Capturing Discourse Structure and Reasoning in Argumentation

Farjana Sultana Mim

Department of System Information Sciences
Graduate School of Information Sciences
Tohoku University
Sendai, Japan

This dissertation is submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

September 2022

Supervisor: **Professor Kentaro Inui**
Natural Language Processing Laboratory,
Department of System Information Sciences,
Graduate School of Information Sciences,
Tohoku University

Examiners: **Professor Akinori Ito**
Department of Communications Engieering,
Graduate School of Engineering,
Tohoku University

Professor Yoshifumi Kitamura
Deputy Director,
Research Institute of Electrical Communication,
Tohoku University

Professor Jun Suzuki
Center for Data-driven Science and Artificial Intelligence (CDS),
Tohoku University

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Kentaro Inui for his valuable guidance and constant encouragement throughout my PhD journey. I cannot thank him enough for all of his care and help that enabled me make the most of Japan life and go through the obstacles during the tough time of pandemic. I have learned so much from him, and not only about how to carry out the research and think critically, but also about the amazing perspectives we can have in life. He is a warmhearted and wise person full of life and energy, and I feel so honored and privileged to be able to do research under his supervision.

I would also like to thank Dr. Naoya Inoue, former assistant professor of Tohoku University and current associate professor of Japan Advanced Institute of Science and Technology (JAIST), who has mentored me from the very beginning. He is one of the most kindhearted, warm and friendly persons I have ever met. He always replied to all of my silly questions with a big smile on his face which played a crucial role in my knowledge development and I eventually learned how to question better and where to look for answers. I am so grateful to have him as my mentor.

I am thankful to Prof. Jun Suzuki for his opinions and advice. I would also like to thank Dr. Hiroki Ouchi and Dr. Paul Reisert for their mentoring, guidance and support in my research.

I am very grateful to all the members of the laboratory. Because of them, I always found the laboratory a friendly place even though I struggled with my Japanese language skills. I thank Ms. Haruka Aizawa, Mrs. Mayumi Sugawara, Mrs. Yoriko Isobe and Mrs. Naoko Odamaki. Without their help and support on official affairs, it would be quite difficult for me to focus on the research. I specially want to express my gratitude to Mrs. Naoko Odamaki for being an amazing friend and giving me comfort and taking care of me during my hard times.

I want to thank all members of the Data Sciences Program (DSP) for their helpful and friendly support, especially I am thankful to Mrs. Ikumi Koyama. I acknowledge the financial support received by the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

I lost my father before my pre-defense. If he were alive, he would be very proud of me and this thesis. He showered me with tremendous amount of love which always gave me the courage to go on no matter how difficult the situation is. I could not tell him how much I love him when he passed away but I hope he knew it in his heart.

I would like to thank my beloved husband Dr. Md. Shafiul Alam for his continuous love and encouragement. Without his support, I would not be here where I am now.

Last but not least, I want to express my thanks, gratitude and love to my mother and other family members as well as to my friends for their understanding, support, love and encouragement.

To the Memory of My Father

Abstract

Assessing the quality of argumentation and capturing the reasoning patterns in it are two very important tasks in computational argumentation and are potential ways to provide feedback to students so that they can improve their argumentative writing.

Existing approaches for automatically assessing argumentative texts typically rely on parsers to capture argumentative discourse. However, the performance of parsers is not always adequate, especially when they are used on noisy texts, such as student essays. To overcome this problem, in this thesis, we establish an unsupervised pre-training approach to capture argumentative discourse that does not require any parser or annotation. Our proposed unsupervised approach achieves state-of-the-art result on a task of scoring student essays.

Essay scoring systems provide feedback about the quality of argumentative essays but do not indicate the issues why the quality of an essay is good or bad. In order to improve the quality of students' argumentative writing, we need feedback systems that do not only provide a score for an argumentative text, but at the same time allow students to inspect the issues in their text. To build such systems, deeper analysis of argumentation is necessary.

For deeper understanding, capturing writer's reasoning in argumentative texts is crucial. However, less attention has been paid to capturing reasoning patterns in argumentation and there are no studies that capture complex strategic moves in argumentation. We address this gap in this thesis and design a novel annotation scheme that capture logic patterns of strategic moves in argumentation. Our annotation study shows that human annotation for the proposed scheme is feasible and results in the creation of a corpus of debates comprising logic patterns of strategic moves. Using our annotated corpus, we then establish the task of automatic identification of logic patterns and our experimental results show moderate performance, setting a baseline for this task.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Imagination is more important that knowledge. Knowledge is limited. Imagination encircles the world.
>
> — *Albert Einstein*

Argumentation is one of the fundamental forms of communication. People use arguments in their everyday life to form an opinion, establish a belief, make some decisions regarding a certain matter, or to persuade others of a stance towards a belief or action they find favorable to their interest. Argumentation is omnipresent everywhere in our lives within diverse forms such as news, debates, essays, online discussions, political speeches, scientific papers, legal texts. The general purposes of argumentation include achieving persuasion, agreement, justification, or a resolution of dispute and doubt.

Argumentation has been studied since ancient times, dating back to the Aristotle's treatise on rhetoric. Today, it is considered as a distinct research subject and spans across diverse fields such as linguistics, philosophy, law, education, psychology and computer science. Since argumentation involves reasoning process and people in all societies and of all languages argue, it has now become a central study within the fields of artificial intelligence and natural language processing.

Argument mining, the automatic identification and extraction of argumentative structures and reasoning from the natural language texts, is an area of natural language processing (NLP). Argument mining has gained considerable attention in recent years due to its critical role in diverse downstream applications such as writing support systems,

debating systems, fact-checking systems, automated feedback systems for students, intelligent personal assistants, automated decision making etc.

In educational domain, argumentation has a significant importance since it is one of the crucial aspects of writing skill acquisition and can be used to foster students' learning. For example, argumentation can enable students think critically, make them understand the importance of justifying or validating an idea or the multiplicity of positions regarding a belief. However, assessing and analyzing students' argumentation is an extremely time consuming task and requires a lot of human efforts. Because of this reason, automated feedback system, which aims to provide automated feedback to students so that they can improve their argumentation, has become one of the most important downstream applications of argument mining.

In order to build such automated feedback system, there are two major tasks in argument mining that need to be done. One of them is assessing the quality of argumentation so that students can learn how good or bad their arguments are and the other is capturing the reasoning patterns in argumentation which can help provide feedback to students regarding the issues that makes their arguments poor.

For the precise assessment of argumentation quality, incorporating its discourse information is crucial. *Discourse* generally refers to how words, sentences, paragraphs or concepts are logically connected to each other to provide comprehensive meaning. Existing studies use discourse annotations, parsers or pre-trained deep language representation models to incorporate such information (Stab and Gurevych, 2014b; Wachsmuth et al., 2016; Ghosh et al., 2016; Liu et al., 2019a; Nadeem et al., 2019). However, discourse annotations are costly, and parsers generally consider that the text is well-written which is not always true (e.g., student essays). To sum up, using parsers for capturing the discourse has its own limitations (Ji and Smith, 2017), especially when used on poorly written text. Moreover, long-range discourse dependencies are not well captured by the pre-trained language models (Xu et al., 2020) because of the token and sentence level pre-training (not document level). In this thesis, we address this gap and set a goal to assess the quality of argumentation by capturing its long-range discourse dependencies in an unsupervised way that doesn't require any expensive parser or annotation.

Quality assessments of argumentation only specify how good or bad an argumentation is but do not indicate the issues why the quality is good or bad. Capturing the underlying reasoning patterns of arguments is needed for such deeper understanding of

argumentation (Walton et al., 2008; Reisert et al., 2018; Jo et al., 2021a). However, representation of reasoning patterns is relatively under-explored. Specially, argumentation often comprise complex strategic moves, e.g., arguers may agree with a logic of an argument while attacking another logic and no existing studies capture such complex strategic moves in argumentation. In this thesis, we also address this problem and set a goal to capture strategic moves in argumentation.

In short, in this thesis, we focus on the two major tasks that are important to achieve the ultimate goal of providing automated feedback to students regarding the quality and issues in their argumentative texts. One of these tasks is capturing the discourse structure of argumentation to assess its quality and the other is capturing underlying reasoning patterns in argumentation. In this thesis, we discuss some methodologies and research directions regarding these two critical tasks.

1.1 Research Issues

In this thesis, we address the following research issues:

- **How to capture discourse structure in argumentation in an unsupervised way?** Existing work capture discourse structure either by using discourse annotations and parsers or by using pre-trained language models. However, annotating discourse structure is costly, and pre-trained language models do not capture long-range discourse dependencies very well because of its word and sentence level pre-tarining. Therefore, one need to explore unsupervised pre-training approaches that can capture long-range discourse dependencies in argumentation without any parsers or annotations.

- **What are the common reasoning patterns in argumentation?** Argumentation often comprise complex strategic moves (e.g., arguers may agree with a logic of an argument while attacking another logic). However, no existing studies capture the reasoning patterns of such strategic moves in argumentation. Thus, analysis is required to discover the common reasoning patterns present in the argumentation.

- **How to capture the common reasoning patterns in argumentation?** If we know what are the common reasoning patterns in argumentation, then the next critical question is if it is possible to develop an annotation scheme that can capture such reasoning patterns and if human annotation for such scheme is feasible.

- **Is it possible to automatically identify the reasoning patterns in argumentation?** Having texts annotated with the reasoning patterns doesn't unfold if the automatic identification of such reasoning patterns is possible or how difficult the task is. One needs to perform experiments and analyze the results in order to answer these questions.

1.2 Contributions

This thesis makes the following contributions:

- **Establishing an unsupervised approach to capture long-range discourse dependencies in argumentation:** We propose a novel unsupervised pre-training approach to capture long-range discourse dependencies in argumentation that does not require any discourse parsers or annotations. We then use our unsupervised pre-training method for the quality assessment of argumentation. We demonstrate that our method is effective in capturing discourse structure of argumentation by achieving state-of-the-art performance on the assessment task.

- **Designing an annotation scheme to capture the reasoning patterns in argumentation:** We analyze the internal structure of how one argument attacks or agrees with another argument which provide insights into how to represent the strategic moves in argumentation so that human annotation is plausible. Based on these insights, we design a novel annotation scheme, define the annotation guidelines and formulate the task of capturing the logic pattern of attacks in argumentation.

- **Construction of a corpus using the invented annotation scheme:** We conduct an annotation study and create a corpus comprising logic pattern of attacks using our proposed scheme. Our annotation study yields moderate agreement between two annotators indicating the feasibility of the human annotation for the scheme.

- **Baseline model experiments for the automatic identification of reasoning patterns:** We consider the automatic identification of reasoning patterns as a reasoning patterns generation task and use a pre-trained language model for the the generation purpose. The model achieves moderate performance, setting a baseline for this task.

1.3 Thesis Overview

The rest of this thesis is structured as follows:

- **Chapter 2: Background**. In this chapter, we first introduce the basic concepts related to argumentation and discourse and then we provide an overview of the existing argument mining tasks as well as the datasets and annotation schemes used in these tasks.

- **Chapter 3: Capturing Discourse Structure for Assessing the Quality of Argumentation**. In this chapter, we explore unsupervised pre-training approaches that can capture long-range discourse dependencies in argumentation without any parsers or annotations and then we use such unsupervised approaches to assess the quality of argumentation. We verify the effectiveness of our approach by comparing it to a strong baseline and the current state-of-the-art models and analyze how the proposed model performs in low resource setting and when we slightly change the pre-training strategy.

- **Chapter 4: Capturing Logic Patterns in Argumentation**. In this chapter, we perform a preliminary study to discover the common underlying logic patterns in argumentation and then based on the insights of this study, we build an annotation scheme that can capture such common patterns in argumentation. We conduct an annotation study using the invented scheme and calculate the inter-annotator agreement between two human annotators. Finally, we analyze the disagreements between annotators and discuss the weakness of the scheme.

- **Chapter 5: Automatic Identification of Logic Patterns in Argumentation**. We establish the task of automatic identification of logic patterns of attacks in argumentation in this chapter. We conduct baseline model experiments considering the logic pattern identification as a logic pattern generation task. We also analyze the results to understand the weakness of the model and provide insights for future research.

- **Chapter 6: Conclusion**. We summarize our contributions and present our future direction.

Chapter 2

Background

This chapter introduces the basic notions related to argumentation and discourse as well as gives an overview of the argument mining researches, existing annotated corpora, and annotation schemes.

2.1 Argument and Argumentation

> *An argument is an attempt to support a conclusion by giving reasons for it*
>
> — Robert Ennis, *Critical Thinking (1995)*

> *Argument refers to the giving of reasons to support or criticize a claim that is questionable, or open to doubt*
>
> — Douglas Walton, *Fundamentals of Critical Argumentation (2005)*

People argue every now and then whenever their opinions do not match which others or when they need to make some decisions regarding certain matters. Arguments are used to persuade people to adopt a belief or to prevent them from adopting a certain belief. In general, an argument is a justifiable position consisting two components: (i) Conclusion (i.e., statement that expresses the position or belief of the arguer) and (ii) Premise (i.e., statements that provide reason for the conclusion). Example of an argument is given below where the conclusion is supported by its premise.

(1) **Conclusion:** We should abolish death penalty.

Premise: Death penalty deprives the chance of rehabilitation of the criminals.

Generally, argumentation refers to the usage of arguments in a situation of disagreement or doubt (e.g., debates, student essays, online discussions) (Lewiński and Mohammed, 2016). Arguments in an argumentation are connected between themselves either with a *support* or *attack* relation. For example, the following argument *attacks* the Example Argument 1.

(2) **Conclusion:** Death penalty should not be abolished.

Premise: Rehabilitation fails in comparison with the death penalty at reducing or eliminating repeat offending.

Argumentation is a communicative or interactional act where people justify or refute an opinion in order to obtain the approval of an audience (van Eemeren et al., 2014). In educational context, argumentative interactions are often considered as a learning tool since through arguments, students can learn how to reason, verify, and the difference between fact and opinion which improves their critical thinking skills (Nathalie, 2015). The ability to form convincing arguments plays a crucial role in negotiation and decision making (Walton et al., 2008). However, people are often unable to develop good argumentation skill due to the lack of constructive feedback or proper guidance (Hattie and Timperley, 2007).

Although providing feedback is very important to improve learners' argumentation skill, providing manual feedback to each learner on their argumentation is not ideal since it is a time consuming task which requires lots of human efforts, and oftentimes difficult given the rise of massive online discussions. Hence, there has been an increasing importance of argumentative writing support systems since it can train individual learner to improve their argumentation skill by analyzing their arguments and providing feedback.

2.2 Discourse

One of the important aspects of argumentation is *discourse*. Discourse is a broad term which generally refers to the coherent written or spoken language longer than a single sentence (Van Dijk, 1997). Discourse has been studied for decades in different areas, specially in linguistics, and several theories of discourse structure have been formalized

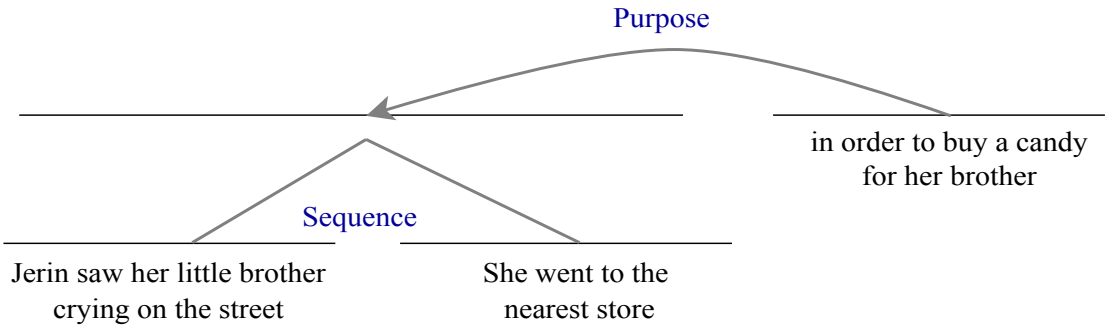

Figure 2.1: A text represented by the Rhetorical Structure Theory (RST)

(Cohen, 1987; Mann and Thompson, 1988; Marcu, 2000). One of renowned and widely used discourse theories is *Rhetorical Structure Theory (RST)* (Mann and Thompson, 1988) where the authors assume that text can be partitioned into some non-overlapping elementary discourse units (EDUs). The authors created 23 rhetorical relations such as elaboration, contrast, background, evidence to represent the connections between these units. In RST, there are two types of EDUs, *nucleus* and *satellite* where the nucleus presents the writer's main purpose, and the satellite contributes to the nucleus and is only interpretable with nucleus. Using RST, the discourse structure of a text can be represented as a hierarchy of EDUs linked with the rhetorical relations. Figure 2.1 shows how the discourse structure of a text is represented by RST.

Discourse in argumentation or "argumentative discourse" involves usage of arguments to persuade an audience or reach a consensus (Van Dijk, 1997). Argumentative discourse is dialectical in nature where two opposite stances contest with each other, either explicitly or implicitly. For example, a news on "abolishing death penalty" not only provides supporting arguments for its stance but also addresses opposite stance by providing counterarguments that explains the disadvantages of abolishing such penalty.

2.3 Argument Mining

Argument mining is an area of natural language processing (NLP) which aims to automatically extract argumentation structures and reasoning from the unstructured natural language texts. Argument mining has gained vast popularity in recent years because of its importance in many NLP applications such as writing support systems, automated feedback systems, decision support systems.

