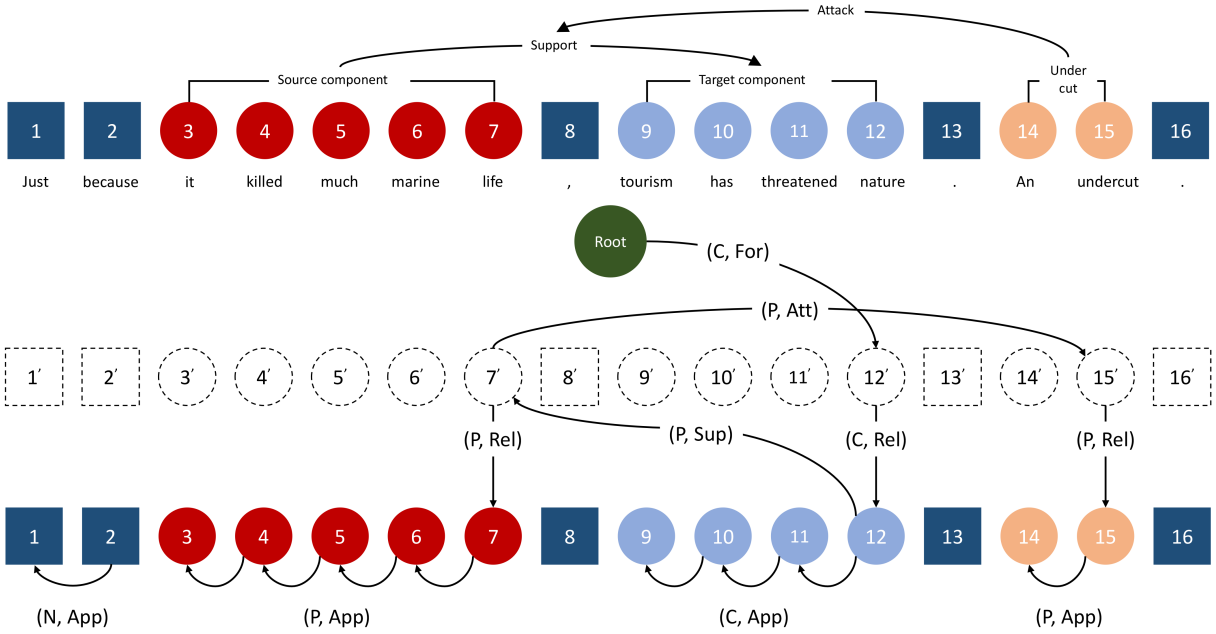


Figure 5.4: The structure of an example argument with a pseudo undercutting component (written as "An undercut"), and its undercut-inclusive dependency representation.

where REL means "relation". For example, the label of the edge between token 7 and relation node $7'$ is (P, REL).

- If the relation is a SUPPORT or a direct ATTACK, the relation node's outgoing edge points to the last token in the source component. Its incoming edge comes from the last token of the target component. For example, the fact that "it killed much marine life" (tokens 3-7) supports "tourism has threatened nature" (tokens 9-12) is expressed by the incoming edge of relation node $7'$ from token 12, and the outgoing edge of relation node $7'$ to node 7. This is also written as "tokens 9-12 $\Rightarrow$ relation node $7'$ $\Rightarrow$ tokens 3-7".

- If the relation is an undercut, the relation node's outgoing edge points to the last token in the undercutting component. The incoming edge comes from another relation node, rather than the last token in a component. For example, the fact that tokens 14-15 undercuts the relation between tokens 3-7 and tokens 9-12 is expressed by the incoming edge of relation node $15'$ from relation node $7'$, and the outgoing edge of relation node $15'$ to node 15. This is also written as "relation node $7'$ $\Rightarrow$ relation node $15'$ $\Rightarrow$ tokens 14-15".

I believe that a crucial insight of my undercut-inclusive representation is its treatment of all relations as nodes. This design allows undercuts to be modelled as relations between nodes, all the while preserving the unique status of undercuts via the mechanism of "relation nodes". In this way, existing neural dependency parsers are able to process undercuts directly. Peldszus and Stede (2015b) chose a different path: they transformed undercuts into direct attacks, redirecting the undercutting component to the target component in the component-component pair. Their method makes undercuts indistinguishable from direct attacks, the nature of undercuts therefore remains untreatable. Thus, my method enables true undercut processing for the first time.

In Sections 5.3 and 5.4, I will experimentally test whether my representations can indeed improve argument mining.

5.2 Neural models for argument mining

In this section, I describe the two modified neural dependency parsers I use, a biaffine parser and a GNN-based parser. The biaffine parser is designed to learn direct dependencies between the tokens in the input text, while the GNN-based parser is expected to capture more complex dependencies by modelling the relations between the nodes in the graph structure. I have adapted both parsers to incorporate my dependency representations for arguments.

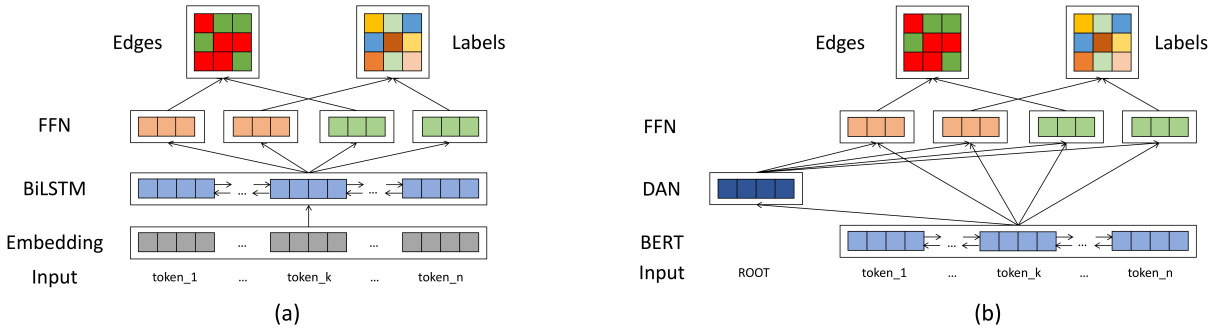


Figure 5.5: (a) the biaffine parser in Dozat and Manning (2018), and (b) my modified biaffine parser. FFN = Feedforward Network. DAN = Deep Averaging Network.

5.2.1 Biaffine dependency parser

The modified biaffine dependency parser is based on the parser proposed by Dozat and Manning (2018) (Figure 5.5a), which has been adopted in many studies (Liu et al., 2019a, Zhang et al., 2019). It is composed of a BiLSTM layer that encodes the input text and a deep biaffine attention classifier that scores and labels each possible head-dependent pair.

The structure of my biaffine parser is shown in Figure 5.5b. The mathematical description of this parser is as follows:

$$r_S = \text{BERT}(s_1 s_2 ... s_n) \tag{5.1}$$

$$r_{ROOT} = \text{FFN}^{\text{ROOT}}(\text{mean}(r_S), \text{axis}=0) \tag{5.2}$$

$$R = [r_S; r_{ROOT}], \text{axis}=0 \tag{5.3}$$

$$H^{\text{e\_h}}, H^{\text{l\_h}}, H^{\text{e\_d}}, H^{\text{l\_d}} = \text{FFN}(R) \tag{5.4}$$

$$H_{\text{r}}^{\text{e\_h}}, H_{\text{r}}^{\text{l\_h}}, H_{\text{r}}^{\text{e\_d}}, H_{\text{r}}^{\text{l\_d}} = \begin{cases} \text{FFN}^{\text{rel\_node}}(R) & \text{if undercut\_inclusive} \\ \emptyset & \text{if undercut\_exclusive} \end{cases} \tag{5.5}$$

$$\text{Biaff}(x, y) = x^{\top} \text{U} y + \text{W}(x \oplus y) + b \tag{5.6}$$

$$sc^{\text{edge}} = \text{Biaff}^{\text{edge}}(([H^{\text{e\_h}}; H_{\text{r}}^{\text{e\_h}}], \text{axis}=0), ([H^{\text{e\_d}}; H_{\text{r}}^{\text{e\_d}}], \text{axis}=0)) \tag{5.7}$$

$$sc^{\text{label}} = \text{Biaff}^{\text{label}}(([H^{\text{l\_h}}; H_{\text{r}}^{\text{l\_h}}], \text{axis}=0), ([H^{\text{l\_d}}; H_{\text{r}}^{\text{l\_d}}], \text{axis}=0)) \tag{5.8}$$

$$y'^{\text{edge}}_{i,j} = \{sc^{\text{edge}}_{i,j} \geq 0\} \tag{5.9}$$

$$y'^{\text{label}}_{i,j} = \arg\max \ sc^{\text{label}}_{i,j} \tag{5.10}$$

$$\mathscr{L} = (1-\lambda)\mathscr{L}^{\text{edge}} + \lambda \mathscr{L}^{\text{label}}, \lambda \in (0, 1) \tag{5.11}$$

I replace the Embedding layer and the BiLSTM layer with a pre-trained BERT encoder (Wolf et al., 2020), so that the parser can benefit from a huge amount of unsupervised

data[2]. Equation 5.1 describes how the encoded representation $r_S \in \mathbb{R}^{n \times d_{enc}}$ of a text sequence $S = s_1 s_2 ... s_n$ is calculated using BERT.

Equation 5.2 shows the calculation of $r_{ROOT} \in \mathbb{R}^{1 \times d_{enc}}$, which is the representation of the root node *ROOT*. *ROOT* is not processed by the BERT encoder. Instead, it is calculated using a Deep Averaging Network (DAN), because it can potentially represent the overall gist of the argument. FFN in Equation 5.2 means a Feedforward Network.

Equation 5.3 shows the calculation of $R \in \mathbb{R}^{(n+1) \times d_{enc}}$, which is the representation of *ROOT* plus S. It is derived by concatenating r_{ROOT} and r_S.

Equation 5.4 creates four representations for R, including two head representations and two dependant representations for edge and label prediction respectively. These are the deep representations for all tokens in *ROOT* plus S.

Equation 5.5 creates four further representations for R in the undercut-inclusive scenario. These are the deep representations for the relation nodes in the undercut-inclusive dependency representation. In the undercut-exclusive scenario, this step is skipped.

The biaffine classifier in Equation 5.6 scores and labels edges between head-dependent pairs, where U, W, and b are trainable variables. The predicted edges and their labels are computed in Equations 5.7, 5.8, 5.9, and 5.10, respectively. The edge-classifier is trained with sigmoid cross-entropy, while the label-classifier is trained with softmax cross-entropy, in the same manner as is done by Dozat and Manning (2018).

The total loss is calculated in Equation 5.11, where λ balances the edge and label losses. The back-propagation of the losses to the label-classifier uses only edges in the gold standard, and not other hypothetical edges between each two tokens in the representation.

Dropout (Srivastava et al., 2014) is applied for each layer in my model. Hyperparameters of the modified biaffine parser are set to the default values in Dozat and Manning (2018).

5.2.2 GNN-based dependency parser

Considering the graph nature of the dependency structure of arguments, GNNs can be seen as a natural choice for argument mining tasks. GNNs have been shown to be effective in modelling graph-structured data and have achieved state-of-the-art performance in many graph-related tasks (Wang et al., 2019, Chen et al., 2020, Minaee et al., 2021). Therefore, I chose to experiment with a modified GNN-based dependency parser. The GNN-based parser I use is based on the dependency parser proposed by Ji et al. (2019) (Figure 5.6a), which adapts the Graph Attention Networks (GANs) (Veličković et al., 2017) to explicitly model higher-order dependencies. Compared to the biaffine parser, I

[2]I abandoned another pre-trained encoder GPT-2 (Radford et al., 2019) because it performed slightly below BERT.

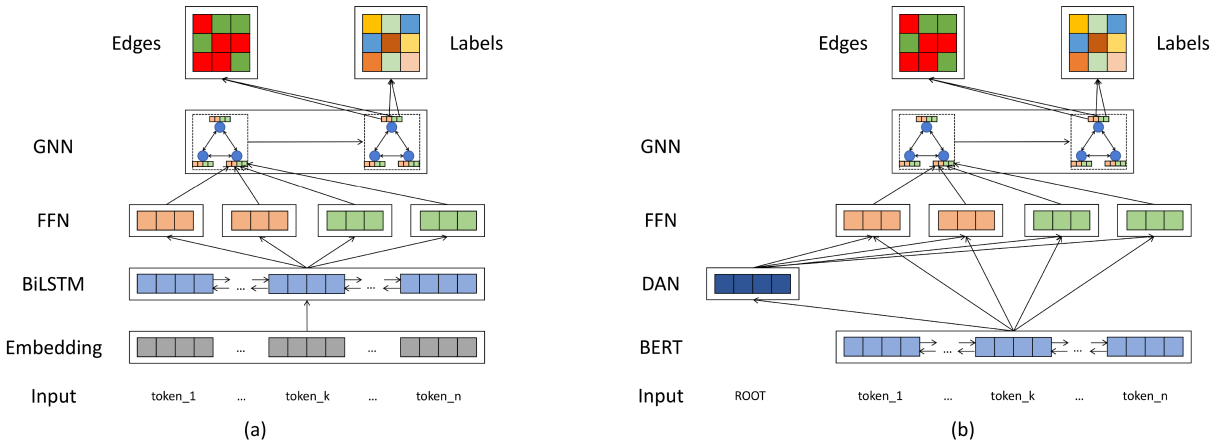


Figure 5.6: (a) the GNN-based parser in Ji et al. (2019), and (b) my modified GNN-based parser.

expect the GNN-based parser to have the advantage of modelling global information of the argument structure and to be better at capturing higher-order dependencies.

The structure of my GNN-based parser is shown in Figure 5.6b. The mathematical description of this parser is as follows:

$$r_S = \text{BERT}(s_1 s_2 ... s_n) \tag{5.12}$$

$$r_{ROOT} = \text{FFN}^{\text{ROOT}}(\text{mean}(r_S), \text{axis}=0) \tag{5.13}$$

$$R = [r_S; r_{ROOT}], \text{axis}=0 \tag{5.14}$$

$$H^{\text{e\_h}}, H^{\text{l\_h}}, H^{\text{e\_d}}, H^{\text{l\_d}} = \text{FFN}(R) \tag{5.15}$$

$$H_{\text{r}}^{\text{e\_h}}, H_{\text{r}}^{\text{l\_h}}, H_{\text{r}}^{\text{e\_d}}, H_{\text{r}}^{\text{l\_d}} = \begin{cases} \text{FFN}^{\text{rel\_node}}(R) & \text{if undercut\_inclusive} \\ \emptyset & \text{if undercut\_exclusive} \end{cases} \tag{5.16}$$

$$\begin{aligned} H_{\text{G}}^{\text{e\_h}}, H_{\text{G}}^{\text{l\_h}}, H_{\text{G}}^{\text{e\_d}}, H_{\text{G}}^{\text{l\_d}} = \text{GNN}_{\text{layer=2}}(([H^{\text{e\_h}}; H^{\text{l\_h}}], \text{axis}=1), ([H^{\text{e\_d}}; H^{\text{l\_d}}], \text{axis}=1), \\ ([H_{\text{r}}^{\text{e\_h}}; H_{\text{r}}^{\text{l\_h}}], \text{axis}=1), ([H_{\text{r}}^{\text{e\_d}}; H_{\text{r}}^{\text{l\_d}}], \text{axis}=1)) \end{aligned} \tag{5.17}$$

$$\text{Biaff}(x, y) = x^{\top} \text{U} y + \text{W}(x \oplus y) + b \tag{5.18}$$

$$sc^{\text{edge}} = \text{Biaff}^{\text{edge}}(H_{\text{G}}^{\text{e\_h}}, H_{\text{G}}^{\text{e\_d}}) \tag{5.19}$$

$$sc^{\text{label}} = \text{Biaff}^{\text{label}}(H_{\text{G}}^{\text{l\_h}}, H_{\text{G}}^{\text{l\_d}}) \tag{5.20}$$

$$y'^{\text{edge}}_{i,j} = \{sc^{\text{edge}}_{i,j} \geq 0\} \tag{5.21}$$

$$y'^{\text{label}}_{i,j} = \arg\max \ sc^{\text{label}}_{i,j} \tag{5.22}$$

The calculation of the four kinds of representation for all tokens in *ROOT* plus S, namely $H^{\text{e\_h}}, H^{\text{l\_h}}, H^{\text{e\_d}}, H^{\text{l\_d}}$, as described in Equations 5.12-5.15, is the same as Equations 5.1-5.4 in the biaffine model described earlier. Equation 5.16 produces the four

kinds of representation for relation nodes when necessary, namely $H_{\mathrm{r}}^{\mathrm{e\_h}}, H_{\mathrm{r}}^{\mathrm{l\_h}}, H_{\mathrm{r}}^{\mathrm{e\_d}}, H_{\mathrm{r}}^{\mathrm{l\_d}}$, which is equivalent to Equation 5.5 in the biaffine model.