Argument mining encompasses a wide range of tasks such as argumentative units (e.g.,

claim, premise) identification (Stab and Gurevych, 2014a; Levy et al., 2014; Rinott et al., 2015), argumentative relations (e.g., support, attack, neutral) classification (Peldszus and Stede, 2015; Cocarascu and Toni, 2017; Niculae et al., 2017; Stab and Gurevych, 2014a; Deguchi and Yamaguchi, 2019; Kobbe et al., 2019; Jo et al., 2021a), qualitative assessment of arguments (Persing et al., 2010; Persing and Ng, 2013, 2014, 2015, 2016b; Rahimi et al., 2015; Wachsmuth et al., 2016; Habernal and Gurevych, 2016; Wachsmuth et al., 2017; Mim et al., 2019b,a, 2021), retrieval or generation of counterarguments (Hua and Wang, 2018; Wachsmuth et al., 2018; Hua et al., 2019; Reisert et al., 2019; Alshomary et al., 2021; Jo et al., 2021b), and capturing or explicating the encapsulated knowledge in arguments (e.g., causal knowledge, commonsense knowledge, factual knowledge) which are often implicit (Habernal et al., 2018; Hulpus et al., 2019; Becker et al., 2019, 2020; Al-Khatib et al., 2020; Becker et al., 2021b,a; Singh et al., 2021; Saha et al., 2021).

Argument mining is also tied to stance and sentiment analysis tasks because every argument carries a *for* or *against* stance towards a topic which people often express with positive or negative sentiment (Wachsmuth et al., 2014; Kobbe et al., 2020). For example, in the argument *"We should abandon marriage. A piece of paper doesn't keep people together"*, there is an *against* stance or *negative* sentiment towards the topic "marriage".

Existing argument mining approaches consider widespread genres of argumentation such as student essays (Persing et al., 2010; Persing and Ng, 2013, 2014, 2016b; Wachsmuth et al., 2016, 2017; Mim et al., 2019b,a, 2021), debates (Habernal and Gurevych, 2015; Al Khatib et al., 2016a; Mim et al., 2022), scientific articles (Green et al., 2014; Lauscher et al., 2018; Fergadis et al., 2021), news editorials (Al Khatib et al., 2016b; El Baff et al., 2018, 2020; Alhindi et al., 2020), product reviews (Wachsmuth et al., 2015; Liu et al., 2017, 2021).

2.4 Annotated corpora and Annotation Schemes

Availability of large-scale annotated datasets is pivotal for designing, training, and testing argument mining algorithms. Therefore, creation of annotated corpora of argumentation structures and reasoning has been one of the main research focuses in argument mining.

A corpus generally comprises three elements: (i) annotated data that represent gold standard and whose annotation has been checked and validated, (ii) annotation guide-

Argument from negative consequences

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

Example

If homework is abolished, many students will not study at all.
Therefore, homework should not be abolished

Figure 2.2: One of the Walton's argumentation schemes and an example argument where this scheme can be applied.

lines that explain the details of how the data has been annotated, and (iii) the unlabeled raw corpus that can be used to test an argument mining algorithm. Generally, the reliability of a corpus is affirmed by performing the annotation with multiple (at least two) annotators and presenting the inter-annotator agreement that measures the degree of agreement in annotation decisions among the annotators. For different argument mining tasks, various corpora have been created such as Araucaria (Reed et al., 2008), AIFdb (Lawrence et al., 2012), Argument Annotated Essays Corpus (AAEC) (Stab and Gurevych, 2014a), European Court of Human Rights (ECHR) corpus (Mochales and Moens, 2008), Debatepedia corpus (Cabrio and Villata, 2012), web discourse corpus (Habernal and Gurevych, 2017), Internet Argument Corpus (IAC) (Walker et al., 2012), TYPIC corpus (Naito et al., 2022).

Creating annotated corpora requires formal structural representations which we refer to annotation schemes. In argument mining, generally most of the datasets are constructed with different annotation schemes in order to serve different purposes, domains or tasks. For example, Stab and Gurevych (2014a) built Argument Annotated Essays Corpus (AAEC) for the education domain where they used a scheme that models three argument components (i.e., MajorClaim, Claim, Premise) and two relations (i.e., support and attack) between these components in persuasive essays.

There are many structured argumentation theories or schemes and a few of them have been exploited for the construction of corpora in argument mining (Freeman, 2001; Walton et al., 2008). One renowned theory that is widely adopted in argument mining is Walton's argumentation schemes (Walton et al., 2008) which specify the common reasoning patterns of how one argument supports another argument and comprise around 60 schemes. Fig 2.2 shows one of these schemes and an example argument which

can be represented by this scheme. Exploiting such existing schemes or creating new schemes for a specific task, domain or purpose has become one of the crucial element of argument mining research.

Chapter 3

Capturing Discourse Structure for Assessing the Quality of Argumentation

3.1 Introduction

Argumentation has a significant importance in education since it is one of the crucial aspects of writing skill acquisition and can be used to foster learning by enabling students think critically, making them understand the importance of justifying or validating an idea or the multiplicity of positions regarding a belief (Mirza and Perret-Clermont, 2009). A typical example of argumentation in educational domain is student's essay. Assessing the quality of student essays is quite important since building well-defined arguments is essential for any type of persuasion or decision making and by such assessments students gets feedback about the quality of their arguments which can help them improve their argumentation. The assessment of students essays has been extensively studied in the context of automated essay scoring.

Automated Essay Scoring (AES), the task of both grading and evaluating written essays using machine learning techniques, is an important educational application of natural language processing (NLP). Since manual grading of student essays is extremely time consuming and requires lots of human efforts, AES systems are widely adopted for many large-scale writing assessments such as Graduate Record Examination (GRE) (Attali and Burstein, 2006). Recent research in AES not only focuses on scoring overall

quality (i.e., holistic scoring) of essays but also scoring a particular dimension of essay quality (e.g., Organization, Argument Strength, Style), in order to provide constructive feedback to learners (Persing et al., 2010; Persing and Ng, 2013, 2014, 2015, 2016b; Wachsmuth et al., 2016; Mathias and Bhattacharyya, 2018; Mim et al., 2019a).

In general, an essay is a discourse where sentences and paragraphs are logically connected to each other to provide comprehensive meaning. Conventionally, two types of connections have been discussed in the literature: *coherence* and *cohesion* (Halliday, 1994). Coherence refers to the semantic relatedness among sentences and logical order of concepts and meanings in a text. For example, *"I saw Jill on the street. She was going home."* is coherent, whereas *"I saw Jill on the street. She has two sisters."* is incoherent. Two types of coherence are well known in the literature: *local coherence* and *global coherence*. Local coherence generally refers to how well-connected adjacent sentences are (Barzilay and Lapata, 2008) whereas global coherence represents the discourse relation among remote sentences to present the main idea of the text (Unger, 2006; Zhang, 2011). Cohesion refers to how well sentences and paragraphs in a text are linked by means of linguistic devices. Examples of these linguistic devices include conjunctions such as discourse indicators (DIs) (e.g., *"because" and "for example"*), coreference (*e.g., "he" and "they"*), substitution, ellipsis, etc.

For the precise assessment of overall essay quality or some dimensions of an essay, it is crucial to encode such discourse structure (i.e., coherence and cohesion) into an essay representation. One such dimension of an essay is *Organization*, which refers to how good an essay structure is (Persing et al., 2010). Essays with high Organization score have a structure where writers introduce a topic first, state their position regarding the topic, support their position by providing reasons, and finally conclude by repeating their position.

An example of the relation between coherence, cohesion, and an essay's Organization is shown in Figure 3.1. The high-scored essay (i.e., Essay (a) with an Organization score of 4) first states its position regarding the prompt and then provides several reasons to strengthen the claim. The essay is considered coherent because it follows a logical order that makes the writer's position and arguments very clear. However, Essay (b) is not clear on its position and what it is arguing about. The third paragraph gives a vibe that the writer is supporting the prompt, but then the fourth paragraph provides a clear statement that the writer is opposing the prompt. Therefore, it can be considered incoherent since it lacks logical sequencing. Furthermore, Essay (a) has cohesive markers (e.g., "in connection with", "as a conclusion") at the beginning of paragraphs

Prompt: Some people say that in our modern world , dominated by science, technology and industrialization, there is no longer a place for dreaming and imagination. What is your opinion?

Essay (a)
Coherent (Organization Score = 4.0)
There is no doubt in the fact that we live under the full reign of science, technology and industrialization. Our lives are dominated by them in every aspect……. In other words, what I am trying to say more figuratively is that in our world of science, technology and industrialization there is no really place for dreaming and imagination. One of the reasons for the disappearing of the dreams and the imagination from our life is one that I really regret to mention, that is the lack of time…… In connection with what I said above I would like to share my own experience. I am a student at Sofia University. I live under a constant stress because I have to study for difficult exams all the time as well as attending lectures and seminars every day……. As a conclusion I would point out the sad truth - our world has progressed to such an extent that we cannot do without science and technology and industrialisation……

Essay (b)
Incoherent (Organization Score = 2.5)
The world we are living in is without any doubt a modern and civilized one..... Perhaps we - the people who live nowadays, are happier than our ancestors, but perhaps we are not. The strange thing is that we judge and analyse their world without knowing it…… On the other hand we do need all these new technical products. We can no longer imagine our lives without a TV set or without a telephone……. In my opinion, technology cannot change us so much and to make us forget what is to dream and imagine. There is always place for dreaming and imagination in our modern world…… This is just a small relief but sometimes it helps you to feel better…….. Imagination and dreaming will always have place in our modern world…….

Essay (c)
Incohesive (Organization Score = 2.5)
Long, freesing winter nights in the Middle Ages somewhere in Europe passed with plucking of feathers ….. Nowadays, we simply do not have the time to sit around and believe every single word our story- teller tells us……. O.K. we have been taught that witches do not exist (anymore?). Then why do we shiver….. Technology has taught us to take up another pace of living but it does not mean the end of imagination; it does not kill our dreams…… What about the seals, the whales, the seagalls? If science really were in such a key…… Television or movies may put limitations on imagination but Virtual Reality…… There is a place for dreaming and imagination just because it is an integral part of human nature, no matter to what extent science or technology…….

Figure 3.1: Example of coherent/cohesive and incoherent/incohesive essays with their respective Organization score. The essays have been shortened for the example, indicated by ellipses.

which helps the reader understand the flow of ideas throughout the essay. Thus, it is considered as a cohesive essay. However, Essay (c) should have some cohesive markers at the beginning of fifth paragraph (e.g., "moreover", "besides") and sixth paragraph (e.g., "therefore", "hence") to connect the ideas between paragraphs, but it doesn't have such cohesive markers. In addition, there is no cohesive marker at the beginning of the last paragraph (e.g., "in conclusion") to indicate that the author is summing up their opinions which makes the last paragraph slightly disconnected from former paragraphs. Due to the absence of these cohesive markers, it is difficult to understand the arguments of the essay and connections between them. Therefore, Essay (c) is considered as an incohesive essay.

Although discourse is one of the most important aspects of documents, less attention has been given to capturing discourse structure in an unsupervised manner for document representation. Most of the works that encapsulate discourse structure into document representation are dependent on argument or Rhetorical Structure Theory (RST) based parser and annotations (Stab and Gurevych, 2014a,b; Mann and Thompson, 1988). However, such annotations are costly, and parsers generally considers that the text is well-written which is not always true, especially in case of student essays that comprise different types of flaws (e.g., grammatical, spelling, discourse etc.). To sum up, using parsers for document representation has its own limitations (Ji and Smith, 2017), especially when used on poorly written text, and it has not yet been explored how long-range discourse dependencies can be included in text embeddings in an unsupervised way without any expensive parser or annotation.

Recent advances in language model (LM) pre-training has inspired researchers to use contextualized language representations for different document-level downstream tasks of NLP, including essay scoring. Several document-level tasks such as document classification, summarization (Adhikari et al., 2019; Zhang et al., 2019; Xu et al., 2020) as well as essay scoring (Steimel and Riordan; Liu et al., 2019a; Nadeem et al., 2019) achieved state-of-the-art performance by leveraging pre-trained language models. Note that many of these tasks obtained only the sentence or text block representations from pre-trained language models instead of a whole document representation. Then they subsequently joined them using some complex architecture, because Transformer-based (Vaswani et al., 2017) pre-trained models (e.g., BERT, RoBERTa (Devlin et al., 2018; Liu et al., 2019b)) are unable to process long document due to token constraints (i.e., they accept up to 512 tokens). Furthermore, due to the self-attention operation of Transformer, processing long documents is very expensive. The recent work of Beltagy et al. (Beltagy et al., 2020) addressed these limitations and introduced Transformer-based

model *Longformer* which is suitable for processing long documents. However, long-range discourse dependencies are not well captured by the pre-trained language models (Xu et al., 2020) because of the token and sentence level pre-training (not document level).

In this thesis, we propose an unsupervised method that enhances a document encoder to capture discourse structure of essay Organization in terms of cohesion and coherence (Section 3.3,3.4). We name our unsupervised technique *Discourse Corruption (DC)* pre-training. We introduce several types of token, sentence, and paragraph level corruption strategies to artificially produce "badly-organized" (incoherent/incohesive) essays. We then pre-train a document encoder which learns to discriminate between original (coherent/cohesive) and corrupted (incoherent/incohesive) essays.

We augment Longformer Beltagy et al. (2020), a strong document encoder pre-trained with Masked Language Modeling (MLM) objective, with our proposed DC pre-training in order to utilize both contextual and discourse information of essays. We assume that the MLM objective will capture the transition of ideas at the local level (e.g., word or sentence level) while our DC pre-training will capture the transition of ideas at global level (e.g., paragraph), and the combination of these two strategies will successfully capture the overall Organization structure of an essay. To the best of our knowledge, we are the first to attach discourse-aware pre-training on top of MLM pre-training. The advantage of our approach is that it is unsupervised and does not require any expensive parser or annotation. Our proposed strategy outperforms two baseline models by a significant margin, and we achieve new state-of-the-art results for essay Organization scoring (Section 3.5,3.6).

3.2 Background

The focus of this study is the unsupervised encapsulation of discourse structure into document representation for essay Organization scoring. In this section, we briefly review the previous works on automated essay scoring, unsupervised document representation learning, and document representation learning using pre-trained language models.

3.2.1 Automated Essay Scoring

AES research generally follows two lines of approaches: feature-engineering approach and deep neural network (DNN) based approach. Traditional AES research utilizes

handcrafted features in a supervised regression or classification setting to predict the score of essays (Larkey, 1998; Attali and Burstein, 2006; Chen and He, 2013; Phandi et al., 2015; Persing et al., 2010; Persing and Ng, 2015; Wachsmuth et al., 2016). Recent studies of AES adopt DNN-based approaches which have shown promising results (Taghipour and Ng, 2016; Alikaniotis et al., 2016; Dong and Zhang, 2016; Dong et al., 2017; Riordan et al., 2017; Farag et al., 2018; Zhang and Litman, 2018; Wang et al., 2018; Cummins and Rei, 2018).

A major shortcoming of many of the AES systems is that they use holistic score of essays (Phandi et al., 2015; Alikaniotis et al., 2016; Taghipour and Ng, 2016; Dong et al., 2017; Wang et al., 2018). Holistic scoring schemes limit the scope of providing constructive feedback to learners since it is not clear how different dimensions of essay quality (e.g., Organization, content, etc.) are summarized into a single score or whether the score refers to only one dimension. In order to address this problem, recent studies have focused on scoring specific dimensions of essay such as Organization, Argument strength (Persing et al., 2010; Persing and Ng, 2015; Wachsmuth et al., 2016), Thesis clarity (Persing and Ng, 2013), Relevance to prompt (Higgins et al., 2004; Persing and Ng, 2014), Stance (Persing and Ng, 2016b), Style Mathias and Bhattacharyya (2018).

Many aspects of essay quality have been exploited for the assessment of essays, and among them, the one that is used often is discourse coherence. Mesgar et al. (Mesgar and Strube, 2018) used an end-to-end local coherence model for the assessment of essays that encodes semantic relations of two adjacent sentences and their pattern of changes throughout the text. Farag et al. (Farag et al., 2018) evaluated the robustness of a neural AES model and showed that neural AES models are not well-suited for capturing adversarial input of grammatically correct but incoherent sequences of sentences. Therefore, they developed a neural local coherence model and jointly trained it with a state-of-the-art AES model to build an adversarially robust AES system. However, these works utilized the particular essay quality "coherence" for the assessment of overall essay quality (holistic scoring). In contrast to these previous works, we capture discourse cohesion and coherence in an unsupervised way to assess a specific dimension of essays i.e., Organization.

Recently, pre-trained deep language representation models have fascinated the NLP community by achieving state-of-the-art results on various downstream tasks of NLP, including essay scoring. One of the widely used masked language models is *BERT* (Devlin et al., 2018), which was trained with MLM objective i.e., predicting the masked tokens in the text. In addition to the MLM objective, BERT is also trained with "next

sentence prediction" task i.e., predicting if the second sentence of a sentence-pair is the actual next sentence or not. Several essay scoring tasks achieved state-of-the-art performance by leveraging BERT. Steimel et al. (Steimel and Riordan) fine-tuned BERT and achieved a state-of-the-art result for content scoring of essays. Liu et al. (Liu et al., 2019a) proposed a two-stage learning framework (TSLF) that integrates both end-to-end neural AES model as well as feature-engineered model and achieved state-of-the-art performance on holistic scoring of essays. In their framework, sentence embeddings are obtained using the pre-trained BERT model. They also incorporated a Grammar Error Correction (GEC) system into their AES model and added adversarial samples to the original dataset which led to a performance gain. Nadeem et al. (Nadeem et al., 2019) used existing discourse-aware models and tasks from literature to pre-train AES models for holistic scoring of essays. They utilized contextualized BERT embeddings for the AES task, hypothesizing that the next sentence prediction task of BERT would capture discourse coherence. They also pre-trained their models with other objectives i.e., natural language inference and discourse marker prediction tasks. Their results showed that contextualized embeddings from BERT performs better than other two pre-training tasks. However, all these studies consider holistic scores where it is unclear which criteria of the essay the score considers. We are the first to show how Transformer-based (Vaswani et al., 2017) architecture with MLM pre-training performs on the assessment of a specific dimension of essays, i.e. essay Organization scoring.

Persing et al. (Persing et al., 2010) annotated essays with Organization scores and established a baseline model for this scoring. They employed heuristic rules utilizing various DIs, words, and phrases to capture the discourse function labels of sentences and paragraphs of an essay. Those function labels were then exploited by various techniques, such as sequence alignment, alignment kernels, and string kernels, for the prediction of Organization score. Later, Wachsmuth et al. (Wachsmuth et al., 2016) achieved state-of-the-art performance on Organization scoring by utilizing argumentative features such as sequence of argumentative discourse units (ADU) (e.g., *(conclusion, premise, conclusion)*, *(None, Thesis)*), frequencies of ADU types, etc. In addition to the argumentative features, they also used sequences of paragraph discourse functions of Persing et al. (Persing et al., 2010) as well as sentiment flows, relation flows, POS n-grams, frequency of tokens in training essays, etc. A simple, supervised regression model is then applied for scoring. However, their work used an argument parser to obtain ADUs, and in this work, we focus on overcoming that parser bottleneck for capturing discourse.

3.2.2 Unsupervised Document Representation Learning

Several unsupervised methods for document representation learning have been introduced in recent years (Le and Mikolov, 2014; Wu et al., 2018; Ionescu and Butnaru, 2019; Gupta et al., 2020). However, less studies have been conducted on unsupervised learning of discourse-aware text representations. One of the studies that illustrated the role of discourse structure for document representation is the study by Ji and Smith (Ji and Smith, 2017) who implemented a discourse structure (defined by RST) (Mann and Thompson, 1988) aware model and showed that their model improves text categorization performance (e.g., sentiment classification of movies and Yelp reviews, and prediction of news article frames). The authors utilized an RST-parser to obtain the discourse dependency tree of a document and then built a recursive neural network on top of it. The issue with their approach is that texts need to be parsed by an RST parser and the parsing performance of RST is not always adequate, especially when used on noisy text. Furthermore, the performance of RST parsing is dependent on the genre of documents (Ji and Smith, 2017).

3.2.3 Pre-trained Language models and Document Representation Learning

Lately, Tansformer-based pre-trained models have achieved significant performance gain in different document-level downstream tasks of NLP. Adhikari et al. (Adhikari et al., 2019) first investigated the effect of pre-trained deep contextualized models on document representation learning. They fine-tuned BERT (Devlin et al., 2018) for several document classification tasks and demonstrated that knowledge can be distilled from BERT to small bidirectional LSTMs which provides competitive results at a low computational expense.

Chang et al. (Chang et al., 2019) proposed methods for pre-training hierarchical document representations that generalize and extend the pre-training method of ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018), respectively. In their approach, LSTM-based architecture consider a document as sequences of text blocks, each block comprising a sequence of tokens, where the text blocks are basically sentences or paragraphs. Zhang et al. (Zhang et al., 2019) presented a strategy to pre-train hierarchical bidirectional transformer encoders for document representation. They randomly masked sentences of documents and predicted those masked sentences with their proposed architecture, a hierarchical fusion of Transformer-based (Vaswani et al., 2017) sentence and document encoders.

A recent work by Beltagy et al. (Beltagy et al., 2020) indicated the attention mechanism and token constraints of Transformer-based (Vaswani et al., 2017) masked language models for long document representation. To mitigate these problems, they introduced a Transformer-based model *Longformer*, which has an attention mechanism that scales linearly with the sequence length, hence being suitable for processing long documents. They pre-trained Longformer with the MLM objective, continuing from the RoBERTa (Liu et al., 2019b) released checkpoint and added extra position embeddings to support long sequence of tokens. The pre-trained Longformer outperformed renowned RoBERTa on various long document tasks.

One recent study by Xu et al. (Xu et al., 2020) utilized a pre-trained language model for capturing the discourse structure of documents. They constructed a discourse-aware neural extractive summarization model *DISCOBERT*. DISCOBERT encodes RST-based discourse unit (a sub-sentence phrase) instead of sentence using BERT. A Graph Convolutional Network is then used to create discourse graphs based on RST trees and coreference mentions. However, this work is dependent on the RST discourse parser, and as mentioned a priori, we would like to overcome that parser bottleneck.

3.3 Model Architecture

3.3.1 Overview

Our model consists of (i) a base document encoder, (ii) an auxiliary encoder, and (iii) a scoring function. The base document encoder produces a vector representation $\mathbf{h}^{\text{base}}$ by capturing a sequence of words in each essay. The auxiliary encoder captures additional essay-related information and produces a vector representation $\mathbf{h}^{\text{aux}}$.

Then, these representations are concatenated into one vector, which is mapped to a feature vector $\mathbf{z}$.

$$\mathbf{z} = \tanh(\mathbf{W} \cdot [\mathbf{h}^{\text{base}}; \mathbf{h}^{\text{aux}}]) \ , \tag{3.1}$$

where $\mathbf{W}$ is a weight matrix. Finally, we use the following scoring function to map $\mathbf{z}$ to a scalar value by the sigmoid function.

$$y = \text{sigmoid}(\mathbf{w} \cdot \mathbf{z} + b) \ ,$$

where $\mathbf{w}$ is a weight vector, b is a bias value, and y is a score in the range of $[0, 1]$. In the following subsections, we describe the details of each encoder.

3.3.2 Base Document Encoder

The base document encoder produces a document representation $\mathbf{h}^{\text{base}}$ in Equation 3.1. For the base document encoder, we use the pre-trained Longformer model (Beltagy et al., 2020).

Longformer is a Transformer-based (Vaswani et al., 2017) model with a modified attention mechanism. Longformer's attention mechanism scales linearly with the input sequence length, making it easy for processing long documents. The attention mechanism of Longformer combines a sliding windowed self-attention for capturing local-context and a task specific global attention. In this attention operation, if the sliding window size is w, then each token will attend to $\frac{1}{2}w$ token on each side, and a token with a global attention will attend to all the tokens across the sequence and all the tokens in the sequence will attend to it as well. Longformer is pre-trained with the MLM objective, continued from the RoBERTa released checkpoint. During pre-training, Longformer's attention mechanism is used as a drop-in replacement for the self-attention mechanism of Transformer-based RoBERTa (Liu et al., 2019b). Specifically, RoBERTa's self-attention is replaced by Longformer's attention. Longformer can process much longer documents by accepting up to 4096 tokens, whereas other pre-trained models like BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019b) only accept up to 512 tokens. Since the Transformer architecture (Vaswani et al., 2017) is well-known and widely used in NLP, we will omit the detailed information. Instead, we present a brief overview of how Longformer is used in our essay scoring model.

Given an input essay of N tokens $t_{1:N} = (t_1, t_2, \cdots, t_N)$, special tokens are inserted at the beginning and the end of the essay, with the input essay of N tokens as $t_{0:N+1} = (\text{[CLS]}, t_1, t_2, \cdots, t_N, \text{[EOS]})$. Next, taking $t_{0:N+1}$ as input, the Longformer model produces a sequence of contextual representations $\mathbf{h}_{0:N+1} = (\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_{N+1})$. Note that, we obtain the representation from the second-to-last layer of Longformer.

$$\mathbf{h}_{0:N+1} = \text{Longformer}(t_{0:N+1}) \ ,$$

Next, we use a mean-over-time layer $\mathbf{h}_{0:N+1}$ as input, which produces a vector averaged over the sequence.

$$h^{\text{mean}} = \frac{1}{N+2} \sum_{n=0}^{N+1} \mathbf{h}_n \ . \tag{3.2}$$

We use this resulting vector as the base document representation, i.e. $\mathbf{h}^{\text{base}} = \mathbf{h}^{\text{mean}}$.

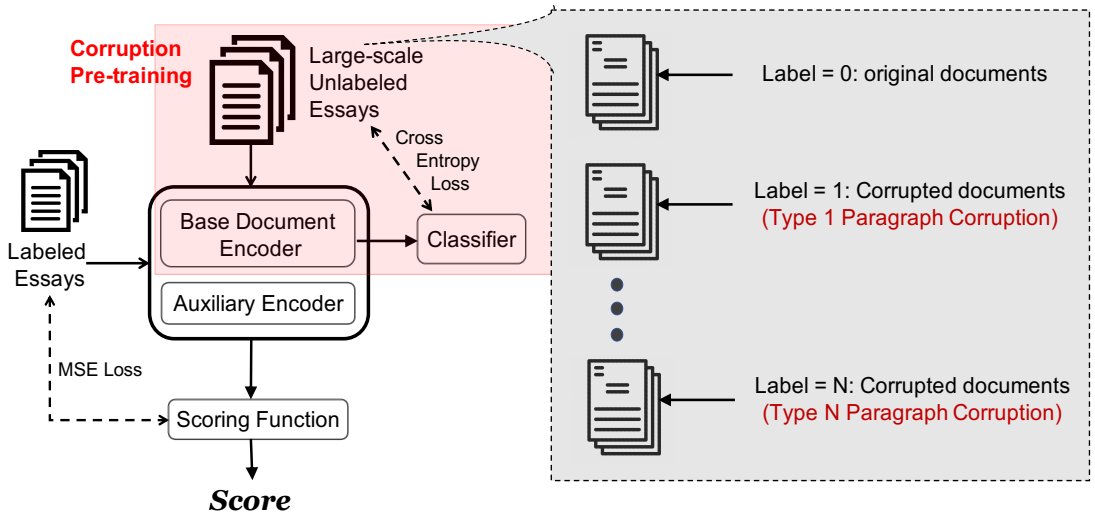

Figure 3.2: Proposed DC pre-training for unsupervised learning of discourse-aware text representation utilizing original and artificially corrupted documents and the use of the discourse-aware pre-trained model for essay scoring.

3.3.3 Auxiliary Encoder (AE)

The auxiliary encoder produces a representation of a sequence of paragraph function labels $\mathbf{h}^{\text{aux}}$ in Equation 3.1.

Each paragraph in an essay plays a different role. For instance, the first paragraph tends to introduce the topic of the essay, and the last paragraph tends to sum up the whole content and make some conclusions. Here, we capture such paragraph functions.

Specifically, we obtain paragraph function labels of essays using Persing et al. (2010) heuristic rules.[1] Persing et al. (2010) specified four paragraph function labels: Introduction (**I**), Body (**B**), Rebuttal (**R**) and Conclusion (**C**). We represent these labels as vectors and incorporate them into our model. Our auxiliary encoder which encodes paragraph function labels consists of two modules, an embedding layer and a Bi-directional Long Short-Term Memory (BiLSTM) Schuster and Paliwal (1997) layer.

We assume that an essay consists of M paragraphs, and the i-th paragraph has already been assigned a function label p_i. Given the sequence of paragraph function labels of an essay $p_{1:M} = (p_1, p_2, ..., p_M)$, the embedding layer ($\text{Emb}^{\text{para}}$) produces a sequence of label embeddings $\mathbf{p}_{1:M} = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_M)$.

$$\boldsymbol{p}_{1:M} = \text{Emb}^{\text{para}}(p_{1:M}),$$

[1] See http://www.hlt.utdallas.edu/~persingq/ICLE/orgDataset.html for further details.

where each embedding $\boldsymbol{p}_i$ is $\mathbb{R}^{d^{\text{para}}}$. Note that each embedding is randomly initialized and learned during training.

Then, taking $\boldsymbol{p}_{1:M}$ as input, the BiLSTM layer produces a sequence of vector representations $\mathbf{h}_{1:M} = (\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M)$.

$$\mathbf{h}_{1:M} = \text{BiLSTM}(\boldsymbol{p}_{1:M}),$$

where $\mathbf{h}_i$ is $\mathbb{R}^{d^{\text{aux}}}$.

We use the last hidden state $\mathbf{h}_M$ as the paragraph function label sequence representation, i.e. $\mathbf{h}^{\text{aux}} = \mathbf{h}_M$.

3.4 Proposed Pre-training Method

3.4.1 Overview

Figure 3.2 summarizes our proposed DC pre-training method. First, we pre-train the base document encoder (Section 3.3.2) to distinguish between original and their artificially corrupted documents. This pre-training is motivated by the following hypotheses: (i) artificially corrupted incoherent/incohesive documents lack logical sequencing, (ii) moderately corrupted documents have better logical sequencing compared to highly corrupted documents and (iii) training a base document encoder to differentiate between original documents and their different types of artificially corrupted documents makes the encoder logical sequence-aware, in other words, discourse-aware. Based on these hypotheses, we train a base document encoder on the original documents and their artificially corrupted documents.

The pre-training is done in two steps. First, the document encoder is pre-trained with large-scale, unlabeled essays from various corpora. Second, the encoder is fine-tuned on the unlabeled essays of the target corpus (essay Organization scoring corpus). We expect that this fine-tuning alleviates the domain mismatch between the large-scale essays and target essays (e.g., essay length). Finally, the pre-trained encoder is then re-trained on the annotations for the essay scoring task in a supervised manner.

Note that our base document encoder (i.e., Longformer) is already pre-trained with the MLM objective, where the aim is to predict randomly masked tokens in a sequence. Previous work (e.g., (Nadeem et al., 2019)) have shown that the next sentence prediction task of BERT i.e., predicting whether the subsequent sentence of a sentence-pair is the

actual next sentence or not, is able to capture discourse coherence. Hence, we also pre-train our model with the binary *next sentence prediction* (N-SentP) task, similar to BERT's. The sentence-pairs are generated from our pre-training corpora and we follow BERT's strategy for the generation of these sentence-pairs. More specifically, when we choose the sentences A and B for each sentence-pair, 50% of the time B is the actual next sentence that follows A and 50% of the time B is a random sentence[2].

We hypothesize that the MLM and N-SentP pre-training would capture local-context while our DC pre-training would capture the long-range dependencies effective for essay Organization scoring.

3.4.2 Corruption Strategies

We would like to produce "badly organized" essays with our corruption techniques so that the encoder can learn the difference between good and bad discourse. Note that essays are not only scored as high or low but throughout a range of scores which means that there is Organization structure which is moderately good/bad. Therefore, in addition to the high corruption techniques, we introduce several types of moderate corruption techniques in order to produce "moderately bad" Organization of essays.

We categorize our corruption strategies into 3 groups: (1) *sentence*, (2) *discourse indicator (DI)* and (3) *paragraph* corruption. Each group has several types of corruption schemes. We discuss the details of each corruption strategy in the following subsections.

3.4.2.1 Sentence Corruption (SC)

This group has 2 different types of corruption. In *Complete Sentence Shuffle* (C-Sent), all the sentences of a document are shuffled. In *Moderate Sentence Shuffle* (M-Sent), only a subset of the sentences of a document are shuffled. Specifically, we randomly select two sentences from a document and shuffle all the sentences between them, including those two sentences as well. Figure 3.3 shows an example of C-Sent and M-Sent.

3.4.2.2 Discourse Indicator Corruption (DIC)

We corrupt DIs since they represent the logical connection between sentences. For example, *"Mary did well although she was ill"* is logically connected, but *"Mary did*

[2]See Appendix **??** for more details

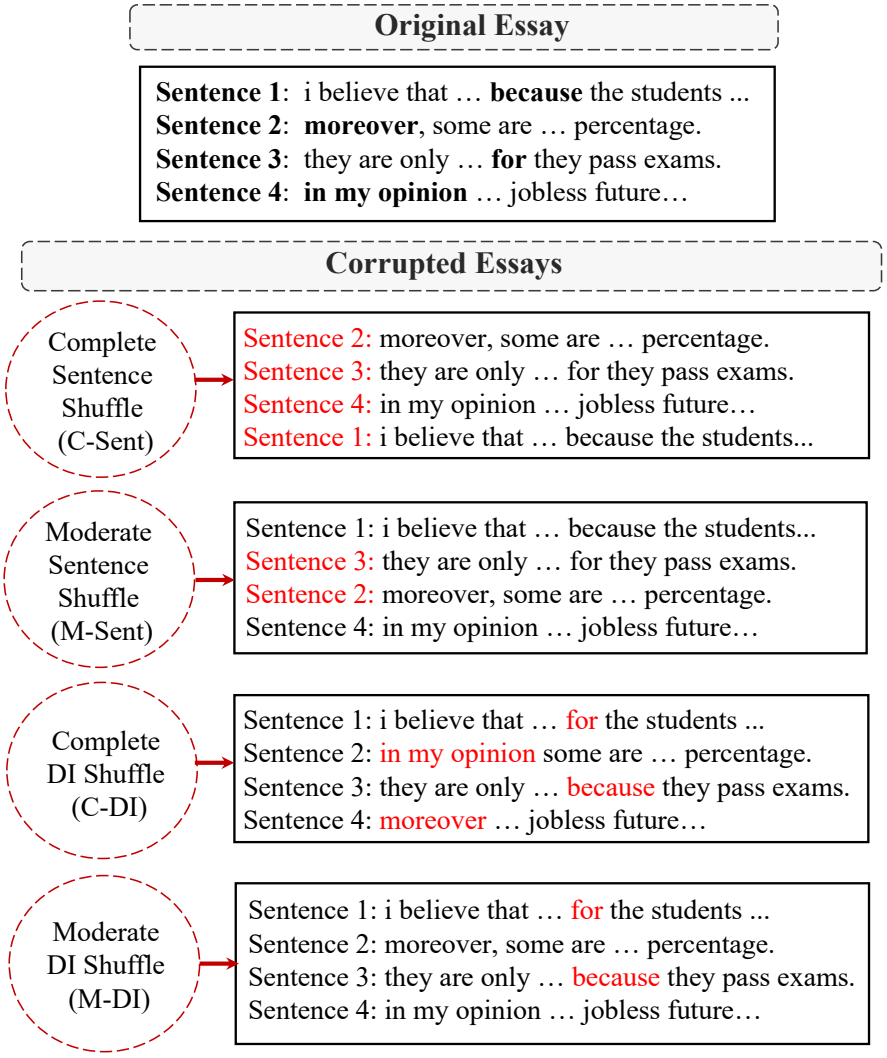

Figure 3.3: Example of different types of Sentence and Discourse Indicator Corruption methods.

well but she was ill." and *"Mary did well. She was ill."* lack logical sequencing because of improper and lack of DI usage, respectively.