In Equation 5.17, the eight kinds of representation above are fed into a GNN encoder consisting of two GNN layers, converting them into GNN-encoded representations. Inside the GNN layers, each token in the input sequence, as well as *ROOT*, is represented as a node. In the undercut-inclusive scenario, each relation node is also represented as a node inside the GNN layers. Such a node connects bidirectionally to all other nodes, so that a fully connected graph is formed. The head representation and the dependant representation of each node are concatenated to form a general representation. The general representation of each node is then aggregated and updated through the GNN layers.

$$\alpha_{ij}^{t} = \mathrm{Attention}(h_i^{t-1}, d_j^{t-1}) \tag{5.23}$$

$$\alpha_{ji}^{t} = \mathrm{Attention}(d_i^{t-1}, h_j^{t-1}) \tag{5.24}$$

$$h_i^t = \mathrm{W}^{\mathrm{head}} \sum_{j \in \mathcal{N}(i)} (\alpha_{ji}^t h_j^{t-1} + \alpha_{ij}^t d_j^{t-1}) + \mathrm{b}^{\mathrm{head}} h_i^{t-1} \tag{5.25}$$

$$d_i^t = \mathrm{W}^{\mathrm{dependant}} \sum_{j \in \mathcal{N}(i)} (\alpha_{ij}^t h_j^{t-1} + \alpha_{ji}^t d_j^{t-1}) + \mathrm{b}^{\mathrm{dependant}} d_i^{t-1} \tag{5.26}$$

Equations 5.23-5.26 show what happens inside Equation 5.17. Equations 5.25 and 5.26 show the aggregation and updating process for the t-th GNN layer. Here, $\mathcal{N}(i)$ denotes the set of neighbours of node i, and W and b are trainable variables. The attention mechanism for aggregating node representations is described in Equations 5.23 and 5.24. Here, α_{ij}^t denotes the attention score between the head-wise representation of node i and the dependant-wise representation of its neighbour j, and α_{ji}^t denotes the attention score between the dependant-wise representation of node i and the head-wise representation of its neighbour j. This aggregation and update mechanism explicitly models higher-order dependencies, namely grandparents, grandchildren, and siblings, allowing the GNN-based parser to capture more complex argument structures.

The GNN-encoded representations are then used for edge and label prediction, as described in Equations 5.18, 5.19, 5.20, 5.21, and 5.22, in the same manner as in the biaffine model.

The output of each GNN layer can be interpreted as a soft parse graph (*i.e.* a fully connected graph with weighted edges) without label information. The structural loss of each GNN layer is incorporated into the final training loss, as shown in Equation 5.27.

$$\mathscr{L} = \lambda_1 \mathscr{L}^{\mathrm{edge}} + \lambda_2 \mathscr{L}^{\mathrm{label}} + (1 - \lambda_1 - \lambda_2) \sum_{t=1}^{t=2} \mathscr{L}_{\mathrm{GNN}}^{t}, \lambda_1 \in (0, 0.5), \lambda_2 \in (0, 0.5) \tag{5.27}$$

The same dropout strategy in the biaffine model is applied for each layer in my model. Hyperparameters of the modified GNN-based parser are set to the default values given by Ji et al. (2019).

Now that I have explained how my dependency representations and neural parsers work, I will proceed to their evaluation.

5.3 Experiment: argument mining without undercut modelling

In this experiment, I evaluate the performance of my end-to-end approach on a popular benchmark dataset that does not include undercuts. The objective is to assess the effectiveness of the approach in comparison to existing approaches, and to experimentally quantify the effects of my representations and parsers.

5.3.1 Dataset

The dataset used in this experiment is the Persuasive Essays dataset introduced in Section 3.1.1, which is a benchmark constructed by Stab and Gurevych (2017). This dataset is also used in Eger et al. (2017). It comprises 402 persuasive essays randomly selected from an online forum, with 322 essays used for training and 80 essays for testing. Table 5.1 provides statistics of the dataset. Table 5.2 shows the distribution of its component and relation distributions.

5.3.2 Dataset-specific adaptation and post-processing

When training my neural parsers on the Persuasive Essays dataset, I first need to make sure that the parsers, the dataset, and the dependency representations are compatible with each other. I therefore performed some dataset-specific adaptation and post-processing.

Considering that most relations hold within paragraphs in this dataset, my approach operates at the paragraph-level, in line with Stab and Gurevych (2017). This means that each paragraph is processed as an argument, having its own argument graph and acting

	All	Per essay
Token	147,271	336.3
Component	6,089	15.1
Sentence	7,116	17.7
Paragraph	1,833	4.6

Table 5.1: Statistics of the Persuasive Essays dataset.

		All	Per essay
Component	MajorClaim	751	10.2
	Claim	1,506	0.1
	Premise	3,832	0.2
	Total	6,089	11.0
Relation	For	2,345	5.8
	Against	496	1.2
	Support	3,613	9.0
	Attack	219	0.5

Table 5.2: Component and relation distribution of the Persuasive Essays dataset.

as an individual data point during training and inference. As a result, each paragraph has a pseudo-token *ROOT* in its dependency representation. If a paragraph contains one or more MajorClaims, *ROOT* considers all of them as dependants. Otherwise, if no MajorClaims are present, *ROOT* includes all Claims in the paragraph as dependants.

Once the structure of a paragraph has been predicted by the parser, I apply a series of post-processing steps, which also operate at the paragraph-level. These steps are designed to resolve conflicts and to create coherent argument graphs, following similar techniques used by Eger et al. (2017). The following post-processing steps are performed in order:

1. All tokens in a segment should be of the same category, but this is not always the case in the predictions. To handle this, I define the category of each segment as the category which is assigned to at least three-fifths of the tokens in that segment.

2. Any incoming or outgoing inter-segment edge of any token in a segment count as an edge of the entire segment.

3. For the links between segments, only valid edges are retained. These valid edges include MajorClaim→Root, Claim→Root, Claim→MajorClaim, Premise→Claim, and Premise→Premise.

4. There cannot be more than one edge between any two segments. Therefore, in the case of two linked segments having multiple edges, the one with the highest prediction probability is retained.

5. When a segment has multiple outgoing edges, only Claim→MajorClaim edges are preserved. This is because only Claims can have multiple heads, and they must be MajorClaims. If there is no Claim→MajorClaim edge among the multiple outgoing edges, only the edge with the highest probability is retained.

After these paragraph-level post-processing steps, I combine the sub-graphs generated from each paragraph into a complete argument graph of the entire essay. In the

paragraph-level predictions, CLAIMs in a paragraph might be linked to the ROOT node of that paragraph. But in the Persuasive Essays dataset, every CLAIM is linked to all MAJORCLAIMs. Adhering to this rule, when generating the argument graph of the entire essay, I first remove all CLAIM→ROOT edges, and then link every CLAIM to all MAJORCLAIMs (if the link does not already exist) while preserving the original edge label. This process ensures the generation of a valid final argument graph of the entire essay.

5.3.3 Systems

I select two models from Eger et al. (2017) as the baselines for evaluation:

- *LSTM-Parser*: the best-performing model in an end-to-end dependency parsing approach.
- *LSTM-ER*: the overall best-performing model that frames argument mining as end-to-end sequence labelling.

To gain access to the actual predictions of the baseline models, which are required by the significance test, I reimplemented the baseline models. My reimplementation yields similar results to those reported in Eger et al. (2017) on the Persuasive Essays dataset. These results are displayed in Table 5.5 from Section 5.3.6.

For my approach AMDP, I have implemented four models to compete with the baselines:

- *Biaff_exc*: the biaffine parser with my undercut-exclusive representation.
- *Biaff_inc*: the biaffine parser with my undercut-inclusive representation.
- *GNN_exc*: the GNN-based parser with my undercut-exclusive representation.
- *GNN_inc*: the GNN-based parser with my undercut-inclusive representation.

Although the Persuasive Essays dataset does not contain any undercuts, I still have included the undercut-inclusive representation in this experiment. This is because I want to test if the undercut-inclusive representation will have any positive or negative effect on analysing arguments without undercuts.

5.3.4 Experimental setup

The baseline models are trained at the paragraph-level using the default hyperparameter configuration provided in the source code. The pre-trained GloVe embeddings (Pennington et al., 2014) used in Eger et al. (2017) are employed for all LSTM encoders throughout the experiment. Additionally, the Stanford syntactic dependency parser (Chen and Manning, 2014) is utilised to generate the syntactic trees required by the LSTM-ER model.

During the training of my AMDP models, I randomly select 30 essays from the training set as the development set. This set is used to tune the following hyperparameters: the interpolation factor λ in Equation 5.11 for the biaffine parser ($\lambda = 0.05$), λ_1, λ_2 in Equation 5.27 for the GNN-based parser ($\lambda_1 = 0.75, \lambda_2 = 0.05$), the dropout rate for the biaffine parser and the GNN-based parser ($dropout = 0.1$), and the number of training epochs for the early stopping mechanism. I use the pre-trained BERT-base-cased model by Wolf et al. (2020) as the encoder, and freeze its weights during training. I abandoned GPT-2 Radford et al. (2019), another pre-trained encoder, because of its slightly inferior performance compared to BERT in the pilot experiment. I use the Adam optimiser (Kingma and Ba, 2014) with parameters $\beta_1 = 0.95$, $\beta_1 = 0.98$, and $\epsilon = 1e^{-9}$, and adopt the learning rate strategy from Vaswani et al. (2017) with warm-up steps set to 1000. All unspecified hyperparameters follow the default values from Dozat and Manning (2018) and Ji et al. (2019), respectively.

Each model is trained ten times with different random initialisations, except the LSTM-based models and their variants, which are trained only once due to excessively long training time. The reported results are the average performance.

5.3.5 Evaluation metrics

I use the standard evaluation metric proposed in Persing and Ng (2016) to compare the predicted argument graphs against the ground truth annotations. The metric measure the F_1 scores for component identification and relation identification respectively: $F = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$, where TP represents true positive, FP false positive, and FN false negative. This metric operates at the segment-level, *i.e.* the units considered are segments (as opposed to tokens) and relations among them. While component segmentation should be evaluated at the token-level, some studies adopt flexible strategies in adapting the segment-level F_1 metric to the evaluation of argument mining models. For example, Eger et al. (2017) report two different types of F_1 scores in their experiments: one at a 50% match and another at a 100% match. A match at 50% is counted if a segment overlaps with at least 50% of the tokens in the corresponding gold standard segment. A 50% match is counted if a segment overlaps with at least 50% of the tokens in the corresponding gold standard segment, while a 100% match occurs if a segment has the exact same boundaries as its corresponding gold standard segment. In my experiments, I only report results using the strictest 100% match criterion.

For component identification, a TP is a gold standard component that has an exact match in the predictions — having exactly the same segment boundaries and the same component label. An FP is a predicted component that has no exact match in the gold standard. An FN is a gold standard component that has no exact match in the predictions.

For relation identification, TPs, FPs, and FNs are defined analogously. Here an exact

match means 1) the source as well as the target is an exact match, and also 2) they have the same relation label.

There exist alternative evaluation metrics for assessing argument mining models, and one such metric that accommodates partial agreement in component segmentation is the CASS metric proposed by Duthie et al. (2016). The CASS metric considers the degree of difference in segment boundaries, assigning a higher score to cases where there is a small mistake in determining segment boundaries (*e.g.* missing a word or two around the gold standard boundaries) compared to a full miss. However, I have chosen not to employ the CASS metric in my evaluation, because partial agreement in component segmentation is not a primary focus of my experiments, and also because the F_1 metric is more widely adopted, facilitating a more straightforward comparison with other systems.

In order to evaluate the statistical significance of the performance difference between two argument mining models, I conduct paired Monte Carlo permutation tests with a two-tailed approach (Dwass, 1957, Nichols and Holmes, 2002, Marozzi, 2004). In a paired Monte Carlo permutation test, the observed differences between paired data points are randomised many times to generate a null distribution of the test statistic under the assumption that the models perform identically. The actual observed test statistic is then compared to this distribution to determine its significance.

Applying these paired tests directly to argument mining is not straightforward, because the numbers of components or relations returned by two models may be different. To address this, I first construct a permutation table as depicted in Table 5.3. Taking component identification as an example, the first column in this table consists of component indices. There are two kinds of components used to construct the permutation table: 1) all components $g_1, g_2, ..., g_m$ in the gold standard, and 2) components $p_1, p_2, ..., p_n$ that are not in the gold standard but are predicted as argumentative by either model, excluding duplicates.

The other two columns in the table contains the predicted labels of the items in the first

Item	Model_a	Model_b
g_1	TP	FN
g_2	FN	FN
...	...	...
g_m	TP	TP
p_1	FP	TN
p_2	FP	FP
...	...	...
p_n	TN	FP

Table 5.3: A permutation table for paired permutation tests.

column, one column per model. The gold standard components $g_1, g_2, ..., g_m$ are assigned TP or FN labels following the definitions above. The predicted components $p_1, p_2, ..., p_n$ are introduced to the permutation tests by me; they can only be labelled as FP or TN (true negative). Since these components are not present in the gold standard, whenever a model proposes such a component, it should be penalised by receiving an FP label. But the other model should not be penalised if it does not propose this component. As a result, if a component p_i is predicted by Model_a but not by Model_b, it is labelled as an FP for Model_a and an TN for Model_b.

Then, permutations are generated by swapping labels in the second and third columns for randomly selected rows. The two-tailed p-value is then calculated using Equations 5.28 and 5.29:

$$diff_i = \begin{cases} 1 & , |F_a - F_b| < |F_a'^i - F_b'^i| \\ 0 & , |F_a - F_b| \geq |F_a'^i - F_b'^i| \end{cases} \tag{5.28}$$

$$p = \frac{1 + \sum_{i=1}^{N} diff_i}{1 + R} \tag{5.29}$$

In Equation 5.28, $diff_i$ indicates if the actual performance difference between the two models is smaller than that in the i-th permutation. Here F_a represents the actual F_1 score for Model_a, and $F_a'^i$ denotes the F_1 score for Model_a in the i-th permutation. In Equation 5.29, p is calculated as the percentage of occasions where the actual performance difference is smaller than that in a random permutation. Here R is the number of sampled permutations.

In this thesis, I choose $\alpha = 0.01$ as the threshold (*i.e.* the "alpha level") for p-value, and the number of sampled permutations $R = 10000$ following the suggestion in Marozzi (2004). In cases where a model is trained with multiple initialisations, the permutation tests are performed using the model instance that performs closest to the average.

5.3.6 Results

Table 5.4 displays the F_1 scores for the baseline models and my models in the AMDP approach, in terms of component identification and relation identification respectively. Results for the corresponding permutation tests are shown in Table 5.5.

The F_1 scores achieved by the various AMDP models range from 72.8% to 73.8% for component identification, and from 45.9% to 49.4% for relation identification. Compared to my four AMDP models, the baseline models *LSTM-Parser* and *LSTM-ER* achieve significantly lower F_1 scores of 58.4% (eight paired permutation tests, all $p < 0.01$) and 70.2% (eight paired permutation tests, all $p < 0.01$) for components, and 34.9% (eight paired permutation tests, all $p < 0.01$) and 45.1% (eight paired permutation tests, all

Model	Component	Relation
Baseline		
LSTM-Parser	58.4 (58.9)	34.9 (35.6)
LSTM-ER	70.2 (70.8)	45.1 (45.5)
AMDP		
Biaff_exc	72.9	45.9
Biaff_inc	72.8	45.9
GNN_exc	73.8	49.4
GNN_inc	73.8	49.4

Table 5.4: F_1 scores for baseline models and models in my approach (AMDP) on the Persuasive Essays dataset. For *LSTM-Parser* and *LSTM-ER*, I report the results in my own implementation and also those published by Eger et al. (2017) (in parentheses).

	LSTM-Parser	LSTM-ER	Biaff_exc	Biaff_inc	GNN_exc	GNN_inc
LSTM-Parser	-	<	<	<	<	<
LSTM-ER	>	-	<	<	<	<
Biaff_exc	>	>	-	=	<	<
Biaff_inc	>	>	=	-	<	<
GNN_exc	>	>	>	>	-	=
GNN_inc	>	>	>	>	=	-

Table 5.5: Results for the permutation tests regarding Table 5.4. P-values in blue are for component identification, red for relation identification. Symbols express whether the system mentioned in the row is significantly better (>), worse (<) or indistinguishable (=) from the system mentioned in the column, at $\alpha = 0.01$.

$p < 0.01$) for relations. Therefore, my models outperform not only the best-performing model in Eger's dependency parsing approach, namely *LSTM-Parser*, but also their overall best-performing model *LSTM-ER*, and therefore establish a new state of the art on this dataset. The results demonstrate the effectiveness of my approach in achieving end-to-end argument mining, and indicate that formulating argument mining as end-to-end dependency parsing is a promising direction.

In terms of the comparison between my models using the GNN-based parser against those using the biaffine parser, *GNN_exc* significantly outperforms *Biaff_exc* by 0.9% absolute F_1 score for component identification ($GNN\_exc = 73.8\%$, $Biaff\_exc = 72.9\%$, $p < 0.01$) and 3.5 % for relation identification ($GNN\_exc = 49.4\%$, $Biaff\_exc = 45.9\%$, $p < 0.01$). The differences are similar between *GNN_inc* and *Biaff_inc*. This reveals that the GNN-based parser exhibits superior performance compared to the biaffine parser.

Regarding the comparison between the undercut-exclusive and the undercut-inclusive representations, the results reveals no significant difference in performance. Both representations yield similar F_1 scores for component and relation identification, which implies

that the undercut-inclusive representation can be used even in scenarios without undercuts, without compromising performance. This result suggests the flexibility of the undercut-inclusive representation.