We perform two types of DI corruption. In *Complete Discourse Indicator Shuffle* (C-DI), we shuffle all the discourse indicators of a document. In *Moderate Discourse Indicator Shuffle* (M-DI), we first select 50% of unique DIs in a document and randomly shuffle each of their instances in a document. Figure 3.3 shows an example of C-DI and M-DI.

3.4.2.3 Paragraph Corruption (PC)

How ideas are transmitted throughout the paragraphs of an essay determines how good its Organization structure is. For example, coherent essays have paragraph sequences

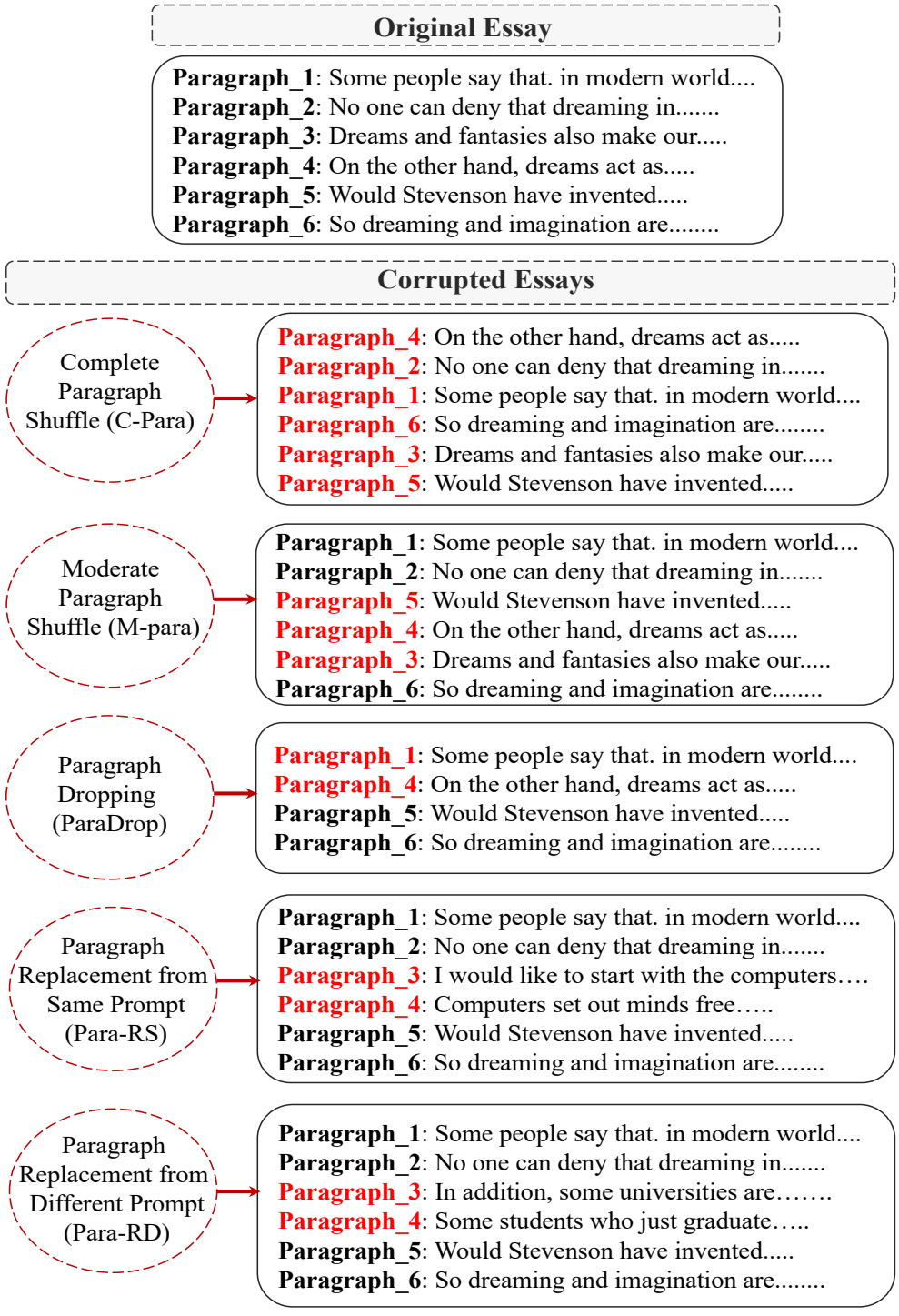

Figure 3.4: Example of different types of Paragraph Corruption

like *Introduction-Body-Conclusion* to provide a logically consistent meaning of the text. Therefore, we conduct five types of paragraph corruption, as illustrated in Figure 3.4.

In *Complete Paragraph Shuffle* (C-Para), we randomly shuffle all the paragraphs of a document. In *Moderate Paragraph Shuffle* (M-Para), we shuffle a subset of the para-

graphs of a document. Precisely, we randomly pick two paragraphs from a document and shuffle all the paragraphs between them including those two paragraphs as well. For example, in the M-Para of Figure 3.4, only paragraph 3,4 and 5 are shuffled.

In *Paragraph Drop* (ParaDrop), we drop 30% of randomly selected paragraphs of a document. Figure 3.4 shows an example of ParaDrop where paragraph 2 and 3 are dropped.

In *Paragraph Replacement from Same Prompt* (Para-RS), we randomly choose two paragraphs from a document and replace all the paragraphs between them (including those two as well) with the paragraphs of another document of the same prompt. Hence, the main theme of the replaced document is still intact but the logical sequencing would be slightly distorted. Note that, during replacement of the paragraphs, the positions of the chosen paragraphs of another document are the same as the positions of the to be replaced paragraphs of the current document. For example, if we want to replace paragraph number 3 and 4 of a document, then we choose paragraph number 3 and 4 of another document of the same prompt for replacement. In the Para-RS example of Figure 3.4, paragraph number 3 and 4 are replaced from paragraphs of another essay of the same prompt. Lastly, we perform a corruption called *Paragraph Replacement from Different Prompt* (Para-RD) which is same as the Para-RS but this time the paragraphs are replaced from another document of different prompt. Therefore, this corruption techniques produce incoherent documents where both main idea as well as logical sequencing are distorted. It is to be noted that, we hope to capture paragraph-level long range dependencies with these corruption strategies.

3.4.3 Discourse Corruption (DC) Pre-training

We treat DC pre-training as a multi-class (or binary) classification task where the encoder assigns a label to each document. In our experiments, we consider many combinations of corruption types (see Table 3.1). For example, for 6-way DC pre-training, the encoder tries to predict which class the document belongs to among the 6 classes (original essays, C-Para, M-Para, ParaDrop, Para-RS, Para-RD corrupted essays). For implementation, we add a classification layer on top of the base document encoder (Section 3.3.2). The classification layer consists of (i) a linear layer that takes $\mathbf{h}^{\text{base}}$ as input and (ii) a softmax layer. To train the model parameters, we minimize the cross-entropy loss function.

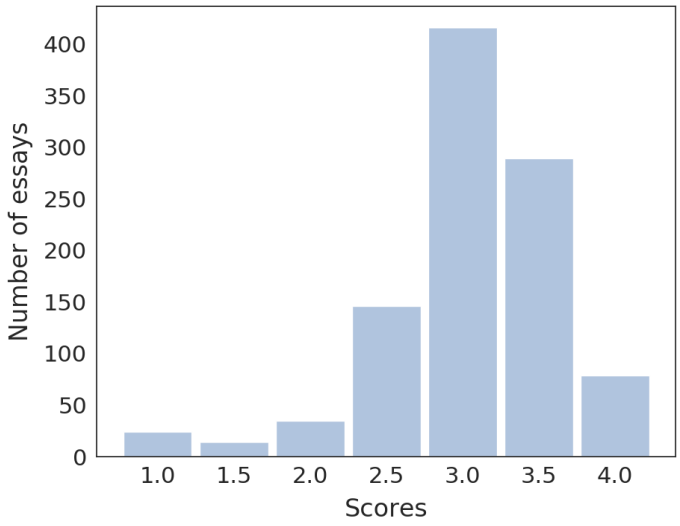

Figure 3.5: Distribution of Organization scores

3.4.4 Extension of Existing Pre-training Idea

We also propose an extension of the idea of next sentence prediction (N-SentP) task, i.e., *next paragraph prediction* (N-ParaP) pre-training. Same as N-SentP, the objective of N-ParaP pre-training is to predict if the second paragraph of a paragraph-pair is the actual next paragraph or not. We follow the same strategy as N-SentP for the generation of paragraph-pairs i.e., when we choose paragraphs A and B for each paragraph-pair, 50% of the time B is the actual next paragraph that follows A and 50% of the time B is a random paragraph.

For both N-ParaP and N-SentP, the random paragraph is either chosen from the same document (*randomS*) or from a different document in the corpora (*randomD*). If B is a random paragraph chosen from the same document, it means that the topic of the paragraphs A and B are the same. However, if B is a random paragraph chosen from a random document in the corpora, the topic of the paragraphs A and B is most likely to be different.

We hope to capture paragraph level dependencies to some extent with this pre-training. We treat the N-ParaP as a binary classification task that pre-trains paragraph-pair representations. We follow our two-step DC pre-training method for the implementation of this task.

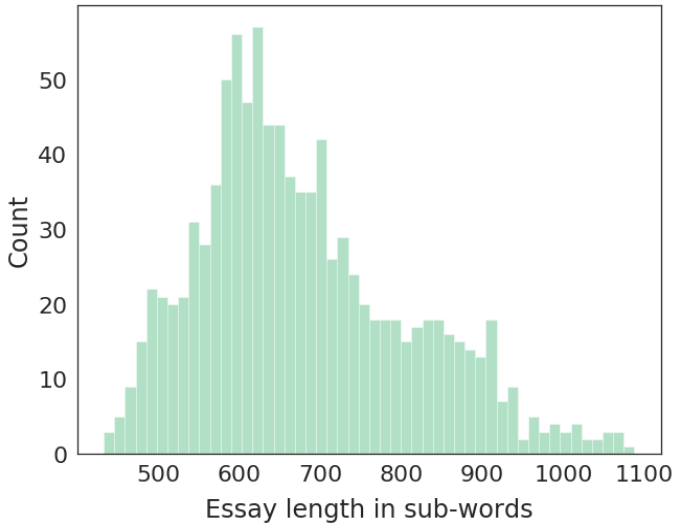

Figure 3.6: Histogram of lengths of ICLE essays used in scoring

3.5 Experimental Setup

3.5.1 Data

3.5.1.1 Essay Organization Scoring

We use the International Corpus of Learner English (ICLE) (Granger et al., 2009) for essay scoring which contains 6,085 essays and 3.7 million words. Most essays (91%) are argumentative and vary in length, having 7.6 paragraphs and 33.8 sentences on average (Wachsmuth et al., 2016). Some essays have been annotated with scores along multiple dimension among which 1,003 essays are annotated with Organization scores. The scores range from 1.0 (worst score) to 4.0 (best score) at half-point increments. The distribution of Organization scores is demonstrated in Figure 3.5. For our scoring task, we utilize these 1,003 essays. The average number of tokens per esssay is 679 (in sub-words) and the longest essay has 1,090 tokens. The histogram of the essay lengths is shown in Figure 3.6.

3.5.1.2 DC Pre-training

To pre-train the document encoder, we use four datasets, (i) Kaggle's Automated Student Assessment Prize (ASAP) dataset[3] (12,976 essays) (ii) TOEFL11 (Blanchard et al., 2013) dataset (12,100 essays), (iii) The International Corpus Network of Asian Learners of English (ICNALE) (Ishikawa, 2013) dataset (5,600 essays), and (iv) ICLE essays not used for Organization scoring (4,546 essays). In total, we acquire 35,222 essays

[3]https://www.kaggle.com/c/asap-aes

from the four datasets which are used during pre-training with N-SentP, SC, and DIC. However, for pre-training with all types of PC and N-ParaP, we use only 16,646 essays (TOEFL11 and ICLE essays) since ASAP and ICNALE essays are limited to single paragraphs.

3.5.2 Evaluation Procedure

We use five-fold cross-validation for evaluating our models with the same split as Persing et al. (2010) and Wachsmuth et al. (2016). However, our results are not directly comparable since our training data is smaller, as we reserve a validation set (100 essays) for model selection while they do not. We use mean squared error (MSE) as an evaluation measure. The reported results are averaged over five folds.

We evaluate two learning strategies of the encoder in the essay scoring task: *fine-tuning* and *fixed*. In the fine-tuning setting, both the pre-trained base document encoder and auxiliary encoder are fine-tuned on the essay scoring task. In the fixed setting, only the parameters of the auxiliary encoder are fine-tuned.

Our first baseline model is the *Base+AE* model. In our preliminary experiments, we first experimented with different settings such as fine-tune Base (pre-trained Longformer) model then merge AE, fine-tune both Base and AE and then merge, etc. However, we found that merging both models simultaneously (either in fine-tuning or fixed encoder setting) results in the best performance. Therefore, even for all the proposed systems, we merge the DC pre-trained Base model and AE at the same time in both fine-tuning and fixed-encoder settings. Our second baseline model is the *Base+AE* model pre-trained with the N-SentP task.

3.5.3 Preprocessing

We use the same preprocessing steps for both pre-training and essay scoring. We lowercase the tokens and specify an essay's paragraph boundaries with special tokens. Special tokens [CLS] and [EOS] are inserted at the beginning and end of each essay respectively. We normalize the gold-standard scores to the range of [0, 1]. During pre-training with SC and DIC, paragraph boundaries are not used.

For DIC, we collect 847 DIs from the Web.[4] We exclude the DI "and" since it is not always used for initiating logic (e.g., milk, banana *and* tea). In essay scoring dataset, we found 176 DIs and around 24 DIs per essay. In the pre-training data, the total number

[4]http://www.studygs.net/wrtstr6.htm, http://home.ku.edu.tr/~doregan/Writing/Cohesion.html etc.

of DIs is 204 and the average number of DIs per essay is around 13. We identified DIs by simple string-pattern matching.

3.5.4 Implementation Choices

From the two sizes of pre-trained Longformer models, we use Longformer-base model. The global attention of Longformer is set on the [CLS] token. For the auxiliary encoder, we use a BiLSTM with hidden units of 200 in each layer ($d^{\text{AUX}} = 200$).

We use Adam optimizer, batch sizes of 4 on the first-step of pre-training and batch sizes of 2 on the second-step of pre-training as well as on the essay scoring. The learning rate is set to $1e-5$ for pre-training and fine-tuning setting of essay scoring while it is set to 0.001 for fixed encoder setting of essay scoring. We use early stopping with patience 12 (5 for pre-training), and train the network for 100 epochs. In the pre-training phase, 80% of the data is used for training and 20% of the data is used for validation. We perform hyperparameter tuning for the scoring task and choose the best model. We tuned dropout rates (0.5, 0.7, 0.9) for all models on the validation set. To select hyper-parameters, we monitor performance on the validation set and choose the model that yields the lowest MSE. We choose the best model for each particular fold. In the testing phase, we re-scale the predicted normalized scores to the original range of scores and then measure the performance.

3.6 Results

3.6.1 Results of DC Pre-training

Table 3.1 shows the classification accuracy of both steps of DC pre-training on the validation data. We observe that the document encoder learns to distinguish not only between coherent/cohesive and incoherent/incohesive documents (binary classification) but also between different types of incoherent (3,4,5 and 6 way classification) documents.

Pre-training with C-DI provides the best classification accuracy. We anticipate that since we do not change the position of the DIs during shuffling, the encoder may only learn the sequence of DIs within each essay and try to distinguish between the DI sequence of original and corrupted essays. Therefore, the task becomes easier for the encoder.

Pretraining Phase	Classification Task	Objective/Corruption Type Used	Validation Accuracy
1st Step (All pre-training data)	Binary	N-SentP (randomS)	0.747
	Binary	N-SentP (randomD)	0.914
	Binary	N-ParaP (randomS)	0.764
	Binary	N-ParaP (randomD)	0.934
	Binary	C-Sent	0.955
	Binary	M-Sent	0.800
	Binary	C-DI	0.984
	Binary	M-DI	0.971
	Binary	C-Para	0.919
	3-way	C-Para, M-para	0.786
	4-way	C-Para, M-para, ParaDrop	0.770
	5-way	C-Para, M-Para, ParaDrop, Para-RS	0.707
	6-way	C-Para, M-Para, ParaDrop, Para-RS, Para-RD	0.734
2nd Step (Finetuned on ICLE pre-training data)	Binary	N-SentP (randomS)	0.728
	Binary	N-SentP (randomD)	0.878
	Binary	N-ParaP (randomS)	0.773
	Binary	N-ParaP (randomD)	0.958
	Binary	C-Sent	0.985
	Binary	M-Sent	0.781
	Binary	C-DI	1.000
	Binary	M-DI	0.998
	Binary	C-Para	0.890
	3-way	C-Para, M-Para	0.717
	4-way	C-Para, M-Para, ParaDrop	0.656
	5-way	C-Para, M-Para, ParaDrop, Para-RS	0.606
	6-way	C-Para, M-Para, ParaDrop, Para-RS, Para-RD	0.666

Table 3.1: Performance of classification tasks in the first step (using large-scale unlabeled essays) and second step of Corruption Pre-training (using unlabeled essays of target essay scoring corpus)

The visualization of document vectors obtained from the first and second step of DC pre-training (5-way classification task) is shown in Figure 3.7. To visualize the high-dimensional document vectors into a 2-dimensional space, we use dimensionality reduction algorithm T-Distributed Stochastic Neighbouring Entities (t-SNE). Figure 3.7 shows that the encoder is able to perfectly separate C-Para essays from other essays since the transition of ideas between paragraphs is fully distorted in these essays, hence easy to distinguish. We also observe that the encoder separates M-Para and ParaDrop essays better compared to Para-RS essays. Para-RS essays lie close to the original coherent essays and frequently overlap. We speculate that since we replace the paragraphs of the same positions, the sequencing of ideas of Para-RS essays is the least distorted compared to M-Para, ParaDrop or C-Para essays, hence these essays are similar to the original essays.

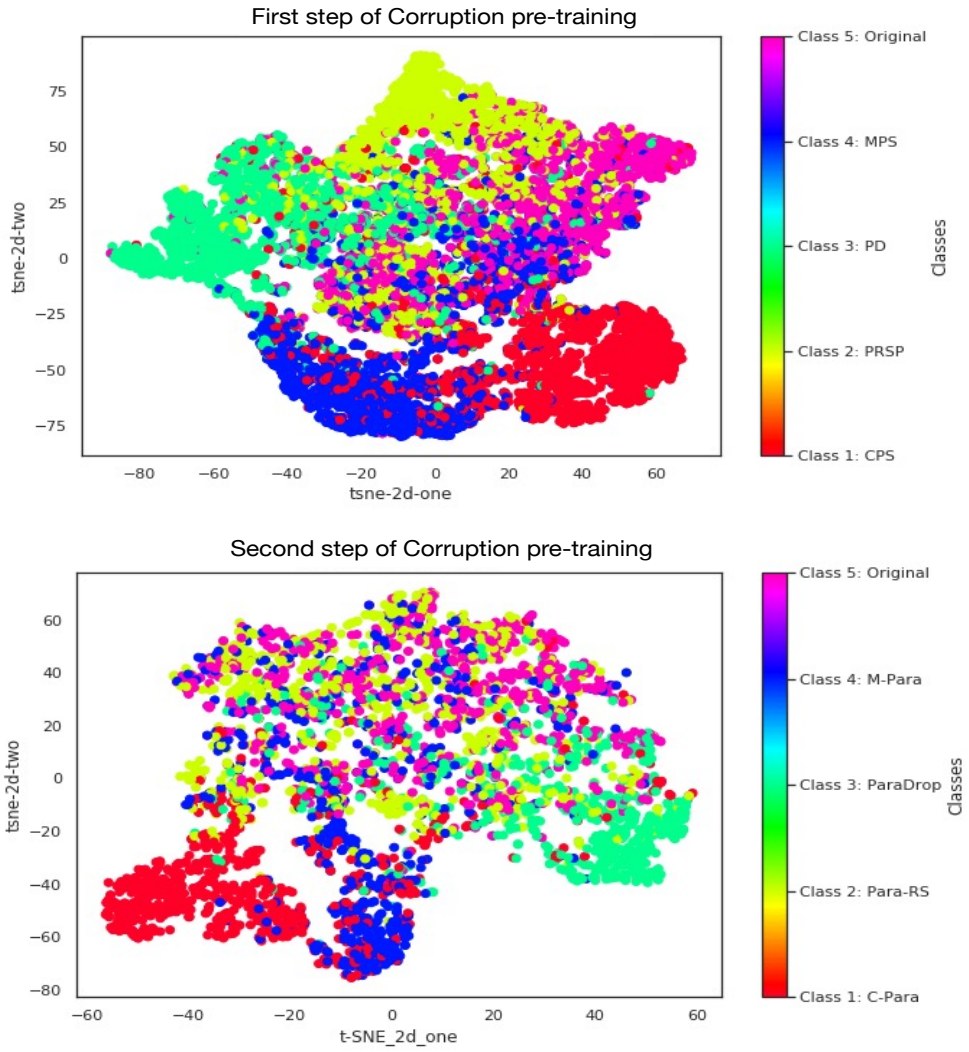

Figure 3.7: Visualization of document representations obtained from DC pre-trained (5-way classification scheme) encoder

3.6.2 Results of Essay Scoring

Table 3.2 lists MSE (averaged over five folds) of baseline models and our proposed systems (N-ParaP and DC pre-trained) for Organization scoring task.[5] It shows that the proposed unsupervised DC pre-training improves the performance of essay Organization scoring (statistically significant by Wilcoxon's signed rank test, $p < 0.05$) and we obtain significant performance gain over the baseline models. Also, we achieve new state-of-the-art result with our proposed method.

The best performance is obtained with the 5-way DC Pre-training. These results sup-

[5]Our model is Base+AE model (Section 3.3.2, 3.3.3). The performance of the Base (pre-traned Longformer) encoder without AE and without any DC pre-training when finetuned on essay Organization scoring is: MSE = 0.246

Model	Classification Task	Objective/ Corruption Type	Fine-tuning	Mean Squared Error Organization
Baseline 1	-	-	-	0.175
	-	-	✓	0.181
Baseline 2	Binary	N-SentP (randomS)	-	0.185
	Binary	N-SentP (randomS)	✓	0.196
	Binary	N-SentP (randomD)	-	0.184
	Binary	N-SentP (randomD)	✓	0.196
Proposed	Binary	N-ParaP (randomS)	-	0.177
	Binary	N-ParaP (randomS)	✓	**0.172**
	Binary	N-ParaP (randomD)	-	**0.172**
	Binary	N-ParaP (randomD)	✓	0.183
	Binary	C-Sent	-	0.184
	Binary	C-Sent	✓	0.198
	Binary	M-Sent	-	0.175
	Binary	M-Sent	✓	0.193
	Binary	C-DI	-	0.189
	Binary	C-DI	✓	0.185
	Binary	M-DI	-	0.183
	Binary	M-DI	✓	0.198
	Binary	C-Para	-	**0.172**
	Binary	C-Para	✓	**0.167***
	3-way	C-Para, M-Para	-	**0.173**
	3-way	C-Para, M-Para	✓	**0.162***
	4-way	C-Para, M-Para, ParaDrop	-	**0.169**
	4-way	C-Para, M-Para, ParaDrop	✓	**0.157***
	5-way	C-Para, M-Para, ParaDrop, Para-RS	-	**0.166***
	5-way	C-Para, M-Para, ParaDrop, Para-RS	✓	**0.155***
	6-way	C-Para, M-Para, ParaDrop, Para-RS, Para-RD	-	0.179
	6-way	C-Para, M-Para, ParaDrop, Para-RS, Para-RD	✓	**0.162***
Persing et al. (2010)				0.175
Wachsmuth et al. (2016)				0.164

Table 3.2: Performance of essay scoring. Numbers in **bold** and **underline** denote improvement over baseline and previous state-of-the-art respectively. '*' indicates a statistical significance (Wilcoxon signed-rank test, $p < 0.05$) against the baselines.

port our hypothesis that training with corrupted documents helps a document encoder learn logical sequence-aware text representations. In most of the cases, fine-tuning the encoder for scoring task provides better performance.

From Table 3.2 we observe that next paragraph prediction or paragraph corruption based DC pre-training is effective for Organization scoring while sentence and DI corruption based pre-training is not. This could be attributed to the fact that the paragraph level transition of ideas (global coherence) is not captured by sentence and DI level corruption. Besides, a manual inspection of DIs identified by the system shows that the identification of DIs is not always reliable. Almost half of DIs identified by our simple pattern matching algorithm (see Section 3.5.3) were not actually DIs (e.g., *we have survived **so** far only external difficulties*). We also found that some DI-shuffled documents are often cohesive. This happens when original document counterparts have two

Model	Classification Task	Corruption Type	Fine-tuning	Mean Squared Error Organization
Baseline 1	-	-	-	0.175
	-	-	✓	0.181
Baseline 2	Binary	N-SentP (randomS)	-	0.185
	Binary	N-SentP (randomS)	✓	0.196
	Binary	N-SentP (randomD)	-	0.184
	Binary	N-SentP (randomD)	✓	0.196
Proposed	5-way	C-Para, M-Para, ParaDrop, Para-RS	-	**0.166***
	5-way	C-Para, M-Para, ParaDrop, Para-RS	✓	**0.155***
	5-way to Binary	C-Para, M-Para, ParaDrop, Para-RS	-	0.179
	5-way to Binary	C-Para, M-Para, ParaDrop, Para-RS	✓	0.185
	5-way to 3-way	C-Para, M-Para, ParaDrop, Para-RS	-	0.181
	5-way to 3-way	C-Para, M-Para, ParaDrop, Para-RS	✓	**0.162***

Table 3.3: Essay scoring results when a 5-way DC pre-training is reduced to a Binary and 3-way DC pre-training

or more DIs with more or less same meaning (e.g., *since* and *because*).

It can be seen that as the classification task of Corruption Pre-training becomes more complicated by adding more corruption types, the essay scoring performance improves (except for 6-way classification). We obtain the best performance with 5-way classification task. We speculate that this is because with more corruption types, the model learns more styles of transition of ideas among paragraphs as well as differences between them. Finally, the model connects those differences to scores at the essay scoring phase by figuring out which flow of concepts is better than the other.

It should be noted that 6-way classification task could not outperform 5-way classification task. This might be because of adding Para-RD corruption in 6-way classification task. Since in Para-RD, we replace the paragraphs of document with paragraphs of a document of different prompt, instead of learning the flow of the ideas throughout the text the encoder might also be learning something else (e.g, topic difference). We speculate that this confuses the document encoder at the essay scoring phase.

3.7 Analysis

3.7.1 Importance of Fine-grained Corruption Types

To investigate how important it is for the model to learn the difference between fine-grained corruption types, we collapsed four corruption types into one or two classes in

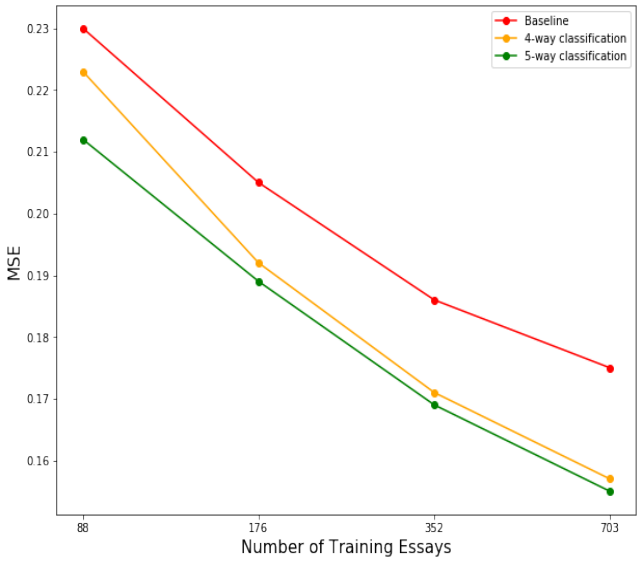

Figure 3.8: Plot of training data vs MSE at essay scoring phase

DC pre-training. Specifically, we reduced the best performing 5-way DC pre-training into (i) binary DC pre-training with original v.s. corrupted essays ({C-Para, M-Para, ParaDrop, Para-RS}), and to (ii) 3-way DC pre-training with original v.s. fully corrupted (C-Para) v.s. partially corrupted essays ({M-Para, ParaDrop, Para-RS}).

Table 3.3 demonstrates the results. It shows that transforming 5-way classification to binary classification performs worse than the baseline. We attribute this to combining fully corrupted (CPS) essays with partially corrupted (MPS, PD, PRSP) essays, so the model cannot distinguish between extremely bad and relatively bad essays. This hypothesis is solved when we transform it to a 3-way classification task. We obtain much better performance during finetuning, but the performance is not as good as the original 5-way classification task. Overall, these experiments indicate that differentiating between fine-grained corruption types is essential.

3.7.2 Effectiveness of Corruption Pre-training in Low Resource Setting

To investigate how beneficial our DC pre-training is when labeled data is less available, we reduce the training data at the essay scoring phase. We examine the two best performing DC pre-trained models (4-way and 5-way classification) and compare them with the baseline model (model without DC pre-training). We select Baseline 1 for comparison since it has the best result among 2 baselines.

Figure 3.8 shows a plot of number of training essays vs. MSE. MSE is obtained with all

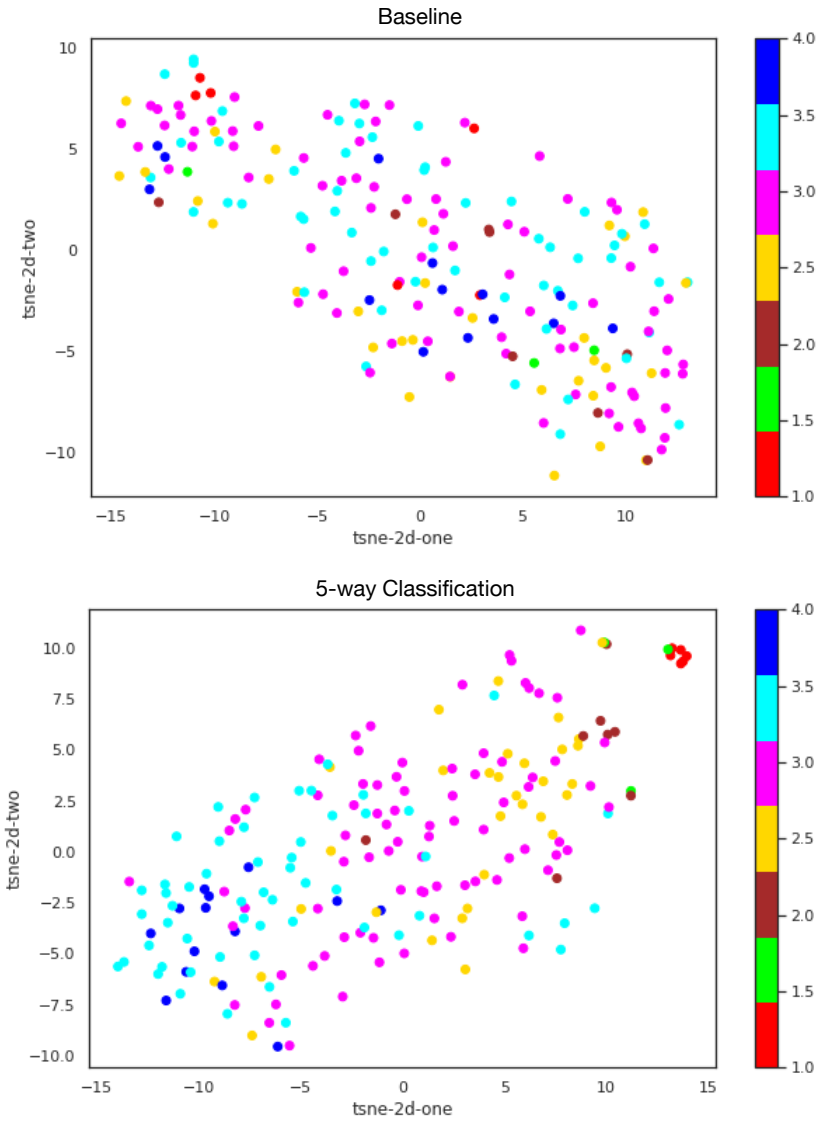

Figure 3.9: Visualization of essay representations

training data (703 essays) as well as with training data being reduced to $\frac{1}{2}$ (352 essays), $\frac{1}{4}$ (176 essays) and $\frac{1}{8}$ (88 essays). We observe that our proposed models constantly outperform the baseline model when we reduce the training data. This indicates both the strength and effectiveness of our DC pre-training with less information from labeled data and that the model understands which Organization structure is better than the others.

Our 4-way DC model (indicated via orange line) does not perform better than the 5-way DC model (green line). This result indicates that having more fine-grained corruption types in DC pre-training helps the model to be less dependent on the annotated information of which essay Organization is better.

Gold Score	Baseline Predicted	5-way Predicted	MSE (gold&baseP)	MSE (gold&5-wayP)
1.0	2.3	1.1	1.69	0.01
2.5	1.2	2.1	1.69	0.16
2.5	3.7	2.5	1.44	0.00
2.5	1.4	2.5	1.21	0.00
1.0	2.3	1.7	1.69	0.49
2.0	3.3	2.9	1.69	0.81
2.0	2.9	2.4	0.81	0.09
4.0	3.1	3.6	0.81	0.16
4.0	3.2	3.7	0.64	0.09
1.5	2.5	2.2	1.00	0.49

Table 3.4: Score prediction of test instances by baseline and our best DC pre-trained model

3.7.3 Essay Embeddings

In order to identify which scores are better distinguished by our models than the baseline model, we visualized essay embeddings (i.e. $\mathbf{h}^{\text{base}}$) obtained from the fine-tuned baseline model[6] and our proposed DC pre-trained (5-way classification) model.

The results are shown in Figure 3.9. In the baseline model essay embeddings, the essays are scattered, and the low-scored essays (scored 1, red dots) are sometimes close to the high-scored essays (scored 4, blue dots) (upper-left of the figure). In contrast, the essay representations of our DC pre-training (5-way classification) shows that our model is good at separating essays of different scores and more cluster of scores appear compared to the baseline model. The highest scored (scored 4, blue dots) and the lowest scored (scored 1, red dots) essays are at the complete opposite position and furthest from each other in the embedding space. This means our model knows the difference between high scored and low scored Organization. We see that the lowest scored essays (red dots) are clustered and fully separated from other essays. Besides, other low scored essays (scored 1.5 and 2.0, lime and brown dots respectively) as well as highest scored essays (scored 4, blue dots) are also well distinguished. This represents that our model is not only good at separating bad Organization from good ones, but our model is also good at distinguishing different levels of "goodness" of essay Organization.