In conclusion, the results show the effectiveness of my AMDP approach in end-to-end argument mining. The GNN-based parser performs better than the biaffine parser, and the undercut-inclusive representation can be applied to either undercuts or normal arguments without compromising performance.

5.3.7 Ablation study: neural parsers *vs.* dependency representations

The observed advantage of the AMDP approach over the competing approaches in Eger et al. (2017) might be attributed to two key factors: my dependency representations and my neural dependency parsers. Firstly, my dependency representations in the AMDP approach might be more informative than the representations used in Eger et al. (2017), as I have speculated in Section 5.1. Secondly, the biaffine parser and the GNN-based parser might be more suitable for argument mining than the LSTM dependency parser used in Eger et al. (2017). I want to determine which of these two factors is the decisive one, or if they both contribute to the success of AMDP. To achieve this, I build systems where the factors can be selectively switched off and present the results in an ablation format. Since there is no significant performance difference between my two representations on the Persuasive Essays dataset, I only use the undercut-exclusive representation.

The ablation study compares six different combinations of parsers and representations:

- *LSTM_Eger*: the LSTM parser and the representation used in Eger et al. (2017).
- *LSTM_exc*: the LSTM parser used in Eger et al. (2017) and my undercut-exclusive representation.
- *Biaff_Eger*: my biaffine parser and the representation used in Eger et al. (2017).
- *Biaff_exc*: my biaffine parser and my undercut-exclusive representation.
- *GNN_Eger*: my GNN-based parser and the representation used in Eger et al. (2017).
- *GNN_exc*: my GNN-based parser and my undercut-exclusive representation.

The results of the cross-comparisons are shown in Table 5.6. Results for the corresponding permutation tests are shown in Table 5.7. Comparing the parsers, both the biaffine parser and the GNN-based parser significantly (eight paired permutation tests, all $p < 0.01$) outperform the LSTM parser used in Eger et al. (2017), regardless of which representation is used. For example, using my undercut-exclusive representation, the

		Representation			
		Eger		Exc	
		Component	Relation	Component	Relation
Parser	LSTM	58.4	34.9	67.9	36.0
	Biaffine	65.7	38.4	72.9	45.9
	GNN	66.1	40.3	73.8	49.4

Table 5.6: F_1 scores for models with different combinations of parsers and representations on the Persuasive Essays dataset. Exc = my undercut-exclusive representation. Eger = the representation used in Eger et al. (2017).

	LSTM_Eger	LSTM_exc	Biaff_Eger	Biaff_exc	GNN_Eger	GNN_exc
LSTM_Eger	-	<	<	<	<	<
LSTM_exc	>	-	>	<	>	<
Biaff_Eger	>	>	-	<	<	<
Biaff_exc	>	>	>	-	>	<
GNN_Eger	>	>	>	<	-	<
GNN_exc	>	>	>	>	>	-

Table 5.7: Results for the permutation tests regarding Table 5.6. P-values in blue are for component identification, red for relation identification.

GNN-based parser achieves an F_1 score of 73.8% for component identification and 49.4% for relation identification, significantly (two paired permutation tests, all $p < 0.01$) higher than those of the LSTM parser, which are 67.9% and 36.0%. This demonstrates that the two parsers in the AMDP approach are more powerful at capturing dependencies than the LSTM parser in Eger et al. (2017).

Regarding the representations, every parser with the undercut-exclusive representation achieves significantly (six paired permutation tests, all $p < 0.01$) higher F_1 scores compared to its counterpart with the representation used in Eger et al. (2017). This shows the effectiveness of the undercut-exclusive representation in my AMDP approach.

When designing this representation, I anticipated it to be more effective at capturing within-segment and long-distance inter-segment dependencies. Figure 5.7 shows an example where my representation indeed captures some such dependencies better than Eger et al.'s. In the given example, *Biaff_exc* accurately predicts all the components and relations in the paragraph from the test data. *Biaff_Eger* fails to identify the parenthetical phrase "such as Beijing Place, Shanghai Artist Museum" within the relatively long component PREMISE1 as an argumentative segment. With Eger et al.'s representation, the parser would be forced to infer a relation between the first token in CLAIM ("historic") and all tokens in this text sequence, the distance between which are 14 to 20 tokens. With my representation, those tokens are treated as one unit through within-segment relations, so that the distance of such relations is only one token. *Biaff_Eger* also fails to detect the relation between

Meanwhile, [historic buildings can be a source of maintenance fees]$_{\text{Claim}}$. [There are many historic buildings, such as Beijing Place, Shanghai Artist Museum, they are all able to support themselves financially by charging tens of thousands of tourists]$_{\text{Premise1}\rightarrow\text{Claim}}$. [Our governments will be happy with those efficient consequences, and a majority of cities also can imitate this economical cycle] $_{\text{Premise2}\rightarrow\text{Claim}}$.	Meanwhile, [historic buildings can be a source of maintenance fees]$_{\text{Claim}}$. [There are many historic buildings]$_{\text{Premise1}\rightarrow\text{Claim}}$, such as Beijing Place, Shanghai Artist Museum, [they are all able to support themselves financially by charging tens of thousands of tourists]$_{\text{Premise2}\rightarrow\text{Claim}}$. [Our governments will be happy with those efficient consequences, and a majority of cities also can imitate this economical cycle] $_{\text{Premise3}}$.
Gold standard & Biaff_exc	Biaff_Eger

Figure 5.7: Argument structures of an example paragraph predicted by *Biaff_exc* (identical to the gold standard) and by *Biaff_Eger*.

PREMISE3 and CLAIM. This might be due to the fact that these two components are located at opposite ends of the paragraph. With Eger et al.'s representation, the parser would be forced to infer many relations between the first token in CLAIM ("historic") and all tokens in PREMISE3 ("Our governments will be happy with those efficient consequences, and a majority of cities also can imitate this economical cycle"). However, in the actual prediction of *Biaff_Eger*, the majority of such relations are not correctly predicted.

In conclusion, the ablation study confirms that it is both the parsers and the dependency representation which contribute to the improved performance of my AMDP approach.

5.4 Experiment: argument mining with explicit undercut modelling

Having established that the biaffine and the GNN-based parsers are good processors and that both my representations work well, I can now move to the experiment that is far more central to my thesis, namely the experiment with explicit undercut modelling on the Quora dataset.

5.4.1 Dataset-specific adaptation and post-processing

In the previous experiment, my AMDP approach operated at the paragraph-level, as most relations are within-paragraph in the Persuasive Essays dataset. In the upcoming experiment, it will operate at the document-level. This means that each document is regarded as a data point during training and inference. This is because relations in the Quora dataset are more likely to span across paragraphs, rather than operating within individual paragraphs, as paragraph breaks are used less consistently in the Quora dataset than in the Persuasive Essays dataset. Therefore, it is necessary to consider the entire

document to capture the full range of argument relations.

In terms of adapting the *ROOT* node in my representations to the Quora dataset, for each argument graph, the *ROOT* node takes all nodes that have at least one dependant but no head as its dependants.

In this experiment, I merged all instances of STAKE and ANECDOTE in the Quora dataset to form a new category called STAKE+ANECDOTE, as discussed in Section 4.4.3.

Post-processing in this experiment operates at the document-level. The same post-processing steps as in Section 5.3.2 are performed, except that when a segment has multiple outgoing edges, all edges except the one with the highest probability are removed. This is because a source component in the Quora dataset cannot have more than one target.

5.4.2 Systems

Unlike in the previous experiment, direct comparisons with existing approaches are not possible as no one so far has computationally modelled undercuts. However, I still wanted to provide a baseline. Therefore, I adapted my reimplementation of the *LSTM-ER* model and use it as the baseline model in this experiment. My baseline simulates Peldszus and Stede (2015b)'s approach by transforming all undercuts into direct ATTACKs during training. During inference, I randomly convert all predicted direct attacks to undercuts in proportion to the ratio of undercuts to overall attacks in the original Quora dataset. The dependency representation chosen for the baseline is my undercut-exclusive representation which I have already shown to be the best known representation for arguments without undercuts. This baseline therefore is considered to be a very strong one.

As for my AMDP approach, I have implemented two models: *Biaff_inc* and *GNN_inc*.

5.4.3 Experimental setup

As we have already seen in Section 4.4.3, the distribution of segment categories in the Quora dataset is skewed: PROPOSITION = 51.7%, STAKE + ANECDOTE = 1.4%, ANALOGY = 2.5%, non-argumentative = 44.4%. To address this imbalance, I use the leaning strategy introduced in Ting (1998) throughout this experiment. This strategy, called the cost-sensitive learning strategy, works by assigning different weights for different segment categories when calculating the loss during training. The weight W_i for a segment category i among the set of all segment categories S is calculated as in Equation 5.31:

$$\alpha_i = \frac{\sum_{k \in S} |k|}{|i|^{\lambda}} \tag{5.30}$$

$$W_i = \frac{\alpha_i}{\sum_{k \in S} \alpha_k} \tag{5.31}$$

Here $|i|$ means the number of instances of category i in the dataset. I have modified the original strategy by adding a hyperparameter λ in Equation 5.31, which can be tuned using the development set. In this experiment, the λ in Equation 5.31 is set to 0.25.

All other experimental settings not specified here follow the same configurations as in the previous experiment (Section 5.3.4) where applicable.

5.4.4 Results

The F_1 scores for the baseline model and my models on the Quora dataset are presented in Table 5.8. Results for the corresponding permutation tests are shown in Table 5.9. Both my models comfortably beat the baseline in terms of the overall performance: *Biaff_inc* outperforms the baseline by 3.3% for component identification ($Biaff\_inc = 62.4\%$, $baseline = 59.1\%$, $p < 0.01$) and 10.2% for relation identification ($Biaff\_inc = 41.7\%$, $baseline = 31.5\%$, $p < 0.01$); *GNN_inc* outperforms the baseline by 7.1% for component identification ($GNN\_inc = 66.2\%$, $baseline = 59.1\%$, $p < 0.01$) and 14.3% for relation identification ($GNN\_inc = 45.8\%$, $baseline = 31.5\%$, $p < 0.01$).

The results confirm that the GNN-based parser is more efficient than the biaffine parser: *GNN_inc* significantly outperforms Biaff_inc by 3.8% for component identification ($GNN\_inc = 66.2\%$, $Biaff\_inc = 62.4\%$, $p < 0.01$) and 4.1% for relation identification ($GNN\_inc = 45.8\%$, $Biaff\_inc = 41.7\%$, $p < 0.01$).

	Baseline	Biaff_inc	GNN_inc
Component	59.1	62.4	66.2
Proposition	60.6	63.7	67.8
Stake + Anecdote	47.9	51.2	53.3
Analogy	35.4	38.6	41.5
Relation	31.5	41.7	45.8
Support	40.5	45.1	48.0
Attack	10.5	33.8	39.9
Direct Attack	12.5	38.4	40.3
Undercut	5.2	21.6	38.8

Table 5.8: F_1 scores for models in my AMDP approach on the Quora dataset.

	Baseline	Biaff_inc	GNN_inc
Baseline	-	<	<
Biaff_inc	>	-	<
GNN_inc	>	>	-

Table 5.9: Results for the permutation tests regarding Table 5.8. P-values in blue are for component identification, red for relation identification.

When considering the performance on undercuts, both models significantly outperform the baseline by a large margin, with an increase of 16.4% for *Biaff_inc* ($Biaff\_inc = 21.6\%$, $baseline = 5.2\%$, $p < 0.01$) and 33.6% for *GNN_inc* ($GNN\_inc = 38.8\%$, $baseline = 5.2\%$, $p < 0.01$). This suggests that undercuts, despite their intricate nature, are structures that can be effectively analysed in argument mining. It also shows that the GNN-based parser is better than the biaffine parser at recognising undercuts: *GNN_inc* significantly outperforms *Biaff_inc* by 17.2% ($GNN\_inc = 38.8\%$, $Biaff\_inc = 21.6\%$, $p < 0.01$).

The overall results show a slight decrease in F_1 scores compared to that on the Persuasive Essays dataset. This could be attributed to the informal and complex nature of arguments in the Quora dataset. These arguments are more challenging to analyse compared to the more well-structured arguments found in the Persuasive Essays dataset.

Let us now look at my models' performance by component category. Both models exhibit a consistent trend: their performance is best on PROPOSITIONs ($Biaff\_inc = 63.7\%$, $GNN\_inc = 67.8\%$), followed by STAKE+ANECDOTEs ($Biaff\_inc = 51.2\%$, $GNN\_inc = 53.3\%$), and lastly ANALOGYs ($Biaff\_inc = 38.6\%$, $GNN\_inc = 41.5\%$). It is understandable that they are best at recognising PROPOSITIONs, because PROPOSITION is the predominant component category in the Quora dataset.

Both models perform better in recognising STAKE+ANECDOTEs compared to ANALOGYs. This observation could be attributed to the frequent presence of distinct linguistic cues in STAKEs and ANECDOTEs. For instance, in the sentence "I have been a teacher in a public school for ten years ...", the phrase "I have been" serves as a clear linguistic cue for the presence of a STAKE. Similarly, in the sentence "When I was in college, I encountered a similar situation ...", the phrase "When I was in college" indicates the presence of an ANECDOTE. On the other hand, ANALOGYs exhibit more varied forms and often lack such explicit cues, and very often they can look like a sudden change of topic. For example, when arguing about whether animal testing should be banned, the author uses the analogy "If someone was gifted a new, functioning ceiling fan as a gift and then they use it to kill themselves, that's on them. That doesn't make the concept of gifting someone a ceiling fan to keep themselves cool a bad thing" to support the point that "Anything, no matter how universally good it's considered, can be misused". The analogy compares the action of gifting someone a ceiling fan to animal testing, and the textual material is mostly about the fan. My models fail to recognise this analogy. Instead, they predict it as non-argumentative, which is understandable as ANALOGYs can appear to be off-topic.

5.4.5 Recognising undercuts: biaffine parser *vs.* GNN-based parser

In the analysis of the results in Table 5.8, I have already shown that the GNN-based parser is more efficient than the biaffine parser at recognising undercuts. To understand the reasons for this disparity in performance, I will now compare the errors made by the two models on undercuts. Undercuts structurally consist of different parts: three components and two relations. This post-hoc analysis aims to discern which parts of an undercut is most error-prone for the GNN-based parser.

Figure 5.8 shows the structure of an undercut with its dependency representation. In this figure, nodes 1-3 are components, with nodes 1' and 2' being their corresponding relation nodes – 1' representing an undercutting relation and 2' representing a supporting relation. An undercut consists of three elements: an undercutting component (*e.g.* node 1 in Figure 5.8), its target relation ("node 2 ⇐ node 2' ⇐ node 3"), and the link between them ("node 1' ⇐ node 2"'). This structure is recursive because the target relation itself consists of a source component (node 2), a target component (here node 3), and the link between them (node 2').

Therefore, an error can be caused by either the failure to recognise any single element in an undercut, or a combination of some of such failures. For example, failing to recognise the undercutting component, failing to recognise the target relation, or failing to recognise either of them will all lead to an error in recognising the entire undercut. Moreover, some such errors are interdependent of each other. For example, an error in recognising the source component in the target relation will automatically lead to an error in recognising the entire target relation. These facts make it infeasible to thoroughly investigate every potential type of error in undercut recognition. I will therefore only look at the three types illustrated in Figure 5.8: type I, *i.e.* errors in recognising the undercutting component;

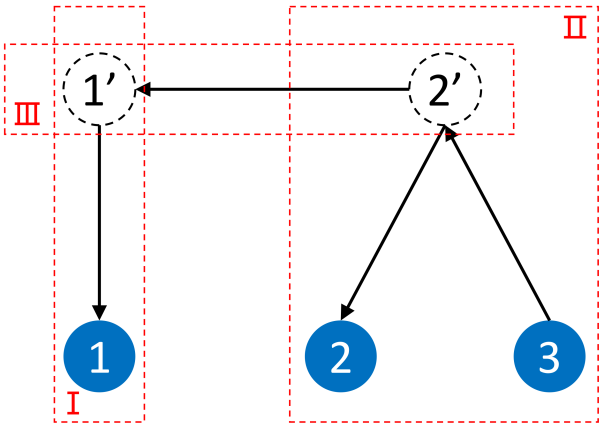


Figure 5.8: The dependency representation of an undercut, and three types of error in undercut recognition. Labels of nodes or edges are not displayed in this figure.

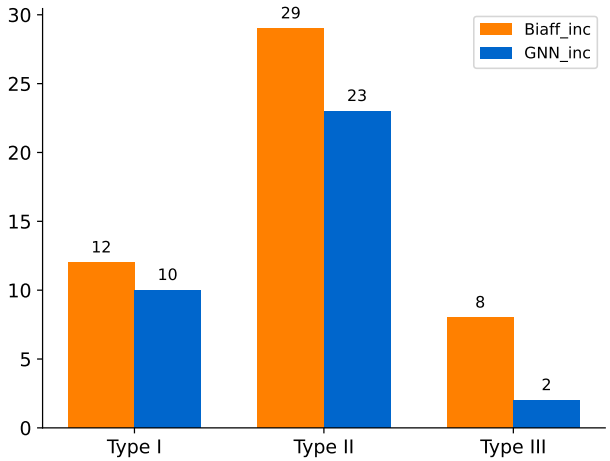


Figure 5.9: Distribution of three types of error in undercut recognition.

type II, *i.e.* errors in recognising the target relation; type III, *i.e.* errors only in recognising the link between them. Please note that type III only include the cases where the link is incorrectly predicted but the undercutting component and the target relation are correctly predicted. Since errors of type I and type II will usually lead to errors in recognising the link, by defining type III in this way, I am able to examine the errors in link recognition independently of undercutting component and target relation recognition.