Table 3.4 presents 10 test instances for which the prediction of our DC pre-trained

[6]We select Baseline 1 for the visualization of essay embeddings since it has the best result among 2 baselines

Model	Classification Task	Corruption Type	Fine-tuning	Mean Squared Error Organization
Baseline 1	-	-	-	0.175
	-	-	✓	0.181
Baseline 2	Binary	N-SentP (randomS)	-	0.185
	Binary	N-SentP (randomS)	✓	0.196
	Binary	N-SentP (randomD)	-	0.184
	Binary	N-SentP (randomD)	✓	0.196
Proposed	Binary	N-ParaP	-	0.177
	Binary	N-ParaP	✓	**0.172**
	5-way	C-Para, M-Para, ParaDrop, Para-RS	-	**0.166***
	5-way	C-Para, M-Para, ParaDrop, Para-RS	✓	**0.155***
	5-way + Binary	(C-Para, M-Para, ParaDrop, Para-RS) + N-ParaP	-	0.178
	5-way + Binary	(C-Para, M-Para, ParaDrop, Para-RS) + N-ParaP	✓	**0.173**
	Binary + 5-way	N-ParaP + (C-Para, M-Para, ParaDrop, Para-RS)	-	0.181
	Binary + 5-way	N-ParaP + (C-Para, M-Para, ParaDrop, Para-RS)	✓	**0.162***

Table 3.5: Essay scoring results of 5-way DC pre-training combined with next paragraph prediction (N-ParaP) pre-training

model is better (i.e., lower MSE between gold and predicted score) than the baseline model. Column 1 shows the gold essay score, columns 2 and 3 show the scores predicted by the baseline model and our best DC pre-trained model (5-way classification) respectively.[7] Column 4 shows the MSE between the gold score and baseline predicted score, whereas column 5 presents the MSE between the gold score and the score predicted by DC pre-trained model. Table 3.4 shows that our DC pre-trained model predicts low-to-medium and high essay scores well in comparison to the baseline. Observing the MSE difference between columns 4 and 5, one can see how better DC pre-trained model's prediction is in comparison to the baseline.

3.7.4 Combining Different Pre-training

We have observed that (from Table 3.2) N-ParaP pre-training improves the Organization scoring performance a bit although not as much as DC pre-training. In order to further analyse the effect of different pre-training, we have combined our DC pre-training with N-ParaP pre-training (e.g., first pre-train the model with the next paragraph prediction task and then pre-train it again with DC corruption strategies). For this combined pre-training task, we choose our best DC pre-trained model (5-way classification). However, The results in Table 3.5 show that combining paragraph level pre-training with document level DC pre-training doesn't perform very well, i.e., the proposed DC pre-training performs better without any additional local pre-training.

[7]The predicted scores are shown to one decimal place.

3.8 Conclusion

In this work, we proposed an unsupervised pre-training strategy to capture discourse structure (i.e., coherence and cohesion) of essay Organization. We have presented various token, sentence, and paragraph level corruption techniques that produce several types of fully corrupted (totally incoherent/incohesive) or partially corrupted (partially incoherent/incohesive) essays. Then, we train a document encoder to discriminate between original essays and their artificially corrupted essays in order to make the encoder logical-sequence aware. Afterwards, the logical-sequence aware encoder is used to obtain feature vectors of essays for the task of essay Organization scoring. Our proposed pre-training strategy does not require any expensive parser or annotation. The experimental results show that the proposed method successfully captures the discourse structures of essay Organization, and we obtain a new state-of-the art result for essay Organization scoring. Our results also show that the combination of MLM pre-trained document encoder and paragraph level discourse corruption pre-training is effective for capturing the discourse of essay Organization. The combination of these two can handle both global and local coherence.

One possible future direction of this work is to determine how to exploit other unannotated argumentative texts (except student essays) for the proposed pre-training method. Since student essays are not perfect (i.e., can contain grammatical and/or spelling errors), it would be interesting to see how the proposed method behaves when pre-trained with perfectly written or error-less texts. We hope that our work inspires the exploration of new ways of unsupervised encapsulation of discourse structure in text representation.

Chapter 4

Capturing Logic Patterns in Argumentation

4.1 Introduction

Argumentation plays a central role in human communication, where refuting or attacking others' arguments is a common persuasion strategy (Walton et al., 2010). *Attack* in arguments can have different modes e.g., the counterargument can deny the *conclusion* (i.e., statement that expresses the position or belief of the arguer) of the attacked argument or it can deny a *premise* (i.e., statement that provides support or reason for the conclusion) of the attacked argument or the counterargument can deny an *argumentative relation* (i.e., support or attack) in the attacked argument. These forms of attack are commonly known as *rebuttal, undermining* and *undercut* in the argumentation theory respectively (Walton, 2009; Cramer and Guillaume, 2018).

Beside of having different forms, attacks in arguments often comprise complex rhetorical moves as well e.g., one might agree with a premise while attacking the conclusion of the argument or one might agree with and challenge a premise at the same time that ultimately leads to denying the conclusion (Afantenos and Asher, 2014). Furthermore, arguments generally consist implicit knowledge (e.g., causal reasoning), sentiments (e.g., positive or negative feeling towards a certain concept or element) (Reisert et al., 2018; Jo et al., 2021a; Saha et al., 2021), presupposition or value judgements (e.g., presupposing that some consequence has greater importance or value than another consequence) which contribute to the internal logical structure of attacks.

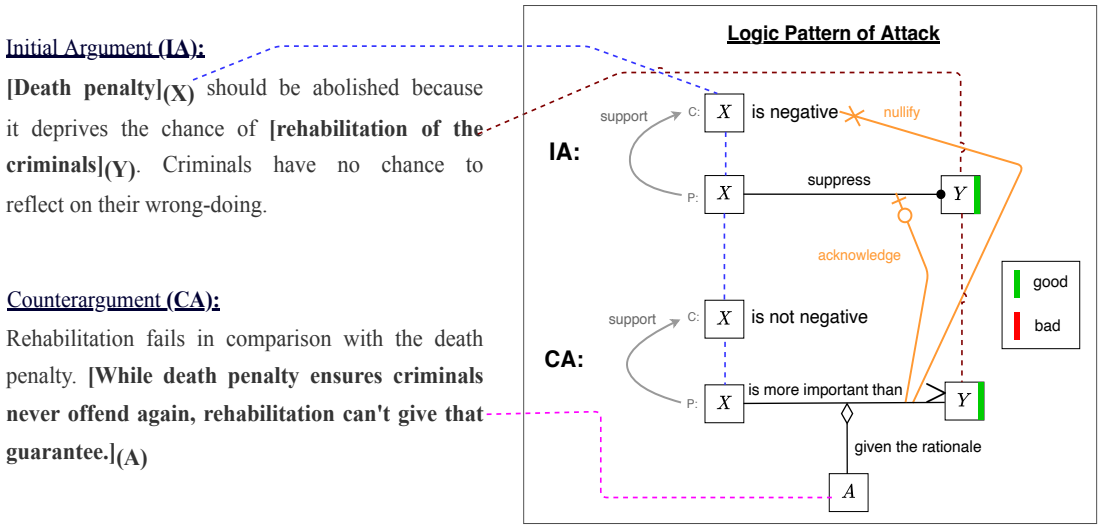

Figure 4.1: An example of logic pattern of attack of a debate captured by the proposed LPAttack annotation scheme.

Consider an example debate in Fig. 4.1, where opposing teams give two argumentative speeches. In the debate, the counterargument (CA) does not deny the *premise* of the initial argument (IA), i.e., *death penalty deprives the chance of rehabilitation of the criminals*. Instead, she implicitly agrees with it while denying the *conclusion* of the IA, i.e., *death penalty should be abolished* by giving more importance or value to the *death penalty* than the *rehabilitation of the criminals*. Although this value judgment is implicit in the CA speech, CA explicitly provides a reason behind her value judgment (bold text in CA). Automatically identifying such internal logic patterns can help a wide range of natural language processing (NLP) applications. For example, in an educational domain, this can help machines diagnose learners' arguments and provide feedback to the learners.

Prior studies in NLP that focused on *attacks* in arguments mainly worked on the classification of argumentative relations (e.g., support, attack, neutral), identifying attackable points in arguments, or counterargument generation (Stab and Gurevych, 2014a; Deguchi and Yamaguchi, 2019; Kobbe et al., 2019; Jo et al., 2021a; Walton et al., 2008; Jo et al., 2020; Wachsmuth et al., 2018; Hua et al., 2019; Reisert et al., 2019; Alshomary et al., 2021; Jo et al., 2021b). Comparatively, less attention has been paid to identifying the logic pattern of attacks in arguments.

Although some recent studies (Reisert et al., 2018; Jo et al., 2021a) developed annotation schemes and logical mechanisms to capture the reasoning process behind support and attack relations where they exploited implicit causal links and sentiments, these

studies did not capture other implicit information, e.g., presupposition or value judgments in arguments that also contribute to the underlying logical structure of attacks. Furthermore, none of these studies capture the modes of *attack* (e.g, whether the counterargument denies the conclusion or the premise of the attacked argument) and the complex rhetorical moves (e.g., agreeing with a premise while attacking the conclusion).

To address these gaps, we introduce LPAttack (**L**ogic **P**attern of **Attack**), a new annotation scheme that captures common modes of attacks and complex rhetorical moves in them as well as the implicit information and value judgments that contribute to the logical structure of attacks. Fig. 4.1 shows an example annotation. The logic pattern of IA speech is represented by our logic pattern, which can be interpreted as follows: *death penalty (=X) is considered a negative thing because death penalty suppresses chance of rehabilitation of the criminals (=Y), something good*. The logic pattern of CA speech then represents their value judgment on the *death penalty* and *chance of rehabilitation of the criminals*, where more value is given to the *death penalty*. This value judgment then attacks the conclusion of IA. Given that information, one can understand how and which part of the IA is attacked by the CA.

Our contributions can be summarized as follows:

- We introduce LPAttack, a novel annotation scheme that captures the common modes and complex rhetorical moves in attacks along with the implicit information, presuppositions, or value judgments (§4.3).
- We conduct an annotation study using the proposed scheme that yields moderate agreement between two annotators indicating the feasibility of the human annotation for the scheme (§4.4).
- We provide the annotated corpus comprising logic patterns of attacks of 250 debates and the annotation guidelines as a publicly available resource to encourage future research[1].

4.2 Related Work

Computational analysis of argumentation has gained considerable attention in recent years because of its importance in many NLP applications such as essay scoring, argu-

[1]Our annotated corpus and annotation guidelines are publicly available at https://github.com/cl-tohoku/LPAttack

mentative writing support systems, and educational feedback. Common lines of work in this area include argumentative units (e.g., claim, premise) identification (Levy et al., 2014; Rinott et al., 2015; Stab and Gurevych, 2014a), argumentative relations (e.g., support, attack, neutral) classification (Peldszus and Stede, 2015; Cocarascu and Toni, 2017; Niculae et al., 2017; Stab and Gurevych, 2014a; Deguchi and Yamaguchi, 2019; Kobbe et al., 2019; Jo et al., 2021a), qualitative assessment of arguments (Persing et al., 2010; Persing and Ng, 2013, 2014, 2015, 2016b; Rahimi et al., 2015; Wachsmuth et al., 2016; Habernal and Gurevych, 2016; Wachsmuth et al., 2017; Mim et al., 2019b,a, 2021) and retrieval or generation of counterargument (Hua and Wang, 2018; Wachsmuth et al., 2018; Hua et al., 2019; Reisert et al., 2019; Alshomary et al., 2021; Jo et al., 2021b)

Lately, researchers have started focusing on one of the complex and challenging facet of argument analysis, i.e., capturing or explicating the encapsulated knowledge in arguments (e.g., causal knowledge, commonsense knowledge, factual knowledge) which are often implicit (**?**Hulpus et al., 2019; Becker et al., 2019, 2020; Al-Khatib et al., 2020; Becker et al., 2021b,a; Singh et al., 2021; Saha et al., 2021). Although for a deeper understanding of argumentation, we also need to comprehend the underlying reasoning patterns of arguments, less attention has been paid to representing such underlying reasoning patterns and explicating the implicit information that contribute to these patterns.

We focus on this gap and address the problem of explicating internal logic pattern of attacks in arguments that comprise complex rhetorical moves and implicit causal information, sentiments, presuppositions as well as value judgments. Our inspiration for designing such an annotation scheme comes from Walton's argumentation schemes (Walton et al., 2008) which represent the common reasoning structures in arguments. For example, Walton's scheme of *Argument from Negative Consequences* has the conclusion that *A should not be brought about*, which is supported by the premise that *if A is brought about, then bad consequences will occur*. Although Walton's schemes explicate the unstated assumptions or propositions as a form of reasoning pattern, they are not intended to capture the logic pattern of *attacks*, i.e., how a counterargument attacks an argument. Note that each of Walton's schemes has a set of critical questions (CQs) that are used to judge if an argument fitting a scheme is good or fallacious. Some CQs for the above scheme are *How strong is the likelihood that the cited consequences will occur?*, *Are there other opposite consequences that should be taken into account?*. However, the CQs in Walton's schemes only specify the attackable points in an argument, they do not represent the reasoning pattern of attacks.

Some recent studies adopted Walton's schemes to represent the logic behind support and attack relations. One of these studies (Reisert et al., 2018) developed an annotation scheme that uses argument templates to capture reasoning patterns behind support and attack relations. Another study (Jo et al., 2021a) composed a set of rules specifying logical mechanisms that signal the support or attack relation. Although these studies identified implicit causal reasoning, sentiments, or factual contradiction in attacks, they did not capture other implicit information, such as contradictory causal reasoning, or assumptions or value judgments that significantly contribute to the logical structure of attacks.

One recent study (Saha et al., 2021) created commonsense explanation graphs that illustrate the commonsense reasoning process involved in inferring support and attack relations. However, the focus of this study is a commonsense explanation, not the reasoning pattern of *attack* (or support). Therefore, although this study exploited implicit causal knowledge, the fine-grained implicit knowledge explicated in this study is not effective in representing the logic pattern of *attacks*, which requires distinct coarse-grained implicit information, e.g., contradictions or value judgments in arguments.

Additionally, there is still no work in computational argumentation that captures the modes of attacks in arguments (e.g, whether the counterargument denies the conclusion or the premise of the attacked argument) or the complex rhetorical moves in them (e.g., agreeing with a premise while attacking the conclusion, providing a contradictory premise that leads to denying the conclusion etc.). Our work addresses these gaps by introducing an annotation scheme that can capture common modes of attacks, complex rhetorical moves in them as well as implicit causal reasoning, sentiments, presuppositions or value judgments that contribute to the logic pattern of attacks.

4.3 LPAttack Annotation Scheme

We hypothesized that the logic pattern of attacks in arguments is not uniformly distributed but is rather highly skewed, and following this hypothesis, we developed our annotation scheme to capture the common logic pattern of attacks in arguments.

4.3.1 Pre-Study and Scheme Design

To examine what sort of strategic moves, assumptions or value judgments are common during an attack, we conducted a preliminary qualitative analysis of how one argument attacks another (see Appendix for further details). For this pre-study, we selected 35

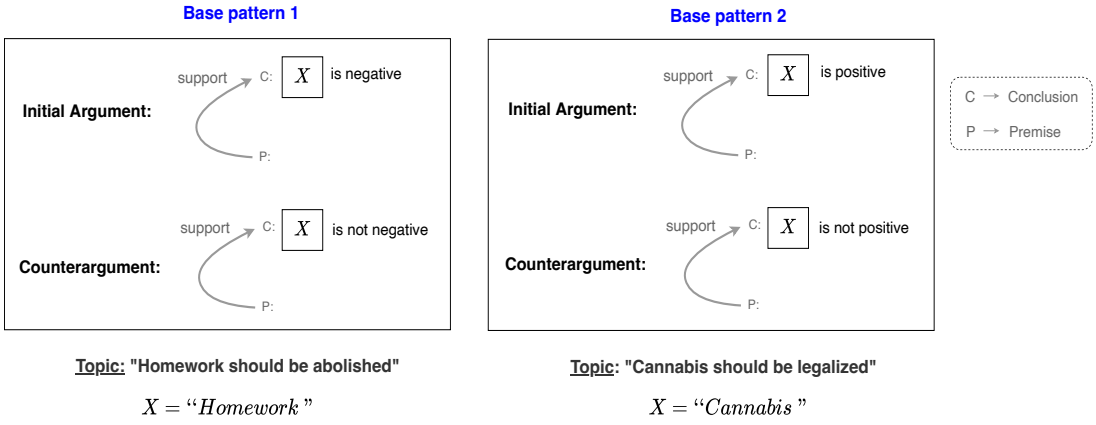

Figure 4.2: Base logic patterns with examples.

debates from the TYPIC dataset[2] (Naito et al., 2022) comprising multiple, diverse debate themes. Each debate comprises an argument and a counterargument conveyed by two opposing teams.

This analysis of the internal structure of attacks provided insights into how we can represent the attacking logic so that human annotation is plausible. Based on these insights, we designed our annotation scheme, defined the annotation guidelines, and formulated the task of capturing the logic pattern of attacks in arguments. Two untrained annotators then explored the initial designs and helped improve the overall design and guidelines. The primary feedback from the annotators included suggestions for a detailed description for each of the relations and attributes, creating categories for them, and having a prioritization map for these relations and attributes. In the following subsections, we describe our annotation scheme and the prliminary study findings.

4.3.1.1 Base Logic Patterns

Generally, when people argue *for* a belief, they show positive sentiment toward a concept of that belief. Conversely, when they argue *against* a belief, they exhibit negative sentiment toward a concept. For example, for the beliefs *"homework should be abolished"* and *"death penalty should be abolished"*, the arguers have an *against* stance. In both cases, they have *negative* sentiments toward the concepts of *homework* and *death penalty*. Now, counterarguments generally have the opposite stance and sentiment of the initial argument. For example, the counterargument *homework should not be abolished* has a *for* stance and a *non-negative or positive* sentiment toward the concept of *homework*. No matter how diverse the argument topic, the sentiment toward a certain

[2]This dataset is publicly available at https://github.com/cl-tohoku/TYPIC

concept is generally dependent on these *for* or *against* stances.