I performed an error analysis by counting the errors made by my models. Figure 5.9 shows the distribution of the three types of error among all undercuts (N=58) in the test set. Due to the small sample size, I refrain from testing the statistical significance of the results in this figure, and instead interpret the numbers only qualitatively. According to Figure 5.9, *Biaff_inc* produces relatively 20% more type I errors and 26% more type II errors than *GNN_inc*. For type III errors, the difference becomes much bigger, which is 300%. This pattern suggests that the GNN-based parser recognises all three elements of an undercut more effectively than the biaffine parser. The most pronounced difference is observed in type III errors. This may suggest that when it comes to undercut recognition, the GNN-based parser's advantage over the biaffine parser could be in recognising the link between the undercutting component and the target relation.

Regarding the disparity in link recognition between the GNN-based parser and the biaffine parser, I hypothesise that the GNN-parser's advantage in identifying higher-order dependencies, as elaborated in Section 5.2.2, plays an important role. As depicted in Figure 5.10:

- (a) In the undercut-exclusive representation, a direct ATTACK requires one hop.
- (b) In the undercut-inclusive representation, a direct ATTACK requires two hops.

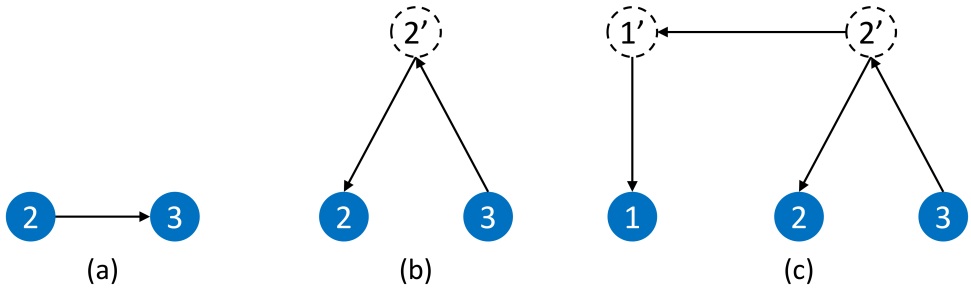


Figure 5.10: Illustration of the number of hops required for different relation representations: (a) a direct attack in the undercut-exclusive representation, (b) a direct attack in the undercut-inclusive representation, and (c) an undercut in the undercut-inclusive representation.

- (c) In the undercut-inclusive representation, an undercut requires three hops.

A direct ATTACK is represented as a 1-hop relation in the undercut-exclusive representation, but in the undercut-inclusive representation it becomes a 2-hop relation. Despite such an increase, the performance difference on the Persuasive Essays dataset in terms of my two dependency representations is not significant, as discussed in Section 5.3.6. This implies neither parser sees a performance loss transitioning from a 1-hop to a 2-hop relation.

However, the narrative changes when comparing direct ATTACKs and undercuts in the undercut-inclusive representation, where direct ATTACKs and undercuts are represented as 2-hop and 3-hop relations, respectively. According to Table 5.8, both *Biaff_inc* and *GNN_inc* display weaker performance in identifying undercuts compared to direct ATTACKs. However, the magnitude of this difference in performance varies between the two models. Specifically, the F_1 score of *Biaff_inc* for undercuts lags behind its score for direct ATTACKs by 16.8%, while for *GNN_inc* the difference is only 1.5%. These results suggest that as hop count increases from two to three, the GNN-based parser suffers a smaller performance loss than the biaffine parser. This aligns well with my earlier prediction that the GNN-based parser would perform well in capturing higher-order dependencies.

5.5 Chapter summary

In this chapter, I introduced my approach to argument mining. This approach formulates argument mining as a dependency parsing task.

My approach models argument mining in an end-to-end fashion. More importantly, it has the ability to model undercuts computationally. To achieve this, I designed two new dependency representations for arguments, one that does not model undercuts, and one that explicitly models undercuts. My representations unify all subtasks of argument mining as tasks at the token-level, so that all can be processed simultaneously within one

neural model. My undercut-inclusive representation converts undercuts into relational nodes, so that they can be directly modelled by existing neural models. Additionally, I introduced two neural dependency parsers for my approach, namely a biaffine parser and a GNN-based parser.

I evaluated my approach on the newly introduced Quora dataset, making it the first end-to-end approach applied to informal arguments in online discussions. The results show that my biaffine parser and my GNN-based parser are both significantly better than a strong baseline. The GNN-based parser is also more effective than the biaffine parser.

In order to compare the relative effectiveness of the two parsers in undercut recognition, I performed an error analysis. Although the data size is small, allowing only preliminary conclusions, the results show that the GNN-based parser outperforms the biaffine parser in undercut recognition. Upon examining the errors from the two parsers, I found that the primary difference is in identifying the link between the undercutting component and its target relation.

I also evaluated my approach on the Persuasive Essays dataset which is a bench mark containing formal arguments. The results show that all models in my approach significantly outperform a strong baseline and achieve a new state of the art. In the ablation study I conducted, I found that my dependency representation and my neural parsers both contribute to the performance boost.

Chapter 6

External knowledge integration

In addition to exploring new dependency representations and neural parsers, I also examine the benefits of integrating external knowledge into argument mining models. In this chapter, I employ two types of external knowledge integration techniques, namely feature-based integration (Section 6.1.1) and transfer-based integration (Section 6.1.2). My exploration encompasses a variety of external knowledge sources, including syntax, discourse structures, knowledge graphs, and large language models. In Section 6.2, I evaluate the effectiveness of these techniques and knowledge sources on the same two datasets that were used in Chapter 5, one formal (the Persuasive Essays dataset), the other informal (the Quora dataset).

The integration methods I employ are designed to be model-agnostic, so that they do not need to be tailored to any particular model. I avoid model-specific strategies in this experiment, such as refining the BERT encoder with supplemental text. Furthermore, given the end-to-end nature of my AMDP approach, I only study holistic techniques that cover all subtasks of argument mining. Strategies that target specific subtasks, like leveraging a pre-trained BERT specifically for enhancing relation identification, fall outside the scope of this investigation.

6.1 Approaches to external knowledge integration

External knowledge can be integrated into argument mining models in different ways, for instance as features to be added to the input text, and as pre-trained model weights through transfer learning. The knowledge to be integrated can be obtained from various sources. Such integration allows models to benefit from additional information and context. In this section, I will introduce the external knowledge sources used in this study and how they are integrated into argument mining models.

6.1.1 Feature-based approach

In the feature-based approach, external knowledge is provided to the model in the form of additional features that augment the input text. In the context of end-to-end neural models, these features are usually vector representations extracted from external knowledge sources, which are then combined with the input original representation of the input text.

For this approach, I investigate various sources of external knowledge, including syntactic information, discourse information, information in a curated knowledge graph, and information obtained from a pre-trained large language model. Notably absent from this list is the inclusion of pre-trained word embeddings, despite their use as a knowledge integration feature in some argument mining studies, *e.g.* Fromm et al. (2019). This is because these embeddings have already demonstrated their effectiveness across multiple NLP tasks (Gururangan et al., 2020, Raffel et al., 2020) and have become a standard and integral component of most neural models. Instead of treating pre-trained word embeddings as a variable to be examined, I will use them in all my experiments. The pre-trained BERT encoder employed in my biaffine and GNN-based parsers already integrates WordPiece embeddings (Wu et al., 2016), a specific kind of pre-trained word embeddings.

6.1.1.1 Syntactic information

Syntactic information plays a crucial role in the analysis of the internal structure of sentences. A syntactic parse tree provides a hierarchical representation of the sentence structure, illustrating the syntactic relationships between tokens. An example of such a tree is given in Figure 6.1. In the example, the token "threatened" is the head of "killed" "tourism", "has", "nature", and two punctuation marks. The label of the relation between "threatened" and "nature" is "obj", which means that "nature" is the direct object of the verb "threatened".

My choice of using syntactic parse trees as a source of external knowledge for argument mining is motivated by two reasons. First, understanding the syntactic structure of a sentence can be beneficial for determining segment boundaries in argument mining. Second, syntactic information can assist in resolving ellipsis and anaphora (Dalrymple et al., 1991, Lappin and Leass, 1994), which are common phenomena in argumentative texts, particularly informal arguments. Ellipsis refers to the omission of certain words or phrases that can be inferred from the context, while anaphora involves the reference of a

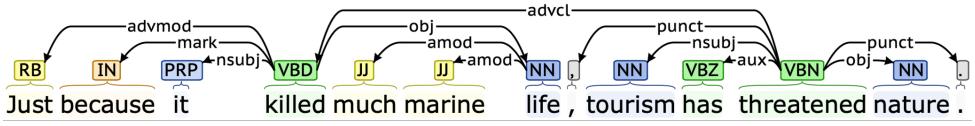


Figure 6.1: An example syntactic parse tree produced by Stanford CoreNLP (Manning et al., 2014).

word or phrase to a previously mentioned antecedent.

In order to integrate syntactic information, I first obtain the syntactic parse tree of each sentence in an argument using the Stanford CoreNLP syntactic dependency parser (Manning et al., 2014). I next encode the information of each token within the syntactic parse tree as its path to the root node in the parse tree, in the form of a list of tuples $(edge\_label, node\_index)$ that are collected along the path. For example, in Figure 6.1, the token "life" has the path $[(obj, 4), (advcl, 11), (root, 0)]$. Such encoded information is then used as an additional feature for each token.

6.1.1.2 Discourse information

While syntactic information reveals the structure of a sentence through token-level dependencies, discourse information describes the overall structure of an entire document through clause-level dependencies. Discourse structures can be represented using the Rhetorical Structure Theory (RST) framework (Mann and Thompson, 1988), which describes how different segments in a document are hierarchically organised into larger discourse units. At the core of RST is the idea that a document is not just a random sequence of sentences but is organised hierarchically, where different segments, or spans of text, contribute to the overall meaning of the document.

Consider the following example text:

> **Example 6.1**
> "[Farmington police had to help control traffic recently]$_1$ [when hundreds of people lined up to be among the first applying for jobs at the yet-to-open Marriott Hotel.]$_2$ [The hotel's help-wanted announcement - for 300 openings - was a rare opportunity for many unemployed.]$_3$ [The people waiting in line carried a message, as refutation, of claims that the jobless could be employed if only they showed enough moxie.]$_4$ [Every rule has exceptions,]$_5$ [but the tragic and too-common tableaux of hundreds or even thousands of people snake-lining up for any task with a paycheck illustrates a lack of jobs,]$_6$ [not laziness.]$_7$" (Mann and Thompson, 1988)

Figure 6.2 shows an RST diagram of the text in Example 6.1. In RST, the basic units of discourse are called "Elementary Discourse Units" (EDUs) (*e.g.* text segments in Example 6.1). These EDUs are grouped hierarchically into larger units called "spans" (*e.g.* indexed nodes in Figure 6.2). A span can consist of a single EDU or multiple EDUs. For instance, in Figure 6.2, span 1 consists of only one EDU, while span 4-7 consists of four EDUs. In an RST diagram, the directed connections between two non-overlapping spans are characterised by rhetorical relations. The source span is called the "Satellite", and the target is called the "Nucleus". For example, the relation between span 6 (the Nucleus) and

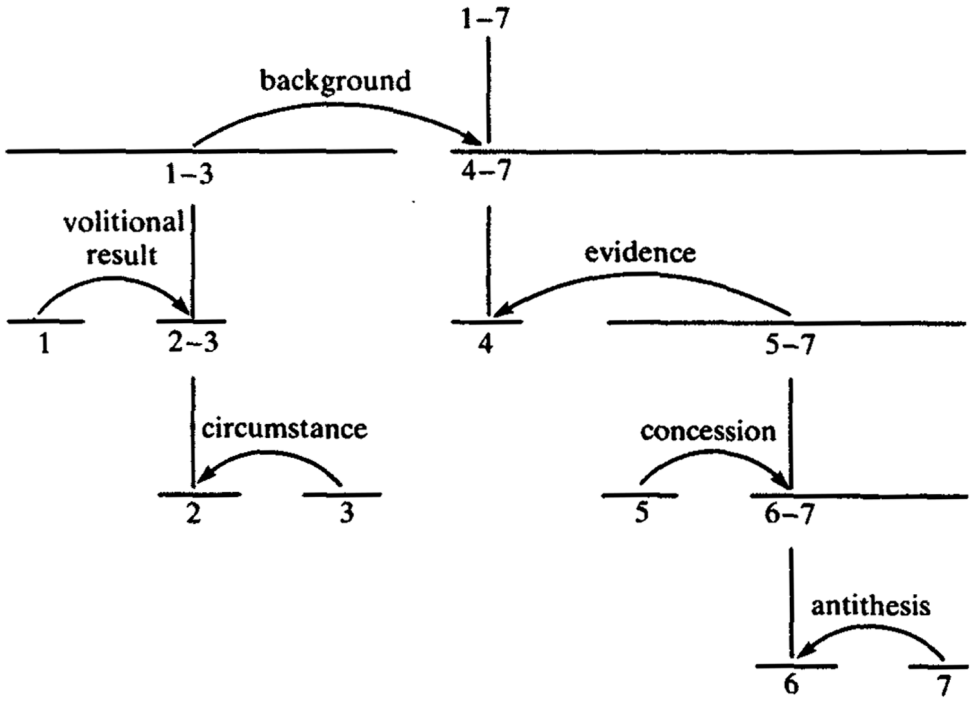


Figure 6.2: An RST diagram of the text in Example 6.1. Numbers are indices of segments. Taken from Mann and Thompson (1988).

span 7 (the Satellite) is ANTITHESIS, which means that the situation describe in span 7 contradicts to that in span 6.

While there are many rhetorical relations identified in RST, some of the central relations include:

- **JUSTIFY**: The Satellite provides a justification for some actions or positions taken in the Nucleus.
- **BACKGROUND**: The Satellite provides background or context to the Nucleus.
- **EVIDENCE**: The Satellite supports or strengthens the claims made in the Nucleus.
- **VOLITIONALRESULT**: The Satellite presents a situation or action that is voluntarily caused or brought about by the agent mentioned in the Nucleus.
- **CONCESSION**: The Satellite presents information that one would expect to contradict or oppose the Nucleus, yet the latter remains unaffected or valid despite the potential counterpoint presented by the former.

I chose to use RST discourse structures because they have been acknowledged to be useful in many NLP tasks, such as summarisation (Marcu, 2000), text classification (Burstein et al., 2001), and argument mining (Accuosto and Saggion, 2019). Moreover, some relations in the RST framework are indicative of relation categories in my annotation

scheme. For instance, in Figure 6.2, the EVIDENCE relation between span [5-7] and span 4 may indicate a SUPPORT relation between span 6 and span 4, and the CONCESSION relation between span 5 and span [6-7] can be a signal of segment 5 undercutting the relation between span 6 and span 4. However, incorporating RST discourse structures as an external knowledge source in argument mining models may introduce a potential source of noise. This is because EDUs in RST trees and argument components in argument graphs may often have different segment boundaries. Despite this challenge, I made the decision to use RST trees with the expectation that the valuable information they contain would not be overshadowed by the noise introduced.

As we can see from Figure 6.2, an RST diagram is not a typical tree structure. RST diagrams as a representation is ambiguous in that its placement of nodes indicates some form of hierarchy, but edges also exist between Nuclei and their Satellites. As a result, there are different ways to transform an RST diagram into an RST tree. One is to adhere to the placement of nodes in the RST diagram, making a Nucleus and its Satellite siblings in a branch. I call an RST tree transformed in this way a "span-oriented RST tree". Another is to adhere to the edges in the RST diagram, so that a Nucleus is the parent of its Satellite. I call it a "dependency-oriented RST tree".

By the convention in the field of RST parsing, RST diagrams are often transformed into span-oriented RST trees (Carlson et al., 2003). In a span-oriented RST tree (shown in Figure 6.3), leaves represent EDUs, and nodes represent spans. Out of the two siblings in a branch, either one is a Nucleus and one Satellites, or both are Nuclei. When an RST diagram is transformed into a span-oriented RST tree, the label of the relation between a Nucleus and its Satellite in the RST diagram is assigned to the Satellite node in the span-oriented RST tree. For instance, in Figure 6.3, node 1 and node 2-3 are sibling nodes.

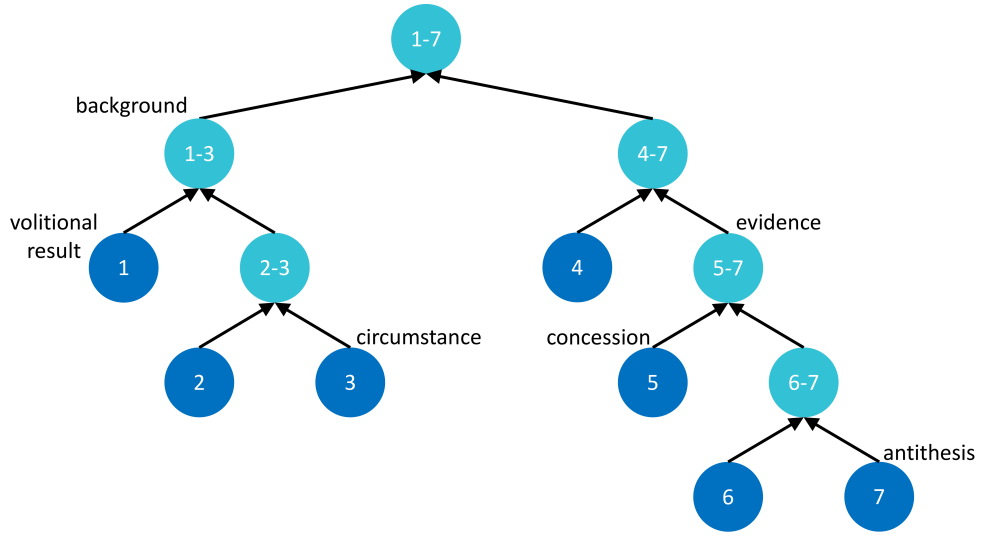


Figure 6.3: A span-oriented RST tree converted from the diagram in Figure 6.2 (nodes only).

Node 1 has the label "VolitionalResult", meaning that node 1 (the Satellite), is a situation that is voluntarily caused by node 2-3 (the Nucleus).

However, such span-oriented RST trees are not ideal for argument mining for two reasons. First, the notion of Nuclei and Satellites to some degree resembles that of heads and dependants in syntactic dependencies (Sagae, 2009). But in span-oriented RST trees, information of such dependencies is not directly available, because the Nucleus and its Satellite are siblings. Second, in span-oriented RST trees, rhetorical relations exist between sibling nodes, and hierarchical relations only indicate nestedness — a parent node is the concatenation of all its child nodes. This is very different from what an edge means in argument graphs: argument relations are shown as the hierarchical relations between a source component and is the target component.

Therefore, in order to make the rhetorical relations in an RST tree more analogous to the argument relations in an argument graph, I reconstruct dependency-oriented RST trees from span-oriented RST trees: staring from the bottom of a span-oriented RST tree, for each labelled node, its outgoing edge is redirected to its unlabelled sibling node in a dependency-oriented RST tree; the label of that node becomes the label of the new outgoing edge of that node in a dependency-oriented RST tree. Figure 6.4 shows the dependency-oriented RST tree reconstructed from the span-oriented RST tree in Figure 6.3. Dashed edges in red are original edges in the span-oriented RST tree that have been redirected. Edges in red are the corresponding new edges. In such a dependency-oriented RST tree, the dependencies between Nuclei and Satellites are directly shown. For instance, in Figure 6.4, node 6 is now the parent of node 7, and the rhetorical relation between them is now shown by the label of the edge between them.

I use the RST parser by Wang et al. (2017) to obtain the span-oriented RST tree of

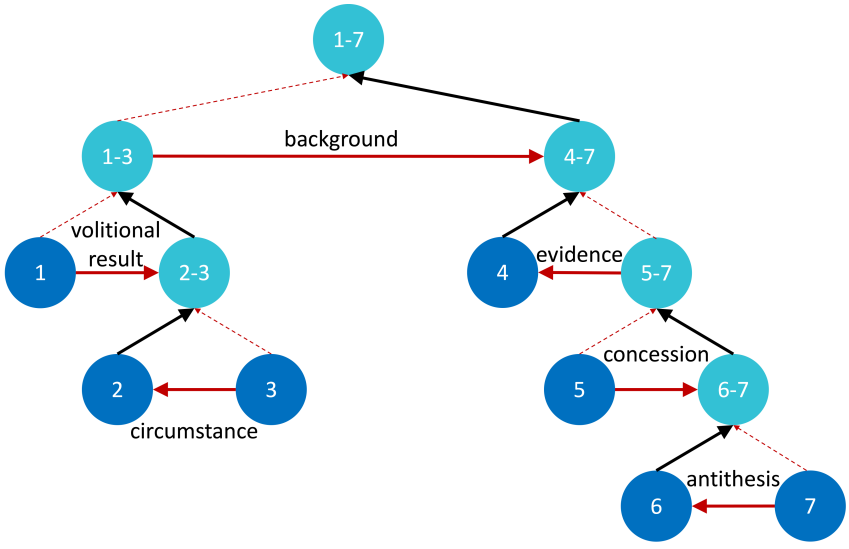


Figure 6.4: A dependency-oriented RST tree reconstructed from the span-oriented RST tree in Figure 6.3.

each document[1], and then convert it into a dependency-oriented RST tree. Since the minimal unit in an RST tree is a segment, each token in a segment shares the same discourse information. To encode the information in a dependency-oriented RST tree, I represent the path from each segment to the root node as a list of tuples in the form of $(edge\_label, start\_index$ of target node$, end_index$ of target node) along that path. As the relation between a Nucleus and its parent is left unlabelled up to now, I use a dummy label called "span" to encode these relations. For example, segment 5 has the path $[(concession, 78, 107),\ (span, 73, 107),\ (evidence, 46, 72),\ (span, 46, 107),\ (span, 1, 107),\ (root, 0, 0)]$.

6.1.1.3 Knowledge graph

Knowledge graphs encode structured information about entities, concepts, and events, as well as the relationships between them. ConceptNet-5.5 (Speer et al., 2017) is a large-scale, open-source curated knowledge graph. In ConceptNet, a node is a word or phrase of a natural language, and an edge is the relationship between two nodes. There are 36 relations in ConceptNet, including bi-directional (*i.e.* symmetrical) relations, such as LocatedNear, RelatedTo, and Synonym, and uni-directional (*i.e.* asymmetrical) relations, such as IsA, PartOf, and UsedFor. The five most frequent relations among these are:

- **RelatedTo**: A connection exists between A and B, though the specific nature of the relationship remains unspecified.
- **FormOf**: A represents an inflected version of B, with B being the base or root form.
- **IsA**: A is either a subset or a specific instance of B, implying that all instances of A fall under B.
- **PartOf**: A is a component or segment of B.
- **HasA**: B is associated with A, either as an inherent component or through a socially constructed notion of ownership.

Figure 6.5 shows a partial list of entries related to the entry "argument" in ConceptNet, together with the four types of relations between them, namely "synonyms", "related terms", "derived terms", and "types of argument". For example, from this list we can see that "clincher" is a type of argument. In ConceptNet, there are twenty types of relations occurring with "argument".

[1]Since this parser does not support segmentation, I use sentence boundaries as segment boundaries for RST parsing.

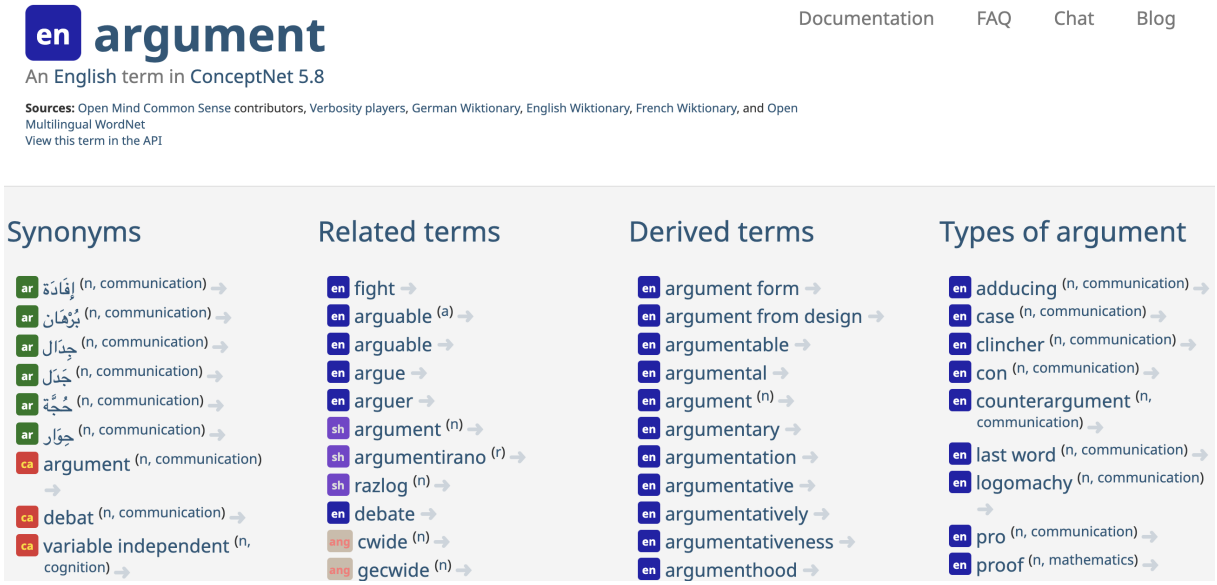


Figure 6.5: A partial list of entries related to "argument" and their relations in ConceptNet.

I decided to evaluate ConceptNet as an external knowledge source for several reasons. One, arguments often contain terms like specialised terminologies, acronyms, and cultural references that might not be explicitly explained in the data. I expect ConceptNet to support argument mining by providing relevant semantic information about such terms. Two, I believe that ConceptNet can assist in uncovering relationships between terms in arguments, especially when these connections require knowledge not evident in the given data. For example, consider the argument, "Carlos does ten minutes of jumping rope every day and his cardiovascular health has significantly improved". "Jumping rope" may appear rarely in the data, and the connection between "jumping rope" and "cardiovascular health" might not be apparent from the data alone. However, in ConceptNet we can find that jumping rope is "a type of" good exercise and it is "used for" cardiovascular health.

Word and phrase entries in ConceptNet cover multiple languages such as Arabic, English, and Spanish. Since this study targets English text and my AMDP approach operates at the token-level, I incorporate only the single-word English entries from ConceptNet. Sometimes, the meaning of a word changes according to its part-of-speech. Most entries in ConceptNet are stated in their non-disambiguated form, but some entries are disambiguated. For instance, the entry "record/n" refers to the noun sense of the word "record". I select only the non-disambiguated entry, given that my AMDP approach does not incorporate any kind of disambiguation, neither by word sense nor by part-of-speech.

I encode the information of each ConceptNet entry as a list of tuples in the form $(relation\_label, linked\_entry, edge\_direction)$. Here, $linked\_entry$ means the entry that is linked to the current entry, $relation\_label$ is the label of the relation between the two entries, and $edge\_direction$ indicates if the edge comes from the current entry, goes to the current entry, or is bi-directional. Examples of such tuples extracted from Figure 6.5 in-

clude $(debate, Synonym, bidirectional)$, $(fight, RelatedTo, bidirectional)$, $(argumentable, DerivedFrom, incoming)$, and $(adducing, IsA, incomining)$.

6.1.1.4 Pre-trained large language model

In recent years, many pre-trained large language models adopt a Transformer-based architecture, and are trained on a broad range of internet text (Radford et al., 2018, Devlin et al., 2019, Radford et al., 2019, Brown et al., 2020). Exemplified by GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023), these models have exhibited strong capabilities in Natural language Understanding (NLU) and Natural language Generation (NLG), such as opinion summarisation (Bhaskar et al., 2023), taking medical examinations (Lin et al., 2023), and writing argumentative essays (Liu et al., 2023b). Also, these models have demonstrated few-shot learning capabilities (Bommarito II and Katz, 2022, Chen et al., 2023), which means that they can generalise and perform tasks effectively with minimal training examples.

Pre-trained large language models may serve as external knowledge sources for argument mining. Their extensive training data may enable them to implicitly acquire world knowledge, linguistic patterns, and reasoning capabilities to some extent. I selected GPT-3.5 as the external knowledge source in this study, because it is the most powerful pre-trained large language model that I have free access to. In order to use GPT-3.5 as an external knowledge source, I use specifically designed prompts that simulate the argument mining task to instruct GPT-3.5 to generate responses. I use the responses to construct an argument graph for each argument, which is then encoded as features given to my parsers. The algorithm for prompting GPT-3.5 and argument graph generation is given as Algorithm 1[2]. Crucial steps in this algorithm are detailed below:

1. Lines 1-3: I designed the first prompt in the hope to find the main statements of an argument. The first prompt ($Prompt_{main}$) given to GPT-3.5 is the entire argument text followed by the question "Which exact segments in the text above are the main statements of this argument?" The segments returned are then added to the argument graph.

2. Lines 7-13: The second prompt aims to find all the supporting components of a given component. For every segment that GPT-3.5 returns in the previous step, the second prompt ($Prompt_{sup}$) is the complete argument followed by the question "Which exact segments in the text above support the claim that '[segment]'?" The segments returned are then treated as candidate supporting dependants of the input segment, if they are not already in the argument graph.

[2]As prompting strategies for large language models is not a major concern of my thesis, the sensitivity of this algorithm to the exact casting of the prompts is not studied.

Algorithm 1 Argument Mining with GPT-3.5

Require: Argument A

Initialise *ArgumentGraph*
$Prompt_{main} \leftarrow$ "*Which exact segments in the text above are the main statements of this argument?*"
$Segments \leftarrow GPT(A + Prompt_{main})$
Add *Segments* to *ArgumentGraph*
$CurrentSegments \leftarrow Segments$
for $i = 1$ to 3 **do**
  Initialise *DepSegments*
  for each *CurrentSegment* in *CurrentSegments* **do**
    $Prompt_{sup} \leftarrow$ "*Which exact segments in the text above support the claim that '*[*CurrentSegment*]*'?*"
    $SupportingSegments \leftarrow GPT(A + Prompt_{sup})$
    for each *SupportingSegment* in *SupportingSegments* **do**
      if *SupportingSegment* in *ArgumentGraph* **then**
        Remove *SupportingSegment* from *SupportingSegments*
      end if
    end for
    $Prompt_{att} \leftarrow$ "*Which exact segments in the text above attack the claim that '*[*CurrentSegment*]*'?*"
    $AttackingSegments \leftarrow GPT(A + Prompt_{att})$
    for each *AttackingSegment* in *AttackingSegments* **do**
      if *AttackingSegment* in *ArgumentGraph* **then**
        Remove *AttackingSegment* from *AttackingSegments*
      end if
      if *AttackingSegment* in *SupportingSegments* **then**
        $Prompt_{op} \leftarrow$ "*Does the claim '*[*AttackingSegment*]*' support or attack the conclusion that '*[*CurrentSegment*]*'?*"
        $OverlapCheck \leftarrow GPT(Prompt_{op})$
        if 'support' in *OverlapCheck* **then**
          Remove *AttackingSegment* from *AttackingSegments*
        else
          Remove *AttackingSegment* from *SupportingSegments*
        end if
      end if
    end for
    Add *SupportingSegments* to *ArgumentGraph* with target segment index and relation label
    Add *AttackingSegments* to *ArgumentGraph* with target segment index and relation label
    Add *SupportingSegments* to *DepSegments*
    Add *AttackingSegments* to *DepSegments*
  end for
  $CurrentSegments \leftarrow DepSegments$
end for
return *ArgumentGraph*

3. Lines 14-19: The third prompt aims to find all the attacking components of a given component. For every input segment in the previous step, the third prompt ($Prompt_{att}$) is the complete argument followed by the question "Which exact segments in the text above attack the claim that '[segment]'?" The segments returned are then treated as candidate attacking dependants of the input segment, if they are not already in the argument graph.

4. Lines 20-28: There are instances when a single segment is identified by GPT-3.5 as a supporting and attacking segment at the same time for the same input segment, which is a structure not permissible in my annotation scheme. The fourth prompt aims to solve this problem by keeping such a segment as either a supporting or attacking segment. For each segment returned in step 2 and step 3 at the same time (*i.e.* an overlapping segment), the question "Does the claim '[overlapping segment]' support or attack the conclusion '[input segment]'?" is presented as the fourth prompt ($Prompt_{op}$). The relation between the overlapping segment and the input segment is determined based on GPT-3.5's response. The overlapping segment is removed from one of the candidate sets generated in step 2 or step 3 accordingly.

5. Lines 30-33: The remaining candidate segments in step 2 are added to the argument graph as supporting dependants of the input segment. The remaining candidate segments in step 3 are added to the argument graph as attacking dependants of the input segment.

6. Steps 2-5 are iteratively conducted up to three times, or until no new segments are returned.

There is one special step that is not shown in Algorithm 1, because it is automatically applied every time GPT-3.5 returns a segment in steps 1-3. This step checks if there is an exact match between a segment returned and a corresponding segment in the original argument. This is because GPT-3.5 does not guarantee to always return exact segments in the original argument. If the segment returned is not an exact match, the original argument followed by the instruction "Please find the exact segment in the original text for '[segment]'." is presented to GPT-3.5. This step is repeated until an exact match is found. The segment initially returned is then replaced by this exact match for subsequent steps of Algorithm 1.

In an effort to exploit the few-shot learning abilities of GPT-3.5, I randomly select ten documents from the training set. These documents serve to teach the model how to respond to the prompting questions in Algorithm 1. I transformed all undercuts into direct ATTACKs in this teaching material. This is because the Persuasive Essays dataset does

not contain undercuts and I wanted all data to have a coherent structure[3].

Biases might arise from the order in which documents are presented if GPT-3.5 processes multiple documents within on session. I therefore present each document in the test set to GPT-3.5 in a separate session.