Argumentation Schemes (Walton et al., 2008) highlighted this fact and extensively utilized the positive or negative sentiment of the arguer toward a certain concept or consequence. Motivated by that, we designed two base patterns (shown in Fig. 4.2) for our scheme where the sentiments toward the main concept in the argument function as the *conclusion* of the argument. *Base pattern 1* represents the *against* stance of the initial argument and therefore presents the logic: {*Initial Argument*: X is negative; *Counterargument*: X is not negative} where X is a slot for the concept. Conversely, *Base pattern 2* represents the case where the initial argument has a *for* stance. The base patterns have two slots for premises: one in the initial argument and the other in the counterargument.

During our pre-study, we noticed that sometimes the counterargument shows strong opposite sentiment toward an argument e.g., {*Argument: X is negative; Counterargument: X is positive*} and sometimes the opposite sentiment of the counterargument is not that strong e.g., {*Argument: X is negative; Counterargument: X is not negative*}. However, the strength of the opposite sentiment usually depends on human perception and may thus vary. To reduce the complexity and confusion during human annotation, we only maintained the representation of the less strong opposite sentiment of counterargument as shown in Fig. 4.2.

4.3.1.2 Relations and Attributes

To capture the logic of the *premise* that will support the conclusion (i.e., the sentiment toward the central concept), we designed a set of relations and attributes. See Table 4.1 and Table 4.2 for an overview.

Causal relation Previous studies on the representation of implicit reasoning behind support or attack mostly adopted Argumentation Schemes (Walton et al., 2008) and have shown that a majority of arguments can be represented by the implicit causal links (Reisert et al., 2018; Al-Khatib et al., 2020; Jo et al., 2021a; Singh et al., 2021). In our pre-study, we observed similar phenomena. Most of the logics in arguments attacked or acknowledged by counterarguments can be represented by causality. For example, in Fig. 4.1, the logic of IA that the *death penalty deprives the chance of rehabilitation of the criminals* can be represented with the "suppress" causality: {*death penalty, suppress, the chance of rehabilitation of the criminals*}. We thus designed our annotation scheme around two causal relations, *"promote"* and *"suppress"* (henceforth, *"base relations"*).

Relations	Description	Example
promote	represents something causing/ encouraging another thing	no homework *promote* free time
suppress	represents something hindering/ preventing another thing.	homework suppress free time
rationale/condition	represents writer's reasoning or justification behind a relation or attribution.	homework is more important than free time given the *rationale/condition* that homework is part of education
X, Y: (Y) is more important/severe/ has greater weight than (X); X ≥ Y: (X) is more important/severe/ has greater weight than (Y)	(i) represents some relation has higher value than another or (ii) some concept or element has higher value than another	(i){no homework promote people fail in exam} which *is more important/ severe/ has greater weight* than {no homework promote free time}, (ii) Example in Fig. 4.1
contradiction	represents opposing logics	{homework promote "problems in family"} *contradicts* {homework promote good family relation}
acknowledgement	represents agreement between relations	Example in Fig. 4.1
nullify (attacking relation)	represents denying a relation or logic	Example in Fig. 4.1
limit (attacking relation)	represents agreeing with and denying a relation at the same time	{death penalty promote executioner's suffering can be mitigated given the condition that executioners have a good mental support system} which *limit* {death penalty" promote executioner's suffering}
π function	represents joining of two or more relations	joining the two relations {homework suppress free time} and {free time promote unproductive activities} would produce the relation {homework suppress unproductive activities}

Table 4.1: Relations in LPAttack scheme

Value judgement We observed that one common reasoning during an attack is based on value-judgements i.e., comparing two factors by giving more value or importance to one than the other. We have found two phenomena: (i) counterarguments give more importance to a certain concept of logic while implicitly acknowledging the logic, as

Attributes	Description	Example
¬ negation	represents negation form of a relation or concept	{homework *doesn't* promote free time} or {*no* homework promote free time}
△ mitigation	represents mitigated form of a relation	{death penalty promote executioner's suffering can be *mitigated* given the condition that executioners have a good mental support system}
good	represents positive feeling of the arguer towards a concept	{homework" should be abolished because homework suppress free time}. Here "free time" is a *good* thing according to the arguer
bad	represents positive feeling of the arguer towards a concept	{death penalty should be abolished because death penalty promote executioner's suffering }. Here, "executioner's suffering" is a *bad* thing according to the arguer

Table 4.2: Attributes in LPAttack scheme

shown in Fig. 4.1, and (ii) counterarguments neither acknowledge nor deny any logic of the initial argument, instead ignore it and deny the conclusion of the initial argument by providing new reasons presupposing that the new reasons have more value. Consider the following example:

(3) **Initial Argument (IA)**
...homework should be abolished (Conclusion)
...if homework were to be abolished, we could have more free time. As a result, we could do what we really wanted like club activities... (Premise)
Counterargument (CA)
.....if homework is abolished, a number of people who don't study at all will increase.....To decrease a number of people who repeat years, homework is necessary...

In Example (3), the CA neither affirms nor denies the IA's logic, but ignores it and provides new reasons that deny the conclusion "homework should be abolished". The CA presupposes that the value or importance of {*if homework is abolished, a number of people who don't study at all will increase*} is greater than the value of {*if homework were to be abolished, we could have more free time*}, and this presupposition is implicit in the CA's argument. To represent the two phenomena of value judgments, we created

the relation *"is more important or severe or has greater weight"*.

Contradiction Another common attacking strategy in counterarguments is providing contradictory logic that ultimately leads to denying the conclusion of the initial argument. One example is given below:

(4) **Initial Argument (IA)**
..death penalty should be abolished (Conclusion)
...death penalty is causing brutalization of modern society.....it validates the notion that the taking of someone's life is a valid choice..... (Premise)
Counterargument (CA)
...death penalty sends a message that taking an innocent life will not be tolerated by a civilized society. Thus, it serves as an antidote to brutality...

Although the IA says: {*death penalty is causing brutalization of modern society*}, the CA says the opposite: {*it serves as an antidote to brutality*}. CA's logic contradicts IA's, which leads to denying the conclusion of the IA. To capture such contradictory logic, we invented the *"contradiction"* relation.

Logic denial/agreement To explicitly represent the denial of a premise's logic or conclusion, we created two relations, *"nullify"* and *"limit"*. These relations are considered the *"attacking relations"* in our scheme. We represent agreeing with a logic by the relation "acknowledgment".

Negation Counterarguments commonly *negate* (explicitly or implicitly) certain logic, especially causal reasoning, by providing some rationales or conditions. Consider the following example:

(5) **Initial Argument (IA)**
...homework should be abolished (Conclusion)
...if students are always given homework, they will always be waiting for instructions.....homework should be abolished so that students can study on their own initiative.... (Premise)
Counterargument (CA)
...students will hardly be able to study on their own without homework because continuous instructions or guidelines are needed for children to study or learn something new...

In Example (5), CA negates IA's logic {*homework should be abolished so that students can study on their own initiative*} by saying {*students will hardly be able to study on their own without homework*} and provides a reason behind it i.e., *"continuous instructions or guidelines are needed for children to study or learn something new"*. We developed the "negation" attribute and "rationale/condition" relation to represent such negation attribution and reasoning behind a logic.

Mitigation Instead of completely negating a logic, counterarguments often express that the severity of it can be mitigated. Consider the following example:

(6) **Initial Argument (IA)**
...death penalty should be abolished (Conclusion)
...death penalty causes executioner's suffering.....they feel that they are responsible themselves for killing the suspect.... (Premise)
Counterargument (CA)
..executioner's stress can be reduced by making sure that would-be executioners are fully prepared for the job and have a good mental support system

In this example, the CA does not completely negate IA's logic {*death penalty causes executioner's suffering*} instead partially negate it by saying {*executioner's stress can be reduced*}. We created a "mitigation" attribute to represent such partial negation attribution.

4.3.1.3 Slot-filling

To computationalize the task of capturing the logic pattern of attacks, we explored if it is possible to represent the logic patterns using only the information present in the argumentative texts. Our preliminary analysis suggested that it is fairly possible to represent the logic behind an attack without any external commonsense concepts. We thus decided to formulate the task of slot filling as a text-span selection task: annotators choose slot fillers in the base patterns from the given arguments.

4.3.2 Task Setting

The task of representing the logic pattern of attack for a given argument and counterargument consists of the following steps:

1. *Selection of base logic pattern and slot-filling:* A base logic pattern is selected based on the central stance of initial argument. Then, the slots of the pattern are

filled with the central concept.

2. *Selection of relations and attributes along with text-spans:* Relations and attributes are chosen along with the text-spans from the given arguments to complete the base pattern by representing the logic of the premises.

Since there is no fixed template for representing the logic of the premises in the initial argument and counterargument of the base pattern, one important question is how many relations, attributes and text spans should be chosen for premise representations and how to choose them. In this regard, we took a summarization approach, in which we created a one or two line summary of the counterargument (CA) considering its main points and finding the logic in the initial argument (IA) that the CA attacks. If the CA attacks the IA's conclusion instead of attacking any logic behind the conclusion, then we created a one line summary of the IA's main points. After that, we chose suitable text spans, relations and attributes to represent that one or two line summary of the CA, the attacked logic or summary of the IA and how the CA attacks the IA (i.e., which part of the IA logic is denied and if the CA agrees with any of the IA logic). We refer to these representations as *CA-pattern*, *IA-pattern* and *attack-pattern*.

We set constraints on how many relations and attributes can be selected for each of these three representations. For IA-pattern, a maximun of two and for CA-pattern, a maximum of three relations or attributes can be selected. Using a base causal relation is mandatory for the IA-pattern and using *good* or *bad* attributes is prioritized in both cases. Choosing at least one attacking relation (i.e., nullify or limit) is mandatory to represent the attack-pattern. Note that the attack-pattern represents the relation between the CA-pattern and IA-pattern or CA-pattern and the conclusion of the IA.

We also set a constraint on how long the text spans should be. We specified that although text spans can be long up to two small sentences or one compound sentence, we should attempt to choose smaller text spans (e.g., short phrases) as much as possible. We also specified that we should choose such text spans that when we read the patterns as a standalone logic (i.e., without reading the debates), they are understandable.

4.4 Annotation Study

The key requirements for identifying the logic pattern of attacks in arguments are two-fold: (i) identify as many logic pattern of attacks as possible and (ii) make human annotation feasible. To verify whether our LPAttack scheme satisfies these requirements,

	Homework should be abolished
PM-1	Abolishing homework will give students more free time
PM-2	Forcing students to do homework will make them passive in character
PM-3	It is not good for students to be obliged to study by their teachers or parents
PM-4	Students have memorized the incorrect way to study with homework
PM-5	Schools should take the responsibility for children's academic skills, not parents at home

	Death penalty should be abolished
PM-1	Death penalty is inhumane punishment
PM-2	Abolishing death penalty will prevent the situation of ending the life of innocent people
PM-3	Because of the high stress on the executioner, death penalty should be abolished
PM-4	The death penalty deprives criminals of the opportunity for rehabilitation
PM-5	Society is being brutalized by the death penalty

Table 4.3: Main points of the initial arguments of the debates in the TYPIC corpus for which counterarguments are written

we observed two metrics: (i) coverage of the scheme and (ii) inter-annotator agreement (IAA).

4.4.1 Source data

For both of our preliminary and annotation study, we utilized the debates from the TYPIC dataset (Naito et al., 2022). This dataset has 1,000 parliamentary style debates where given a topic, two opposing teams, Prime Minister (PM) and the Leader of the Opposition (LO) argue by taking a position in favor and against the topic respectively. In each debate, the PM speech acts as the *initial argument* and the LO speech acts as the *counterargument*. The corpus comprises 10 PM speeches on two topics: *"Homework should be abolished"* and *"Death penalty should be abolished"*. Table 4.3 shows the main points of these PM speeches. For each PM speech, there are 100 LO speeches. The arguments of 8 PM speeches out of 10 are causal arguments (underlined in the table).

Since our scheme is designed around two causal relations "promote" and "suppress", when we chose debates, we only chose these 8 PM speeches (initial arguments) for annotations whose arguments are causal and then, we randomly chose the LO speeches (counterarguments) associated with these PM speeches.

4.4.2 Setup

Two expert annotators participated in the annotation study and annotated the logic pattern of attacks independently using our annotation scheme[3].

We trained the annotators in a pilot annotation phase in which they were asked to annotate 20 debates. After the pilot annotation, we discussed the disagreements and, if needed, adjourned the annotation guidelines. One issue that we observed in the pilot annotations is that when we read the annotated patterns as a standalone logic (without reading the debates), some of them did not make complete sense because the chosen text spans had information gaps. To reinforce that the logic patterns are understandable on their own, we asked the annotators to write the text form of the logic patterns during our main annotation. For example, the text form of the logic pattern in Fig. 4.1 is as follows:

> **IA:** {*"death penalty" is negative*} because {*"death penalty" suppress ("chance of rehabilitation of the criminals" which is good)*}
>
> **CA:** {*"death penalty" is not negative*} because {*"death penalty" is more important/severe/has greater weight than "chance of rehabilitation of the criminals which is good" given the rationale/condition that "while executing prisoners is completely effective in ensuring...."*}

We expected that writing the text form would serve as a second check for the logic patterns and when the annotators read it separately from the debates, they will understand if there is an information gap or if the logic pattern is self-sufficient.

In our main annotation study, 50 debates were annotated by two annotators and 145 debates were annotated by a single annotator. For coverage and IAA, we report the results of dual annotations for 50 debates.

4.4.3 Rules for calculating IAA

One factor to consider during the IAA calculation is that in our scheme, we kept the flexibility of human representation i.e., the same interpretation can be represented in a slightly different way. For example, "no homework promote free time" has the same meaning as "homework suppress free time", but they are different representations. To handle such different representations that generally have the same meaning, we created

[3]We use diagrams.net for annotation.

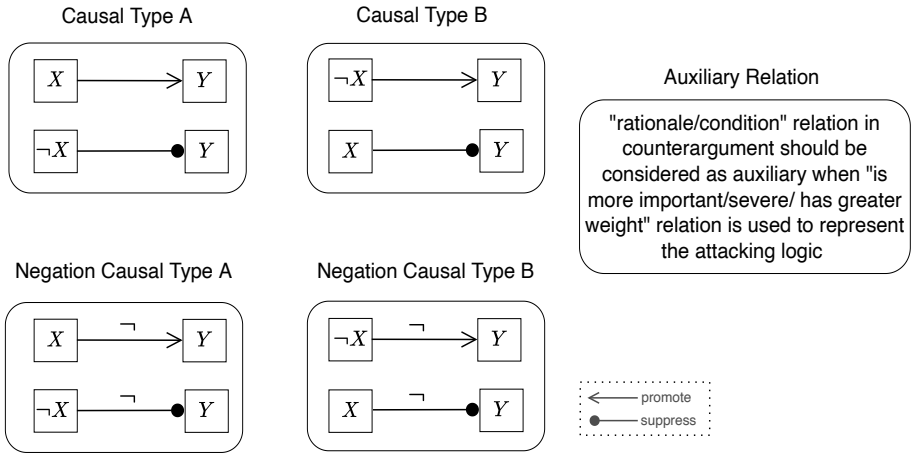

Figure 4.3: Rules for calculating inter-annotator agreement (IAA)

some rules to consider these different representation as the same. Figure 4.3 shows these rules. As shown in the figure, we considered the representations "no X promote Y" and "X suppress Y" as the same (marked as *Causal Type A*) since they have the same meaning. One of our rules consider rationale/condition relation as an auxiliary in certain cases where having or not having it does not affect the understanding of the logic. For example, in the case of the logic, {{no homework promote people fail in exam *given the rationale/condition that not doing homework will lead to lack of preparation*} which is more important or severe or has greater weight than {no homework promote free time}}, even if we remove the rationale/condition relation, the interpretation is understandable. We ignored this relation in such cases during the calculation of IAA.

4.4.4 Coverage

We asked the annotators to mark an attacking strategy as "Not Applicable (NA)" if our scheme cannot represent it. We obtained 90% (45/50) coverage for the LPAttack scheme. This result validates our hypothesis that the logic pattern of attacks in arguments is not uniformly distributed but rather is highly skewed i.e., logic pattern of a wide range of attacks can be captured with a limited set of relations and attributes.

4.4.5 Inter-annotator agreement (IAA)

We measured the IAA for relations and attributes[4] using Cohen's (κ) (Cohen, 1960). For the calculation of the IAA, we consider IA-pattern, CA-pattern and attack-pattern (described in §4.3.2) as markables. Since we want to know how much the annotators

[4]We ignore calculating the agreement for the selection of base pattern since all the debates used in the annotation study has the same base pattern (i.e., base pattern 1).

agree on each of these logics and the overall debate, we applied two strategies for calculating IAA: (i) calculate IAA considering each markable and (ii) concatenate the three markables to have a single representation of the whole debate and calculate IAA.

We obtained Cohen's κ of 0.63 in case (i), which indicates a substantial agreement and in case (ii), we obtained a κ of 0.49, indicating moderate agreement (Artstein and Poesio, 2008; Spooren and Degand, 2010).

We also examined whether the text spans were the same[5] in cases where relations and attributes were agreed. Among the three markables, only IA-pattern and CA-pattern have text spans, and therefore we considered these two markables for the matching calculation but followed the same strategy as above (i.e., (i) and (ii)). In each of the markables, if all of the text spans matched exactly, we called it *exact-match*, if all of the text spans shared at least one word, we called it *lenient-match* (including the case where some of them have lenient matching while others have exact matching). We saw that in (i), 68% (47/69) of text spans were similar (43% (30/69) exact-match, 25% (17/69) lenient-match). For (ii), we obtained a 46% (12/26) match (19% (5/26) exact-match, 26% (7/26) lenient-match).

4.4.6 Analysis of Annotations

We performed a manual analysis of the annotations in order to examine the correctness of the logic patterns, disagreements between the annotators and common attacking strategies captured by these annotations.