After an argument graph is generated for each document by GPT-3.5, I encode the information in the graph in a manner similar to that of the discourse information, as described in Section 6.1.1.2.

6.1.2 Transfer-based approach

In the transfer-based approach, I examine cross-domain and cross-task transfer methods. Cross-domain transfer refers to the practice of applying knowledge learned from one domain to another. Cross-task transfer involves applying knowledge learned from one task to another.

In the setting of end-to-end neural models, the integration of external knowledge begins by training models on a source task or domain. This process allows external knowledge to be stored within the pre-trained model weights. Subsequently, instead of initialising from scratch, these pre-trained weights serve as the foundation for training the model on the target task or domain.

The external knowledge used for this approach comes from an argument mining dataset distinct from the target datasets, and from two NLP tasks closely related to argument mining: syntactic parsing and discourse parsing.

6.1.2.1 Cross-domain transfer

In the context of argument mining, different domains can refer to different types of text, such as online forum debates, legal documents, and scientific literature. Domain differences can also arise from the topics of arguments, such as education, religion, and politics.

Each of these domains may have its own semantic, stylistic, and structural characteristics. For instance, online forum debates may be characterised by informal language, personal anecdotes, and emotion-driven assertions. Legal documents are more formal, follow a stringent structure, and prioritise clarity and precision. Scientific literature is often dense with domain-specific terminology and follows a systematic approach in presenting arguments, often rooted in empirical evidence. Such discrepancies in argument structure and language use across different domains can create challenges for the successful transfer of knowledge.

When transferring the knowledge in the source dataset to the target dataset, it is also important to consider the difference in annotation schemes between them. Differences in

[3]In future work, it is possible to devise a prompting strategy that teaches GPT-3.5 more specifically about undercuts, at least for the Quora section of these experiments.

annotation schemes can result in inconsistencies in how data is interpreted or labelled. For instance, a MajorClaim and a Claim in the Persuasive Essays dataset would both be labelled as Proposition under my annotation scheme. Such differences might mislead the model and lower the accuracy of its predictions.

Therefore, for successful knowledge transfer, the chosen source dataset should closely align with the target datasets, namely the Persuasive Essays dataset and the Quora dataset, in terms of argument structures, linguistic style, and annotation schemes. Given these prerequisites, I selected the Gold Standard Toulmin dataset from the datasets reviewed in Section 3.1 as the source for cross-domain transfer. While the Microtext dataset was a promising contender, its limited size, which is only 7,846 tokens, made me disregard it.

In the Gold Standard Toulmin dataset, argument relations are not labelled, but Habernal and Gurevych (2017) describe a procedure on how they can be inferred:

- Backings are not related to any other components.
- The relation between a Premise and a Claim is always a Support.
- The relation between a Rebuttal and a Claim is always an Attack.
- The relation between a Refutation and a Rebuttal is always an Attack.

After establishing these relations, I use this dataset to pre-train my model from scratch. The pre-trained model wights are then used to initialise the model trained on the target datasets.

6.1.2.2 Cross-task transfer

For cross-task transfer, the source task and the target task should bear some resemblance to each other. More importantly, they should be solvable using the same neural model.

I chose syntactic parsing and discourse parsing as the source tasks for cross-task transfer. In Sections 6.1.1.1 and 6.1.1.2, I have already discussed the connection between syntactic parsing and argument mining, as well as the connection between discourse parsing and argument mining. Syntactic parsing provides information on the structure of sentences, and discourse parsing on that of documents. I expect these two tasks to help argument mining, a task which is similar in spirit.

The datasets for the source task and the target task usually originate from the same underlying raw text. In order to perform cross-task transfer for the Persuasive Essays dataset and the Quora dataset, I therefore need datasets with syntactic and discourse parses of their text. To achieve this, I use the Stanford CoreNLP syntactic dependency parser to create syntactic parse trees for the two dataset, and the RST parser by Wang et al. (2017) to generate discourse parse trees.

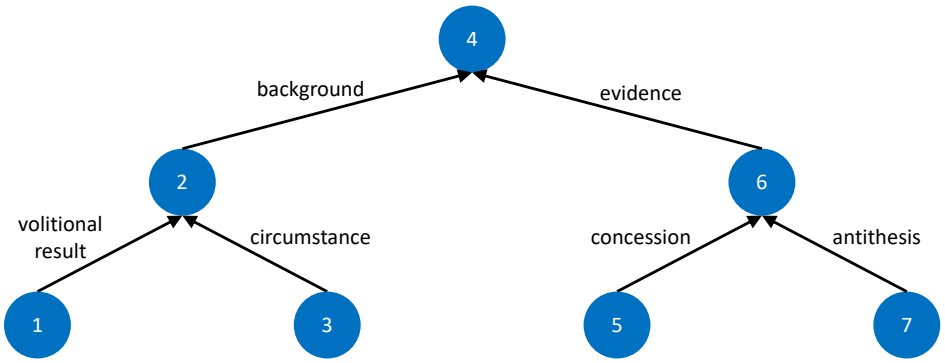


Figure 6.6: A new discourse parse tree reconstructed from the dependency-oriented RST tree in Figure 6.4.

When adapting syntactic parse trees for my neural parser, I regard all syntactic parse trees (one per sentence) in a document together a graph. Such a graph composed of many isolated subgraphs is one data point for the parser.

It is not as straightforward to adapt RST trees to my neural parser. Besides nodes that contain just one EDU, RST trees also include nodes that contain multiple EDUs. My neural parser does not support this structure, as it is only capable of predicting relationships between individual segments, not between a single segment and a unit comprising multiple segments. I therefore have to exclude such nodes from RST trees, but in such a way that the loss of information is minimised. I designed an method to automatically transform a span-oriented RST tree into a new discourse parse tree in which each node only contains one EDU. Figure 6.6 shows the new tree reconstructed from the dependency-oriented RST tree in Figure 6.4. When describing this method, I refer to a node that only contains one EDU as a single-EDU node (*e.g.* node 1 in Figure 6.4), a node containing multiple EDUs as a multi-EDU node (*e.g.* node [2-3]). The method is as follows:

- I define the "core span" of a multi-EDU node as the first single-EDU node encountered when traversing from the multi-EDU node down the tree, while bypassing any labelled edges. For instance, the core span of node [1-3] is node 2. The core span of a single-EDU node is itself.

- I define the "temporal parent" of any node as the node reached by traversing from the initial node up the tree, after a labelled edge is encountered. The label of this labelled edge also becomes the "temporal edge label" between the source node and its temporal parent. For instance, the temporal parent of node 2 is node [4-7], and the temporal edge label between node 2 and node [4-7] is BACKGROUND. The temporal parent of node 6 is node 4, and the temporal edge label is EVIDENCE.

- A new discourse parse tree is constructed as follows: for each single-EDU node in the original dependency-oriented RST tree, its parent in the new discourse parse tree is the core span of its temporal parent in the original dependency-oriented RST tree; the label of the edge between that single-EDU node and its parent in the new discourse parse is the same as the temporal edge label between the source node and its temporal parent in the original dependency-oriented RST tree.

After these adaptations, I use these adapted syntactic parse trees and discourse parse trees to pre-train my model from scratch. The pre-trained model wights are then used to initialise the model trained on the target datasets.

6.2 Experiment

In this section, I describe an experiment carried out to evaluate the feature-based and transfer-based approaches described earlier. The goal is to explore the impact of diverse knowledge sources and integration techniques on formal versus informal arguments. Additionally, I explore their effect on undercut recognition, aiming to pinpoint which sources and techniques stand out in their effectiveness for this specific recognition task.

6.2.1 Systems

In this experiment, I use my the GNN-based dependency parser in conjunction with my undercut-inclusive dependency representation as the baseline. Specifically, there are two models within this baseline: one trained on the Persuasive Essays dataset (*i.e. GNN_inc* in Table 5.4) and another on the Quora dataset (*i.e. GNN_inc* in Table 5.8).

For the feature-based approach, I investigate the four kinds of features described in Sections 6.1.1.1-6.1.1.4:

- *Syn*: Using syntactic information as additional features.
- *Dis*: Using discourse information as additional features.
- *CN*: Using ConceptNet as additional features.
- *GPT*: Using GPT-3.5 as additional features.

For the transfer-based approach, I experiment with three transfer techniques:

- *GST*: Cross-domain transfer using the Gold Standard Toulmin dataset as the source dataset.
- *SynPar*: Cross-task transfer using syntactic parsing as the source task.
- *DisPar*: Cross-task transfer using discourse parsing as the source task.

6.2.2 Experimental setup

To integrate features into the neural parser, I first project each kind of features with an LSTM layer, and then combine the projected tensors with the BERT-encoded representations of the input text through point-wise addition.

During transfer-based integration, I use the fine-tuning strategy in Howard and Ruder (2018) to fine-tuning the GNN-based parser on the target datasets and task. The BERT encoder in the GNN-based parser is kept static during fine-tuning.

In all other aspects, the experimental setup follows that from Sections 5.3 and 5.4.

6.2.3 Results

Results of the experiment on the Persuasive Essays dataset are presented in Table 6.1 and those on the Quora dataset in Table 6.3. Results for the corresponding permutation tests are shown in Table 6.2 and Table 6.4.

In terms of the overall performance, feature-based knowledge integration with GPT-3.5 significantly (28 paired permutation tests, all $p < 0.01$) outperforms all other methods on both datasets and for both tasks. Specifically, *GPT* achieves an F_1 score of 75.3% for component identification and 50.5% for relation identification on the Persuasive Essays dataset. On the Quora dataset, *GPT* reaches an F_1 score of 69.2% for components and 48.5% for relations.

Regarding the effectiveness of diverse external knowledge sources within the feature-

		Feature-based				Transfer-based		
	Baseline	Syn	Dis	CN	GPT	GST	SynPar	DisPar
Component	73.8	74.4	75.0	74.2	75.3	73.3	74.6	74.9
Relation	49.4	49.8	50.1	49.6	50.5	48.9	49.7	50.2

Table 6.1: Results for external knowledge integration on the Persuasive Essays dataset.

	Baseline	Syn	Dis	CN	GPT	GST	SynPar	DisPar
Baseline	-	<	<	<	<	>	<	<
Syn	>	-	<	>	<	>	=	<
Dis	>	>	-	>	<	>	>	=
CN	>	<	<	-	<	>	<	<
GPT	>	>	>	>	-	>	>	>
GST	<	<	<	<	<	-	<	<
SynPar	>	=	<	=	<	>	-	<
DisPar	>	>	=	>	<	>	>	-

Table 6.2: Results for the permutation tests regarding Tables 6.1. P-values in blue are for component identification, red for relation identification.

		Feature-based				Transfer-based		
	Baseline	Syn	Dis	CN	GPT	GST	SynPar	DisPar
Component	66.2	67.8	68.0	67.0	69.2	65.7	67.4	68.6
Relation	45.8	47.0	47.5	46.4	48.5	45.6	46.6	47.9

Table 6.3: Results for external knowledge integration on the Quora dataset.

	Baseline	Syn	Dis	CN	GPT	GST	SynPar	DisPar
Baseline	-	<	<	<	<	>	<	<
Syn	>	-	<	=	<	>	>	<
Dis	>	>	-	>	<	>	>	<
CN	>	<	<	-	<	>	<	<
GPT	>	>	>	>	-	>	>	>
GST	<	<	<	<	<	-	<	<
SynPar	>	<	<	=	<	>	-	<
DisPar	>	>	>	>	<	>	>	-

Table 6.4: Results for the permutation tests regarding 6.3. P-values in blue are for component identification, red for relation identification.

based approach, each knowledge source brings significant ($p < 0.01$) improvement to the baseline. The results suggest a pattern: GPT-3.5 consistently leads in effectiveness, followed in order by discourse information, syntactic information, and ConceptNet. Among these knowledge sources, we can see that:

- The knowledge from GPT-3.5 leads to the biggest improvement for component identification. Compared to the baseline, *GPT* excels by 2.5% on the Persuasive Essays dataset ($GPT = 75.3\%$, $baseline = 73.8\%$, $p < 0.01$) and 3.0% on the Quora dataset ($GPT = 69.2\%$, $baseline = 66.2\%$, $p < 0.01$). It is also the best-performing system for relation identification, with an increase of 1.1% on the Persuasive Essays dataset ($GPT = 50.5\%$, $baseline = 49.4\%$, $p < 0.01$) and 2.7% on the Quora dataset ($GPT = 48.5\%$, $baseline = 45.8\%$, $p < 0.01$). This success might be attributed to GPT-3.5's ability in few-shot learning and in using context, and the vast array of world knowledge embedded in its training data.

- Discourse information appears to be the second most effective knowledge source. For component identification, *Dis* achieves a 1.2% increase over the baseline on the Persuasive Essays dataset ($Dis = 75.0\%$, $baseline = 73.8\%$, $p < 0.01$) and 1.8% on the Quora dataset ($Dis = 68.0\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the increase is 0.7% ($Dis = 50.1\%$, $baseline = 49.4\%$, $p < 0.01$) and 1.7% ($Dis = 47.5\%$, $baseline = 45.8\%$, $p < 0.01$).

- Syntactic information ranks third. For component identification, *Syn* outperforms the

baseline by 0.6% on the Persuasive Essays dataset ($Syn = 74.4\%$, $baseline = 73.8\%$, $p < 0.01$) and 1.6% on the Quora dataset ($Syn = 67.8\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the increase is 0.4% ($Syn = 49.8\%$, $baseline = 49.4\%$, $p < 0.01$) and 1.2% ($Syn = 47.0\%$, $baseline = 45.8\%$, $p < 0.01$).

- The improvement introduced by ConceptNet is the least, but it is still significant. For component identification, *CN* outperforms the baseline by 0.4% on the Persuasive Essays dataset ($CN = 74.2\%$, $baseline = 73.8\%$, $p < 0.01$) and 0.8% on the Quora dataset ($CN = 67.0\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the increase is 0.2% ($CN = 49.6\%$, $baseline = 49.4\%$, $p < 0.01$) and 0.6% ($CN = 46.4\%$, $baseline = 45.8\%$, $p < 0.01$). The limited effect size could be due to the discrepancy between what the knowledge ConceptNet provides and what is required by the target datasets. For example, while the Quora dataset contains many names of actual persons, most of them do not have corresponding entries in ConceptNet.

Discourse information and syntactic information both provide information about the structure of text. Comparing these two knowledge sources, I find that in the feature-based approach discourse information is significantly more effective than syntactic information. For component identification, *Dis* outperforms *Syn* by 0.6% on the Persuasive Essays dataset ($Dis = 75.0\%$, $Syn = 74.4\%$, $p < 0.01$) and 0.2% on the Quora dataset ($Dis = 68.0\%$, $Syn = 67.8\%$, $p < 0.01$). For relation identification, the increase is 0.3% ($Dis = 50.1\%$, $Syn = 49.8\%$, $p < 0.01$) and 0.5% ($Dis = 47.5\%$, $Syn = 47.0\%$, $p < 0.01$). This difference in performance could be attributed to two potential reasons. Firstly, the scope of information in a discourse parse tree is the entire document, while for a syntactic parse tree it is only a sentence. When it comes to identifying how components are hierarchically structured to form an argument, the global view of discourse information may be more effective. Secondly, most relations in an argument stretch across sentences. Therefore, the relations in syntactic parsing, which only occur within sentences, might not be that useful for relation identification in argument mining.

The two cross-task transfer methods significantly outperform the baseline. Among the two source tasks used in cross-task transfer, we can see that:

- Discourse parsing as a source task has the biggest improvement over the baseline. For component identification, *DisPar* outperforms the baseline by 1.1% on the Persuasive Essays dataset ($DisPar = 74.9\%$, $baseline = 73.8\%$, $p < 0.01$) and 1.6% on the Quora dataset ($DisPar = 68.6\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the increase is 0.8% ($DisPar = 50.2\%$, $baseline = 49.4\%$, $p < 0.01$) and 1.9% ($DisPar = 47.9\%$, $baseline = 45.8\%$, $p < 0.01$).

- Syntactic parsing as a source task can also improve argument mining. For component identification, *SynPar* outperforms the baseline by 0.8% on the Persuasive Essays

dataset ($SynPar = 74.6\%$, $baseline = 73.8\%$, $p < 0.01$) and 1.6% on the Quora dataset ($SynPar = 67.4\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the increase is 0.3% ($SynPar = 49.7\%$, $baseline = 49.4\%$, $p < 0.01$) and 0.8% ($SynPar = 46.6\%$, $baseline = 45.8\%$, $p < 0.01$).

Comparing the two source tasks, I find that discourse parsing is significantly more effective than syntactic parsing as a source task. For component identification, *DisPar* outperforms *SynPar* by 0.3% on the Persuasive Essays dataset ($DisPar = 74.9\%$, $SynPar = 74.6\%$, $p < 0.01$) and 1.2% on the Quora dataset ($DisPar = 68.6\%$, $SynPar = 67.4\%$, $p < 0.01$). For relation identification, the increase is 0.5% ($DisPar = 50.2\%$, $SynPar = 49.7\%$, $p < 0.01$) and 1.3% ($DisPar = 47.9\%$, $SynPar = 46.6\%$, $p < 0.01$). This finding is in line with my previous finding from feature-based integration that discourse information as an additional feature is more useful than syntactic information. All of this suggests that knowledge about the structure of an entire document can be particularly effective for argument mining.

Compared to the baseline, cross-domain transfer with the Gold Standard Toulmin dataset as the source dataset significantly lowers performance. For component identification, *GST* shows a decline of 0.5% compared to the baseline on the Persuasive Essays dataset ($GST = 73.3\%$, $baseline = 73.8\%$, $p < 0.01$) and 0.5% on the Quora dataset ($GST = 65.7\%$, $baseline = 66.2\%$, $p < 0.01$). For relation identification, the decrease is 0.5% ($GST = 48.9\%$, $baseline = 49.4\%$, $p < 0.01$) and 0.2% ($GST = 45.6\%$, $baseline = 45.8\%$, $p < 0.01$). This result implies that the Gold Standard Toulmin dataset might have introduced too much noise during knowledge transfer. The noise may stem from the disparity in annotation schemes between the source dataset and the two target datasets. This problem appears to have overshadowed the potentially useful knowledge offered by the Gold Standard Toulmin dataset.

Regarding the different impact of external knowledge on formal versus informal text, the Quora dataset benefits more from external knowledge than the Persuasive Essays dataset does. For component identification, the average improvement of all the knowledge integration systems over the baseline is 0.7% on the Persuasive Essays dataset and 1.4% on the Quora dataset. For relation identification, the average improvement is 0.4% and 1.3%. The information contained in the more formal Persuasive Essays dataset may already be in the form that models can exploit more easily. In contrast, the informal nature of the Quora dataset, coupled with its diverse styles and potentially less clear argument structures, could result in a greater profit from external knowledge.

In conclusion, most of the methods investigated in this chapter are effective for argument mining. GPT-3.5 emerges as a standout feature-based knowledge source. Discourse information and syntactic information are also useful as either additional features or source tasks in cross-task transfer. Knowledge transfer from the Gold Standard Toulmin

dataset is not successful, which indicates the difficulties in cross-domain adaptation. The results further suggest distinctions between formal and informal arguments, with the latter benefiting more from external insights.

6.2.4 Influence of external knowledge integration on undercuts

Considering the importance of undercuts for this thesis, I will now look in more details at the specific results related to undercuts in the Quora dataset.

Table 6.5 shows the F_1 scores for undercut recognition. Results for the corresponding permutation tests are given in Table 6.6.

The results show that discourse information as an additional feature leads to the most prominent increase over the baseline. As the best-performing system, *Dis* significantly outperforms the baseline by 6.6% ($Dis = 45.4\%$, $baseline = 38.8\%$, $p < 0.01$). Feature-based integration with syntactic information (*i.e. Syn*) also significantly outperforms the baseline by 3.7% ($Syn = 42.5\%$, $baseline = 38.8\%$, $p < 0.01$).

The improvement brought by ConceptNet as an additional feature (*i.e. CN*) or syntactic parsing as a source task (*i.e. SynPar*) over the baseline is not significant.

The remaining methods, namely the Gold Standard Toulmin dataset as a source dataset (*i.e. GST*), GPT-3.5 as an additional feature (*i.e. GPT*), and discourse parsing as a source task (*i.e. DisPar*), are all significantly worse than the baseline. *GST* achieves an F_1 score significantly lower than the baseline by 5.9% ($GST = 32.9\%$, $baseline = 38.8\%$, $p < 0.01$). For *GPT* and *DisPar*, the decreases are 8.1% ($GPT = 30.7\%$, $baseline = 38.8\%$, $p < 0.01$) and 9.7% ($DisPar = 29.1\%$, $baseline = 38.8\%$, $p < 0.01$). This indicates that

		Feature-based			Transfer-based		
Baseline	Syn	Dis	CN	GPT	GST	SynPar	DisPar
38.8	42.5	45.4	39.1	30.7	32.9	40.4	29.1

Table 6.5: Results for undercut recognition on the Quora dataset using various external knowledge integration methods.

	Baseline	Syn	Dis	CN	GPT	GST	SynPar
Syn	>	-	-	-	-	-	-
Dis	>	>	-	-	-	-	-
CN	=	<	<	-	-	-	-
GPT	<	<	<	<	-	-	-
GST	<	<	<	<	>	-	-
SynPar	=	=	<	=	>	>	-
DisPar	<	<	<	<	=	<	<

Table 6.6: Results for the permutation tests regarding Table 6.5.

these methods might not be suitable for undercut recognition.

An interesting point is the difference in performance between *Dis* and *DisPar*. While both methods incorporate discourse information, *Dis* significantly outperforms *DisPar* by a margin of 16.3% ($Dis = 45.4\%$, $DisPar = 29.1\%$, $p < 0.01$). This disparity might be related to my adaptation of RST trees when I accommodate my neural parser. During this adaptation, all segments initially targeting multi-segment spans were redirected to single-segment spans, as I have described in Section 6.1.2.2. If some of these segments happen to be undercutting segments, pre-training with such adapted RST trees may confuse the parser in terms of undercut recognition. This leaves the challenge for future studies to effectively use discourse parsing for cross-task transfer.

Another observation is the difference in *GPT*'s performance between addressing the overall task and addressing undercut recognition specifically. *GPT* outperforms all other systems in both component and relation identification, but when it comes to undercut recognition, its performance is much lower than the baseline. I assume this might be caused by the constraints of my prompting strategy. The prompting strategy only asks GPT-3.5 to find the supporting and attacking segments, so that no undercuts can possibly be correctly recognised by GPT-3.5. Out of the 326 undercutting components in the Quora dataset, GPT-3.5 misclassifies 191 as direct attacking components[4]. These incorrect classifications could potentially confuse the parser when in terms of distinguishing between undercuts and direct ATTACKs.

6.3 Chapter summary

In this chapter, I experimented with various methods for model-agnostic external knowledge integration, in order to evaluate their influence on end-to-end argument mining.

In the feature-based approach, I incorporated knowledge from multiple sources as additional features. These knowledge sources include syntactic information, discourse information, ConceptNet, and GPT-3.5. In the transfer-based approach, I used the Gold Standard Toulmin dataset as the source dataset for cross-domain transfer, as well as syntactic parsing and discourse parsing as the source tasks for cross-task transfer.

In the experiment on the Persuasive Essays dataset and the Quora dataset, we saw that most of these methods significantly outperforms the baseline. Among the methods, feature-based knowledge integration with GPT-3.5 is most effective. Cross-domain transfer with the Gold Standard Toulmin dataset is the only method that significantly lags behind the baseline. I also found that the Quora dataset benefits more from external knowledge than the Persuasive Essays dataset does.

[4]GPT-3.5 does not predict precisely very often in terms of segment boundaries. Therefore, when compiling this number, I declare a match if a segment overlaps with at least 50% of the tokens in the corresponding gold standard segment.

When analysing the specific results on undercuts in the Quora dataset, I found that only two methods significantly outperform the baseline, namely feature-based integration with discourse information and with syntactic information.

The results in this chapter, together with those in Chapters 4 and 5, set the stage for the final chapter, where I will draw overarching conclusions, discuss the limitations of my thesis, and explore potential avenues for future research.

Chapter 7

Conclusions and future work

In this thesis, I have addressed the challenge of argument mining, with a specific focus on informal arguments in online discussions. The driving force behind this research is to foster better understanding among diverse parties engaged in controversial topics, by developing methods capable of automatically analysing the vast amount of user-generated text in online discussions. In pursuit of the research objectives, I have designed an annotation scheme for informal arguments, proposed an effective end-to-end approach to argument mining, and carried out an extensive investigation into the integration of external knowledge for argument mining.

7.1 Contributions

The contributions of this thesis are four-fold:

1. **Designing an annotation scheme for informal arguments in online discussions.** I have designed an annotation scheme that is both simple and informative, tailored specifically for informal arguments in online discussions. This scheme goes beyond most existing models by explicitly marking undercuts, thus capturing the deeper structural nuances of arguments. The argument components in the scheme were designed to reflect persuasion patterns within arguments. To test this scheme, I have collected 400 texts from Quora, an online platform for rational discussions on controversial topics. I have provided a layer of annotation using two trained annotators. My annotation study indicates good inter-annotator agreement, implying the practicability of the annotation scheme and the reliability of the resultant dataset.

2. **Proposing an effective end-to-end approach to argument mining.** My novel end-to-end approach to argument mining is based on dependency parsing. I have created two new dependency representations for arguments, and applied two neural parsers to the task. This approach is the first end-to-end approach applied

to informal arguments in online discussions. When applied to a more traditional benchmark containing formal arguments, it establishes a new state of the art.

3. **Computational modelling of undercuts.** An undercut represents a relation that exists between a component and another relation. Undercuts have rarely been investigated in the field of argument mining and have yet to be computationally modelled by automatic approaches. This thesis presents the first computational modelling of undercuts. One of the above-mentioned dependency representations I have created allows existing neural models to handle undercuts directly. I have evaluated my method on the Quora dataset which contains a substantial amount of undercuts.

4. **Investigating external knowledge integration for argument mining.** I have investigated various integration techniques and external knowledge sources for integrating external knowledge into end-to-end argument mining models. I have proposed feature-based and transfer-based approaches for integration, and have explored different types of knowledge sources: syntax, discourse, knowledge graphs, and large language models. These integration techniques and knowledge sources have been evaluated on both formal and informal arguments, and most of them have boosted the performance of argument mining models.

7.2 Limitations and future work

In this section, I present an outlook into the future for argument mining with text in online discussions in particular and for argument mining in general, drawing from the findings of this thesis.

Direct adaption of pre-trained large language models for argument mining. In my investigation into external knowledge integration, GPT-3.5, a pre-trained large language model, emerges as the most potent external knowledge source for argument mining out of those I have considered. In this thesis, it is employed to generate argument graphs using a structured prompting strategy. It would be advantageous to directly adapt pre-trained large language models for argument mining, thereby fully exploiting their robust capabilities in reasoning, few-shot learning, and the integration of a wide range of common sense knowledge. For instance, the dependency representations for arguments proposed in this thesis could be serialised into sequences, which could then be directly generated by pre-trained large language models.

Datasets with more balanced component categories. My annotation scheme includes

four types of components. I designed those categories because of their ability to reflect persuasion patterns and their prevalence during my manual analysis on Quora texts. In the actual annotation of the Quora dataset, STAKEs are relatively rare, so I had to merge them with ANECDOTEs for statistical analysis in the experiments. I hypothesise that these component categories may appear more frequently in topics closely related to personal experience. Future studies can include more arguments in such topics that empirically have more STAKEs, so as to achieve more balanced component categories in the dataset.

Expanding sources and advancing techniques for external knowledge integration. The methods for external knowledge integration explored in this thesis are confined to model-agnostic methods. I chose this approach because I wanted to concentrate on the impact of external knowledge sources and integration techniques, rather than their interaction with a specific argument mining model. However, this focus also restricts the selection of external knowledge sources and integration techniques. Consequently, future research could expand the range of sources and explore more advanced techniques to enhance argument mining.

Investigating undercuts as a device for expressing value-based disagreement. Besides their unique structural characteristic, I was particularly interested in undercuts because I have a hypothesis that they often express disagreement on values. Not all disagreements arise from a divergence on basic facts: many are based on differing interpretations of how these facts lead to a conclusion. This type of attacking strategy aligns with the structural characteristics of undercuts, making undercuts a potential subject for investigation in the study of value-based disagreement. The Quora dataset, with its relatively frequently occurring undercuts, and the ability of my argument mining approach to effectively identify them, can provide a solid foundation for this line of research in the future.

Improving external knowledge integration for enhanced undercut recognition. Most knowledge integration methods I investigated can effectively improve argument mining. However, when it comes to undercut recognition, some methods not only fall short of enhancing performance but on the contrary even impair it. Notably, while GPT-3.5 and discourse information boost argument mining in general, they hinder undercut recognition. I suppose this is due to the limitations in my integration strategy, which lacks specific considerations for undercuts. It would be desirable if future work could develop improved integration strategies that specifically cater to the intricacies of undercut recognition.

Argument mining with multiple arguments. It is my vision to establish a form of argument mining that can effectively summarise multiple arguments on the same topic. I

hope it can help people understand ideas of different parties by distilling and comparing their arguments. In this thesis, I only studied argument mining with single arguments, which I consider the basis for multi-document argument mining. Future studies can build upon this work to develop methods that can aggregate and contrast various arguments from multiple sources.

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452, 2016.

Patrick Abels, Zahra Ahmadi, Sophie Burkhardt, Benjamin Schiller, Iryna Gurevych, and Stefan Kramer. Focusing knowledge-based graph argument mining via topic modeling. *arXiv preprint arXiv:2102.02086*, 2021.

Pablo Accuosto and Horacio Saggion. Transferring knowledge from discourse to arguments: A case study with scientific abstracts. In *Proceedings of the 6th Workshop on Argument Mining*, pages 41–51, 2019.

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128, 2017.

Khalid Al Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. Cross-domain mining of argumentative text through distant supervision. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1395–1404, 2016a.

Khalid Al Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. A news editorial corpus for mining argumentation strategies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443, 2016b.

Rie Kubota Ando, Tong Zhang, and Peter Bartlett. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11), 2005.

Aristotle. *On rhetoric: A theory of civic discourse. Translated by G. A. Kennedy.* Cambridge University Press, Cambridge, 2006.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, pages 722–735. Springer, 2007.

Sharon Bailin and Mark Battersby. *Reason in the balance: An inquiry approach to critical thinking*. Hackett Publishing, 2016.

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, 2021.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

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

Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.

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

Philippe Besnard and Anthony Hunter. Argumentation based on classical logic. *Argumentation in artificial intelligence*, pages 133–152, 2009.

Adithya Bhaskar, Alex Fabbri, and Greg Durrett. Prompted opinion summarization with gpt-3.5. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9282–9300, 2023.

Muhammad Mahad Afzal Bhatti, Ahsan Suheer Ahmad, and Joonsuk Park. Argument mining on twitter: A case study on the planned parenthood debate. In *Proceedings of the 8th Workshop on Argument Mining*, pages 1–11, 2021.

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

Bernd Bohnet and Joakim Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1455–1465, 2012.

Filip Boltužić and Jan Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, 2014.

Michael Bommarito II and Daniel Martin Katz. Gpt takes the bar exam. *arXiv preprint arXiv:2212.14402*, 2022.

Francis Bond and Ryan Foster. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, 2013.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jill Burstein, Daniel Marcu, Slava Andreyev, and Martin Chodorow. Towards automatic classification of discourse elements in essays. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 98–105, 2001.

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

Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. Ampersand: Argument mining for persuasive online discussions. *arXiv preprint arXiv:2004.14677*, 2020.

Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1725–1735, 2020.

Yulong Chen, Yang Liu, Ruochen Xu, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Yue Zhang. Unisumm and summzoo: Unified model and diverse benchmark for few-shot summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12833–12855, 2023.

Carlos Chesnevar, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, Steven Willmott, et al. Towards an argument interchange format. *The knowledge engineering review*, 21(4):293–316, 2006.

Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

Kevin Coe, Kate Kenski, and Stephen A Rains. Online and uncivil? patterns and determinants of incivility in newspaper website comments. *Journal of communication*, 64(4):658–679, 2014.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

Mary Dalrymple, Stuart M Shieber, and Fernando CN Pereira. Ellipsis and higher-order unification. *Linguistics and philosophy*, 14:399–452, 1991.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *Proceedings of the International Conference on Learning Representations*, 2017.

Timothy Dozat and Christopher D Manning. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, 2018.

Rory Duthie, John Lawrence, Katarzyna Budzynska, and Chris Reed. The cass technique for evaluating the performance of argument mining. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 40–49, 2016.

Subhabrata Dutta, Jeevesh Juneja, Dipankar Das, and Tanmoy Chakraborty. Can unsupervised knowledge transfer from social discussions help argument mining? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7774–7786, 2022.

Meyer Dwass. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pages 181–187, 1957.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, 2015.

Jack Edmonds. Optimum branchings. *Journal of Research of the national Bureau of Standards*, 71(4):233–240, 1967.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, 2017.

Harold D Fishbein. *Peer prejudice and discrimination: The origins of prejudice*. Psychology Press, 2014.

Susan T Fiske. Stereotyping, prejudice, and discrimination. 1998.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

Austin J Freeley and David L Steinberg. *Argumentation and debate*. Cengage Learning, 2013.

James B Freeman. *Dialectics and the macrostructure of arguments: A theory of argument structure*. de Gruyter, 1991.

James B Freeman. *Argument Structure:: Representation and Theory*, volume 18. Springer Science & Business Media, 2011.

Michael Fromm, Evgeniy Faerman, and Thomas Seidl. Tacam: topic and context aware argument mining. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 99–106, 2019.

Andrea Galassi, Marco Lippi, and Paolo Torroni. Multi-task attentive residual networks for argument mining. *arXiv preprint arXiv:2102.12227*, 2021.

Debela Gemechu and Chris Reed. Decompositional argument mining: A general purpose approach for argument graph construction. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 516–526. Association for Computational Linguistics, 2019.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the first workshop on argumentation mining*, pages 39–48, 2014.

Ronald L Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

Nancy Green, Kevin D Ashley, Diane Litman, Chris Reed, and Vern Walker. Proceedings of the first workshop on argumentation mining. In *Proceedings of the First Workshop on Argumentation Mining*, 2014.

Nancy L Green. Representation of argumentation in text with rhetorical structure theory. *Argumentation*, 24:181–196, 2010.

Wayne Grennan. *Informal logic: Issues and techniques*. McGill-Queen's Press-MQUP, 1997.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

Ivan Habernal and Iryna Gurevych. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2127–2137, 2015.

Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179, 2017.

Arthur Claude Hastings. *A Reformulation of the Modes of Reasoning in Argumentation*. Northwestern University, 1962.

Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. Qt30: A corpus of argument and conflict in broadcast debate. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 3291–3300. European Language Resources Association (ELRA), 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Yufang Hou and Charles Jochim. Argument relation classification using a joint inference model. In *Proceedings of the 4th Workshop on Argument Mining*, pages 60–66, 2017.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Neslihan Iskender, Robin Schaefer, Tim Polzehl, and Sebastian Möller. Argument mining in tweets: Comparing crowd and expert annotations for automated claim and evidence detection. In *Natural Language Processing and Information Systems: 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021, Saarbrücken, Germany, June 23–25, 2021, Proceedings*, pages 275–288. Springer, 2021.

Tao Ji, Yuanbin Wu, and Man Lan. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2475–2485, 2019.

Manfred Kienpointner. *How to classify arguments*. International Soc. for the Study of Argumentation (ISSA), 1992.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.

Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11, 2015.

Klaus Krippendorff. Measuring the reliability of qualitative text analysis data. *Quality and quantity*, 38:787–800, 2004.

Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage publications, 2018.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

John Lawrence and Chris Reed. Mining argumentative structure from natural language text using automatically generated premise-conclusion topic models. In *Proceedings of the 4th Workshop on Argument Mining*, pages 39–48, 2017.

John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2020.

John C Lin, David N Younessi, Sai S Kurapati, Oliver Y Tang, and Ingrid U Scott. Comparison of gpt-3.5, gpt-4, and human user performance on a practice ophthalmology written examination. *Eye*, pages 1–2, 2023.

Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, 2019a.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023a.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*, 2023b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Luca Lugini and Diane Litman. Argument component classification for classroom discussions. In *Proceedings of the 5th Workshop on Argument Mining*, pages 57–67, 2018.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8449–8456, 2020.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, 2016.

William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven 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.

Daniel Marcu. *The theory and practice of discourse parsing and summarization*. MIT press, 2000.

Marco Marozzi. Some remarks about the number of permutations one should consider to perform a permutation test. *Statistica*, 64(1):193–201, 2004.

André FT Martins, Mário AT Figueiredo, Pedro MQ Aguiar, Noah A Smith, and Eric P Xing. Ad3: Alternating directions dual decomposition for map inference in graphical models. *The Journal of Machine Learning Research*, 16(1):495–545, 2015.

Tobias Mayer, Elena Cabrio, and Serena Villata. Transformer-based argument mining for healthcare applications. In *ECAI 2020*, pages 2108–2115. IOS Press, 2020.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 523–530, 2005.

Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5: 64–67, 2001.

Stefano Menini, Federico Nanni, Simone Paolo Ponzetto, and Sara Tonelli. Topic-based agreement and disagreement in us electoral manifestos. Association for Computational Linguistics, 2017.

Hugo Mercier and Dan Sperber. Why do humans reason? arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74, 2011.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Farjana Sultana Mim, Naoya Inoue, Shoichi Naito, Keshav Singh, and Kentaro Inui. Lpattack: A feasible annotation scheme for capturing logic pattern of attacks in arguments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2446–2459, 2022.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.

LENZ Mirko, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann. Towards an argument mining pipeline transforming texts to argument graphs. In *Computational Models of Argument: Proceedings of COMMA*, volume 326, page 263, 2020.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, 2016.

Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19:1–22, 2011.

Marie-Francine Moens. Argumentation mining: How can a machine acquire common sense and world knowledge? *Argument & Computation*, 9(1):1–14, 2018.

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, 2007.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, 2020.

Ashley Muddiman and Natalie Jomini Stroud. News values, cognitive biases, and partisan incivility in comment sections. *Journal of communication*, 67(4):586–609, 2017.

Shoichi Naito, Shintaro Sawada, Chihiro Nakagawa, Naoya Inoue, Kenshi Yamaguchi, Iori Shimizu, Farjana Sultana Mim, Keshav Singh, and Kentaro Inui. Typic: A corpus of template-based diagnostic comments on argumentation. In *13th International Conference on Language Resources and Evaluation Conference, LREC 2022*, pages 5916–5928. European Language Resources Association (ELRA), 2022.

Thomas E Nichols and Andrew P Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1):1–25, 2002.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1): 11–33, 2015.

Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument mining with structured svms and rnns. *arXiv preprint arXiv:1704.06869*, 2017.

Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholom, Sweden, 1999.

E Michael Nussbaum. Argumentation, dialogue theory, and probability modeling: Alternative frameworks for argumentation research in education. *Educational Psychologist*, 46(2):84–106, 2011.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Daniel J O'Keefe. Conviction, persuasion, and argumentation: Untangling the ends and means of influence. *Argumentation*, 26:19–32, 2012.

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, 2009.

Joonsuk Park. *Mining and evaluating argumentative structures in user comments in eRulemaking*. Cornell University, 2016.

Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.

Andreas Peldszus and Manfred Stede. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815, 2015a.

Andreas Peldszus and Manfred Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948, 2015b.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Chaim Perelman. *The new rhetoric*. Springer, 1971.

Isaac Persing and Vincent Ng. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, 2016.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

John L Pollock. Defeasible reasoning. *Cognitive science*, 11(4):481–518, 1987.

John L Pollock. How to reason defeasibly. *Artificial Intelligence*, 57(1):1–42, 1992.

Henry Prakken. On the nature of argument schemes. *Dialectics, dialogue and argumentation. An examination of Douglas Walton's theories of reasoning and argument*, pages 167–185, 2010.

Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. *Handbook of philosophical logic*, pages 219–318, 2002.

Jan Wira Gotama Putra, Simone Teufel, and Takenobu Tokunaga. Annotating argumentative structure in english-as-a-foreign-language learner essays. *Natural Language Engineering*, 28(6):797–823, 2022.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21 (1):5485–5551, 2020.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, 2019.

Ruty Rinott, Lena Dankin, Carlos Alzate, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence-an automatic method for context dependent evidence detection. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 440–450, 2015.

Gil Rocha, Christian Stab, Henrique Lopes Cardoso, and Iryna Gurevych. Cross-lingual argumentative relation identification: from english to portuguese. In *Proceedings of the 5th Workshop on Argument Mining, 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, 2018.

João António Rodrigues and António Branco. Transferring confluent knowledge to argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6859–6874, 2022.

Sebastian Ruder. *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway, 2019.

Carlos Ruiz, David Domingo, Josep Lluís Micó, Javier Díaz-Noci, Koldo Meso, and Pere Masip. Public sphere 2.0? the democratic qualities of citizen debates in online newspapers. *The International journal of press/politics*, 16(4):463–487, 2011.

Ramon Ruiz-Dolz, Jose Alemany, Stella M Heras Barberá, and Ana García-Fornes. Transformer-based models for automatic identification of argument relations: a cross-domain evaluation. *IEEE Intelligent Systems*, 36(6):62–70, 2021.

Kenji Sagae. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 81–84, 2009.

Patrick Saint Dizier. Argument mining: the bottleneck of knowledge and language resources. In *10th International conference on language resources and evaluation (LREC 2016)*, pages pp–983, 2016.

Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 56–66, 2015.

Robin Schaefer and Manfred Stede. Annotation and detection of arguments in tweets. In *Proceedings of the 7th Workshop on Argument Mining*, pages 53–58, 2020.

Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 35–41, 2018.

Push Singh et al. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, volume 3, 2002.

Anders Søgaard. Semi-supervised learning and domain adaptation in natural language processing. *Synthesis Lectures on Human Language Technologies*, 6(2):1–103, 2013.

Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, 2016.

Alexandru Spatariu, Kendall Hartley, Gregory Schraw, Lisa D Bendixen, and Linda F Quinn. The influence of the discussion leader procedure on the quality of arguments in online discussions. *Journal of Educational Computing Research*, 37(1):83–103, 2007.

Robyn Speer, Catherine Havasi, et al. Representing general relational knowledge in conceptnet 5. In *LREC*, volume 2012, pages 3679–86, 2012.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 116–125, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, 2017.

Christopher W Tindale. *Fallacies and argument appraisal*. Cambridge University Press, 2007.

Kai Ming Ting. Inducing cost-sensitive trees via instance weighting. In *Principles of Data Mining and Knowledge Discovery: Second European Symposium, PKDD'98 Nantes, France, September 23–26, 1998 Proceedings 2*, pages 139–147. Springer, 1998.

Stephen E Toulmin. The uses of argument. 1958.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394, 2010.

Frans H Van Eemeren and Rob Grootendorst. Speech acts in argumentative discussions: A theoretical model for the analysis of discussions directed towards solving conflicts of opinion. *Studies of argumentation in pragmatics and discourse analysis*, 1, 1984.

Frans H Van Eemeren and Rob Grootendorst. *A systematic theory of argumentation: The pragma-dialectical approach*. Cambridge University Press, 2004.

Frans H Van Eemeren and A Francisca Sn Henkemans. *Argumentation: Analysis and evaluation*. Taylor & Francis, 2016.

Frans H Van Eemeren, Rob Grootendorst, Ralph H Johnson, Christian Plantin, and Charles A Willard. *Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments*. Routledge, 2013.

Frans H Van Eemeren, Rob Grootendorst, and Tjark Kruiger. *Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies*, volume 7. Walter de Gruyter GmbH & Co KG, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Bart Verheij and David Hitchcock. *Arguing on the Toulmin Model: New Essays in Argument Analysis and Evaluation*. Springer, 2006.

Jacky Visser, Barbara Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. Argumentation in the 2016 us presidential elections: annotated corpora of television debates and social media reaction. *Language Resources and Evaluation*, 54(1): 123–154, 2020.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. A review corpus for argumentation analysis. In *Computational Linguistics and Intelligent Text Processing: 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II 15*, pages 115–127. Springer, 2014.

Henning Wachsmuth, Manfred Stede, Roxanne El Baff, Khalid Al Khatib, Maria Skeppstedt, and Benno Stein. Argumentation synthesis following rhetorical strategies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3753–3765, 2018.

Jean Wagemans. Constructing a periodic table of arguments. In *Argumentation, objectivity, and bias: Proceedings of the 11th international conference of the Ontario Society for the Study of Argumentation (OSSA), Windsor, ON: OSSA*, pages 1–12, 2016.

Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *LREC*, volume 12, pages 812–817. Istanbul, Turkey, 2012.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

Douglas N Walton. *Argument structure: A pragmatic theory*. University of Toronto Press Toronto, 1996.

Douglas N Walton. *The new dialectic: Conversational contexts of argument*. University of Toronto Press, 1998.

Wei Wang and Thomas E Daniels. Building evidence graphs for network forensics analysis. In *21st Annual Computer Security Applications Conference (ACSAC'05)*, pages 11–pp. IEEE, 2005.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

Yizhong Wang, Sujian Li, and Houfeng Wang. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. Building a corpus of legal argumentation in japanese judgement documents: towards structure-based summarisation. *Artificial Intelligence and Law*, 27:141–170, 2019a.

Hiroaki Yamada, Simone Teufel, and Takenobu Tokunaga. Neural network based rhetorical status classification for japanese judgment documents. In *Legal Knowledge and Information Systems*, pages 133–142. IOS Press, 2019b.

An Yang and Sujian Li. Scidtb: Discourse dependency treebank for scientific abstracts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 444–449, 2018.

David Zarefsky. *The Practice of Argumentation: Effective Reasoning in Communication*. Cambridge University Press, 2019. doi: 10.1017/9781139540926.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. Amr parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, 2019.

Appendix A

List of topics and candidate questions on Kialo, and selected questions on Quora

Under each topic, the first five questions are candidate questions on Kialo, and the last question (in bold) is the selected question on Quora.

1. Politics
    - Should the primary focus of prisons be rehabilitation or punishment
    - Should the US adopt stricter gun controls
    - Is Putin's attack on Ukraine unjustifiable and wrong
    - Should there be a universal basic income (UBI)
    - there should be no limit to freedom of speech
    - **Do you support gun control**
2. Ethics
    - All humans should be vegan
    - The existence of god
    - Is morality objective
    - Is BDSM abusive
    - Should teachers be allowed to wear religious symbols at school
    - **Should teachers be allowed to wear religious symbols at school**
3. Society

- Should women only spaces be open to anyone identifying as a woman
- Should humans act to fight climate change
- Would the world be a better place without humans
- Was Donald Trump a good president
- Should inheritance be minimized to create an equal outset for everyone
- **Should women only spaces be open to anyone who identifies as a female**

4. Law

- Pro life vs. pro choice: should abortion be legal
- Should private cars be forbidden in large cities
- Should jury trials be abolished
- Should all drugs be legalised
- Should cosmetic surgery be banned
- **Should abortion be legal**

5. USA

- Is it time to free the nipple toplessness and gender equality in the US
- Is the USA a good country to live in
- Is Joe Biden better than Donald Trump
- Should racial profiling be banned
- Should the US remove Confederate memorials flags and monuments from public spaces
- **Do you think all Confederate monuments statues should be removed from public areas in the South**

6. Philosophy

- Free will or determinism do we have free will
- Is political correctness detrimental to society
- should all religions be banned on a global scale
- Which one is more accurate rationalism or empiricism
- Should humans procreate
- **Is political correctness good or bad for society**

7. Government

    - Is preferential voting the most effective system for ensuring fairer election outcomes
    - Can man-made climate change be reversed
    - Who should provide healthcare the government or the market
    - Should the government provide funding for arts programs
    - Is GDPR beneficial
    - **Would a completely free market healthcare system with zero government involvement and no subsidies for anyone work better in the long run**

8. Technology

    - Do we need nuclear power for sustainable energy production
    - Can creativity be enhanced by technology
    - Can Kleros (PNK) provide an effective platform for arbitration and dispute resolution
    - Should mobile phones be used in the classroom
    - Should humans be allowed to explore DIY gene editing
    - **Should nuclear energy be banned**

9. Economics

    - Governments should structure fiscal policy in line with modern monetary theory (MMT) to better manage their economy
    - Is Elon Musk's takeover of Twitter good for the public
    - Does trickle down economics do more harm than good
    - A socialist economy would work better than a capitalist economy
    - Is globalisation good or bad
    - **Which do you prefer: a socialist or capitalist economy**

10. Religion

    - Has religion been a good thing for humanity
    - Does the Bible support the conclusion that homosexuality is a sin against god
    - Should creationism be taught in schools

- Should religion be involved in politics
- Do we need religion for morality
- **Are religions good for us**

11. Science

    - Do aliens exist
    - If scientifically possible should humans become immortal
    - Is sexual orientation a choice
    - Is cloning animals ethical
    - Is there a need for testing on lab animals in research
    - **Should animal testing be banned**

12. Health

    - Is sex reassignment surgery the best option for transgender people
    - The ethics of eating animals: is eating meat wrong
    - Has social media been good for humanity
    - Is vegetarianism a healthy or unhealthy diet to have
    - Is organic farming better than conventional farming
    - **Which is a better way to be healthy: being vegetarian or non-vegetarian**

13. Education

    - Should school uniforms be banned
    - Should single sex schools be banned
    - Should children learn about gender identity and sexual orientation in school
    - Should higher education be publicly funded
    - Should school be mandatory
    - **Is it necessary for children to wear school uniforms**

14. Children

    - Should a license be required in order to have a child
    - Is opposite sex parenting preferable to same-sex parenting
    - The primary purpose of marriage is procreating and raising children

- Should men pay child support when abortion is legal
- Should the age of consent be lowered to fourteen
- **Should same-sex couples be able to adopt children**

15. Gender

    - Does feminism empower both women and men
    - Gender neutral bathrooms: should they be standard
    - Are trans women real women
    - Should all transgender athletes have to compete in men's divisions
    - Does feminism strive for equality
    - **Is feminism corrupting the society**

16. Environment

    - Are the rich or the poor more responsible for environmental damages
    - We should adapt to climate change rather than advert it
    - Disposable plastic items should be banned
    - Is Greta Thunberg's impact on society positive
    - Should drinking water be fluoridated
    - **Is Greta Thunberg bringing a positive change**

17. Justice

    - Do gun control laws reduce crime
    - Should animal testing be banned
    - Juveniles who commit violent crimes should be treated as adults in the criminal justice system
    - Is OKC police officer Daniel Holtzclaw really guilty of the crimes he was convicted of
    - Should US presidents have the power to issue pardons
    - **Should juvenile criminals be punished like adults**

18. Equality

    - Should society work towards becoming colorblind in regards to race/ethnicity
    - Should hate speech be legally protected

- Should society normalise men wearing dresses/skirts
- Should private education be banned
- Should sex work be legal
- **Should hate speech be protected under the free speech amendment**

19. Democracy

- Democratic governments need safeguards against their own military
- Will liquid democracy be a better mechanism of governance than representative democracy
- Should burning the US flag be illegal
- Compulsory voting: should voting be mandatory
- Is capitalism good
- **Should voting in US elections be mandatory for US citizens**

20. Culture

- Is cultural appropriation wrong
- Is freedom of sexuality inducing harm?
- The fewer languages there are the better the world is
- Should bullfighting be banned
- Should public nudity be legal
- **Do you support same-sex marriage**