4.4.6.1 Correctness of the logic patterns

We determined how many annotated logic patterns were correct i.e., the logic patterns capture the essence of the attacks and are understandable enough when read independently of the debates. We adopted the following strategy:

> Exact logic pattern match between annotators ⇛ mark as a correct logic pattern
>
> Non-exact match between annotators ⇛ manually check the logic patterns and discuss with the annotators ⇛ mark as correct or incorrect based on the results of the discussion

[5]While we are aware of the work of Zeyrek et al. (2013), we did not apply any Kappa statistic on the text spans since the chance agreement is expected to be rare in our task.

Initial Argument (IA):
.....the **[death penalty]$_{(X)}$** should be abolished.... We are going to abolish the **[death penalty]$_{(X)}$** all over the world and introduce a **[life-imprisonment system]$_{(Y)}$**..... Our claim is that the death penalty deprives the chance of **[rehabilitation of the criminals]$_{(Y)}$**..... The criminal has no chance to reflect on what they have done. Life imprisonment system is a very severe punishment. Criminals are restricted of their freedom all day. They need to continue to apologize for the rest of their life while thinking about their victims....

Counterargument (CA):
They said that we should sentence prisoners to life instead of death so they can be rehabilitated. However, the reality is that **[a life spent in prison creates individuals who are unable to function when returned to society]$_{(A)}$**. These prisoners have become institutionalized, meaning they have become dependent on the rigid structure of the prison system. **[By being imprisoned for years, often in isolation and deprived of meaningful stimuli, these inmates gradually lose their life skills and ability to interact with others]$_{(A)}$**..... The death penalty, makes no claims to accomplish rehabilitation. Its goal is more straightforward: to deter future crime.

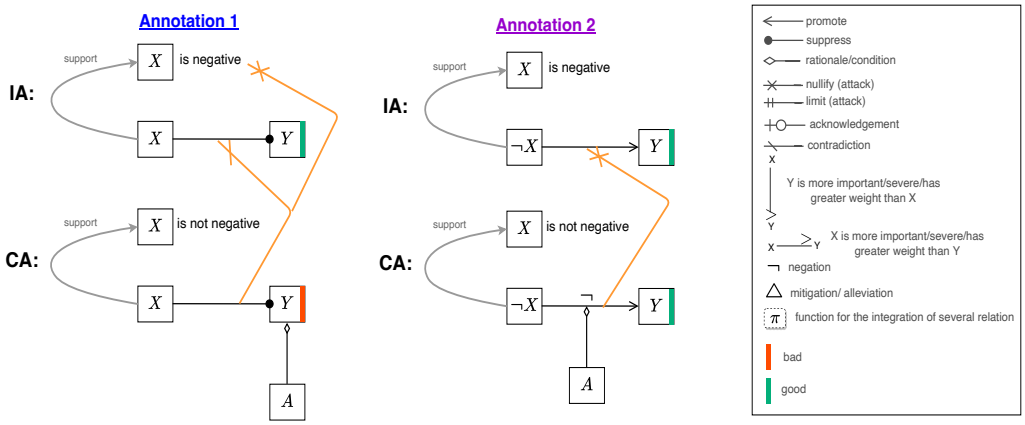

Figure 4.4: Example of debate where two annotators have different interpretation.

Following the above strategy, we found that 90% (45/50) of annotations of one annotator were correct whereas 86% (43/50) of annotations by the other annotator were correct. Both annotators had incorrect patterns for two of the debates. For four of the debates, one annotator chose "NA", whereas the other had correct patterns. For one of the debates, one annotator chose "NA", whereas the other had an incorrect pattern. This result indicates that having at least two annotations for a single debate provides a substantial likelihood of obtaining a correct annotation from one of the annotators.

4.4.6.2 Disagreements between annotators

There are generally two types of disagreements between the annotators: (i) the same interpretation of the debate but different logic patterns and (ii) different interpretations of the debate. One example of case (i) is given below:

> **Annotation 1:** {*"homework" is more important than "free time" given the condition/rationale that "homework can establish basic foundation of studying"*}
>
> **Annotation 2:** {{*"homework" promote "establish basic foundation of studying"*} *which is more important/severe/has greater weight than* {*"homework" suppress "free time"*}}.

	Common	Overlapping	Different
Interpretation	33	8	9
Logic pattern	27	15	8
Text span	21	18	11

Table 4.4: Detailed statistics of disagrement in interpretations, logic patterns and text spans. Each cell indicates the number of speeches whose annotations given by two annotators are common, overlapping, or different (see the text for the definition).

In this example, both annotations have the same interpretation i.e., *homework is more important than free time because it establish basic foundation of studying* but the interpretations are represented differently.

One important factor that we noticed is that in all of the cases of (ii), where annotators had a different interpretation of the debates, one of the annotations was found incorrect. One example of such case is shown in Fig. 4.4. In this example, the interpretation of CA-pattern is different in two annotations i.e., {*death penalty suppress life imprisonment system which is a bad thing*} and {*no death penalty doesn't promote rehabilitation of the criminals*}. Although both of the annotated patterns are understandable without reading the debates, *Annotation 1* has been marked as incorrect because in this debate, CA does not exactly express that life imprisonment is bad, but rather expresses that the reason behind abolishing the death penalty is the rehabilitation of the criminals whereas even if we abolish the death penalty, it does not result in rehabilitation in life imprisonment, and *Annotation 1* has failed to capture that notion.

We also see that many disagreements happen in the choice of text spans. When we manually checked the debates, we noticed that even when some interpretations are quite the same such as in case (i), the text spans are different. This is because sometimes, two or more sentences express the same meaning and annotators choose text spans from these different sentences. Consider the following example:

> **Annotation 1:** {*"homework" promote "learns that the way to succeed is by making schedule"*} *is more important/severe/has greater weight than* {*"homework" suppress "do more of what we really wanted"*}
>
> **Annotation 2:** {*"homework" promote "learns the importance of scheduling"*} *is more important/severe/has greater weight than* {*"homework" suppress "free time"*}.

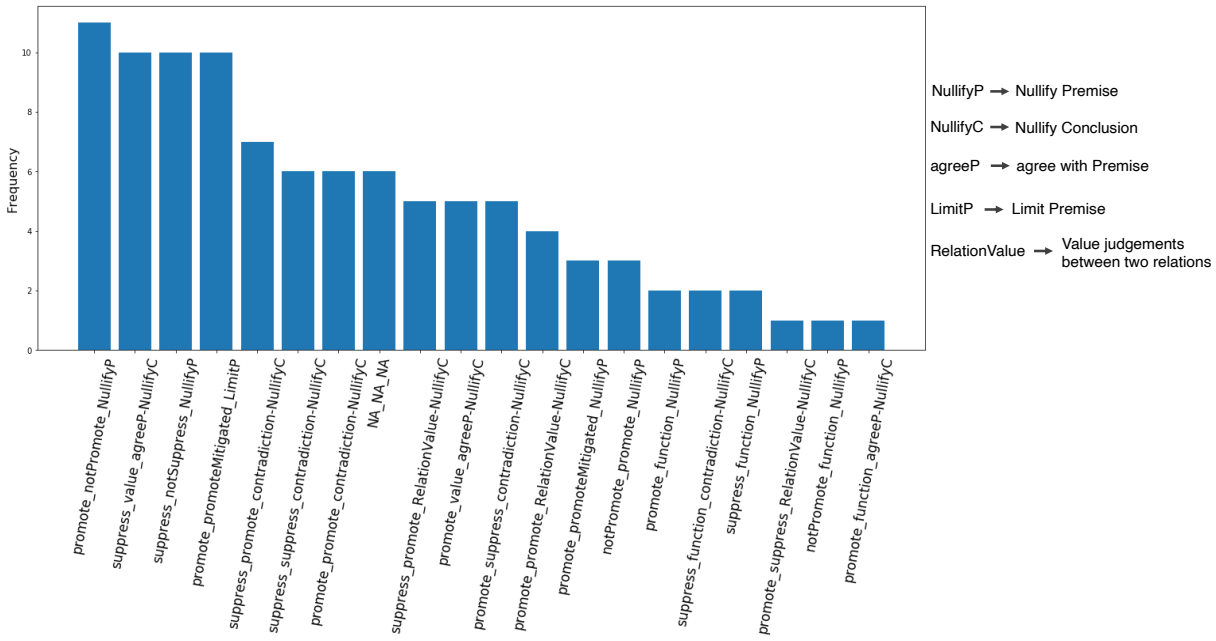

Figure 4.5: Distribution of logic patterns

In both cases, the text spans *"learns that the way to succeed is by making schedule"* and *"learns the importance of scheduling"* basically have the same interpretation but were chosen from different sentences and therefore are considered mismatched.

Besides, when the CA attacks the IA's conclusion, annotators often choose different main points to represent the premise of the IA and, in such cases, text spans do not match. In the above example we see that the representation of the main points of the IA is different i.e., *'do more of what we really wanted'* and *'free time'*, since the CA does not attack these premises.

Table 4.4 displays the number of speeches whose interpretations, logic patterns, and text spans selected by the two annotators are *common*[6] (i.e., two annotations are the same[7] in both the initial argument and counterargument), *overlapping* (i.e., two annotations are the same only in the initial argument), or *different* (i.e., otherwise).

4.4.6.3 Common attacking strategies captured by the annotations

To examine what sort of common rhetorical moves, assumptions, or value judgments in attacks are captured by our annotations, we looked into the distribution of relations

[6]For *interpretation*, we consider if the attacking point in the initial argument has same interpretation, i.e., if the *conclusion* of the initial argument is attacked in both annotations, we consider them *common* and ignore the premise interpretation of the initial argument.

[7]For text spans, we consider them *same* if they are *exact* or *lenient* match.

Initial Argument (IA):
Hello, everyone. Today's topic is "Death penalty should be abolished". We define that the [**death penalty**](X) should be abolished and instead of the death penalty, we propose that the suspected are sentence to life in. We have two points. The first point is [**"Executioner's suffering"**](Y). The second point is "Cruelty of death penalty". I will explain the first point. In present situations, a person who executes the death penalty for a criminal whose death penalty has been confirmed by a trial suffers a lot. Some methods of the death penalty include hanging and using gas chambers. Let me illustrate the case of hanging in Japan. A prisoner does not know when they will be executed until the day of execution. At the day of execution, they first enter the teacher's room and write a farewell letter. Then, they go to the antechamber for execution and are separated from the execution room by a curtain. The convict on death row is blindfolded and handcuffed, and a curtain is closed to the execution room. Finally, they go to the execution room. A rope is hung around their neck and they stand on a tread plate marked in the center of the room. Then, multiple prison officers push the button to open and close the tread, and the convict on death row falls. Those executioners feel strong stress. They don't know which button is actually connected to the input of the tread. They feel that they are responsible themselves for killing the suspect on death row by their own hands. Executers' stress is extremely overwhelming. That's why the death penalty should be abolished. Thank you.

Counterargument (CA):
They said that prison workers who take part in executions suffer stress, so the death penalty should be eliminated. However, instead of abolishing a punishment that the Constitution endorses, we should find ways to effectively deal with executioner stress. Obviously, if the job involves carrying out executions on a daily basis, without relief or counseling, the executioner is going to feel bad and probably exhibit PTSD. We can combat that by [**making sure would-be executioners are fully prepared for the job, that they are mentally sound and have a good support system**](A). In addition, we can relieve the stress of dealing with executions day in and day out by rotating the task so the number of executions carried out by a single guard is limited. In this way, executioner stress is reduced and the ultimate penalty can remain a legal option for the worst crimes.

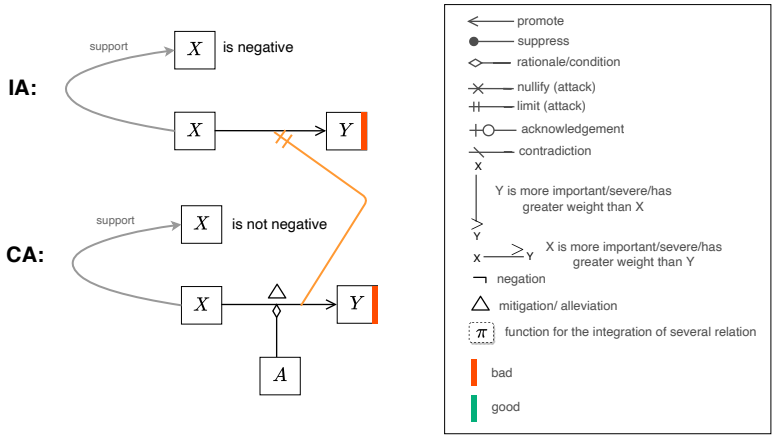

Figure 4.6: Annotation example of logic pattern of attack of a debate

and attributes used to annotate the debates. Figure 4.5 shows this distribution. We see that, the most common logic patterns are "attacking a premise by negating it", "value judgements between two concepts of a premise that leads to agreeing with the premise but denying the conclusion", "providing a way for mitigating the consequence of a premise that leads to agreeing with the premise and nullifying it at the same time" and "providing a contradictory premise that leads to denying the conclusion". Moreover, we observe that "value judgement between two causal relations" also happens quite often.

4.4.6.4 Annotation examples of the logic pattern of attacks

To provide a better understanding of what the annotated logic patterns look like and what sort of text spans are chosen from the given arguments, we provide annotation examples in Fig. 4.6, 4.7, and 4.8.

Initial Argument (IA):
Hello everyone. Today's topic is "**[Homework]**$_{(X)}$ should be abolished". We have two points: The first point is "free time" and the second point is "decrease burden on teachers". I will explain the first point of **["free time"]**$_{(Y)}$. We believe that if homework were to be abolished, we could have more free time. As a result, we could do more of what we really wanted like club activities, hobbies, or playing with friends. In my case, I go to tennis club after class until 5:00 pm and then I go to cram school until 8:00 pm. After this full day, I arrive at my home around 8:40 pm to eat dinner and take a shower. At nearly 10:00 pm I start my homework. I have a lot of homework. As a result, I go to bed late at night at nearly 1:00 am in the morning and I don't have the opportunity to sleep for a long period of time. It is not healthy. Therefore, homework should be abolished. Thank you.

Counterargument (CA):
They said that if we don't have homework, we have more free time and more healthy day. And teachers' burden will be decreased. However, a number of people who don't study at all will increase. People are forgetful, so not doing homework leads to insufficient fixing of class contents of the day. Thus during a week immediately before a semester test people who don't do class reviews will be more busy and then, they will fail in the examination for lack of preparation. To decrease **[a number of people who repeat years]**$_{(A)}$, homework is necessary.

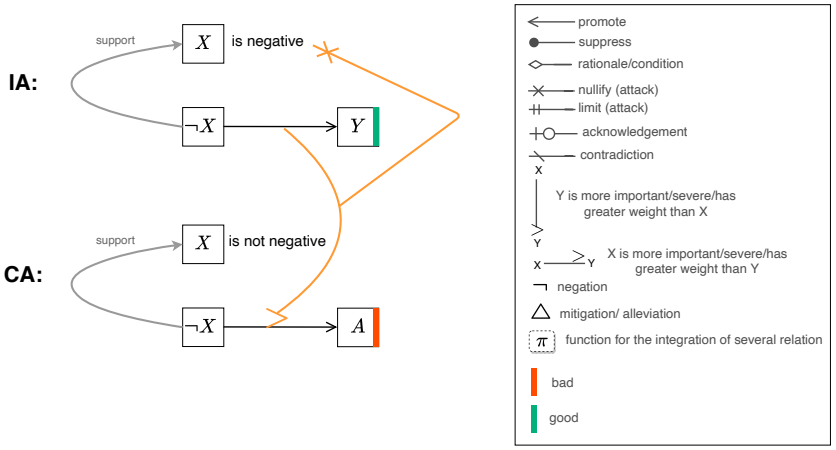

Figure 4.7: Annotation example of logic pattern of attack of a debate

4.5 Discussion and Future Work

In our annotation study, we observed that although the initial arguments were causal arguments, some logics in the arguments were evaluative judgments e.g., "death penalty is cruel" or "A truly just society can do without the death penalty" and the counterarguments focused on those logics. In such cases, the annotators failed to annotate the attacking strategy. In the future, we would like to enrich our scheme so that these sorts of logics in attack can be captured.

We also plan to have a second annotation for the 145 debates in our corpus that currently have only a single annotation. Besides, we plan to perform a voting between the two annotations to choose a single representation for each attacking strategy based on majority voting. In addition, we intend to apply the LPAttack annotation scheme on top of other existing debate corpora. Furthermore, we plan to formulate the task of automatic identification of logic pattern of attacks from given arguments and counterarguments.

We acknowledge the fact that capturing the logic pattern of attacks is a challenging task, especially when the arguments are long, and there is many room for improvements.

Initial Argument (IA):
Hello everyone. Today's topic is "**[Homework]**(X) should be abolished". We have two points: The first point is "free time" and the second point is "decrease burden on teachers". I will explain the first point of **["free time"]**(Y). We believe that if homework were to be abolished, we could have more free time. As a result, we could do more of what we really wanted like club activities, hobbies, or playing with friends. In my case, I go to tennis club after class until 5:00 pm and then I go to cram school until 8:00 pm. After this full day, I arrive at my home around 8:40 pm to eat dinner and take a shower. At nearly 10:00 pm I start my homework. I have a lot of homework. As a result, I go to bed late at night at nearly 1:00 am in the morning and I don't have the opportunity to sleep for a long period of time. It is not healthy. Therefore, homework should be abolished. Thank you.

Counterargument (CA):
They said that if homework were to be abolished, we can enjoy more free time. However, it's not true. Because instead of doing homework, we have to **[take time to catch up with classes]**(A). Please recognize purpose of homework. Homework exists to facilitate our efficient review and preparation for classes such as practice of using some formulas, or writing kanji. That's why even without homework, we have to study by ourselves anyway to understand classes. But problem is; we will take time to decide contents and review knowledge. Because we don't know what we should do. Given that, we can't have more free time on Gov side and homework rather allow we to study efficiently and have more free time.

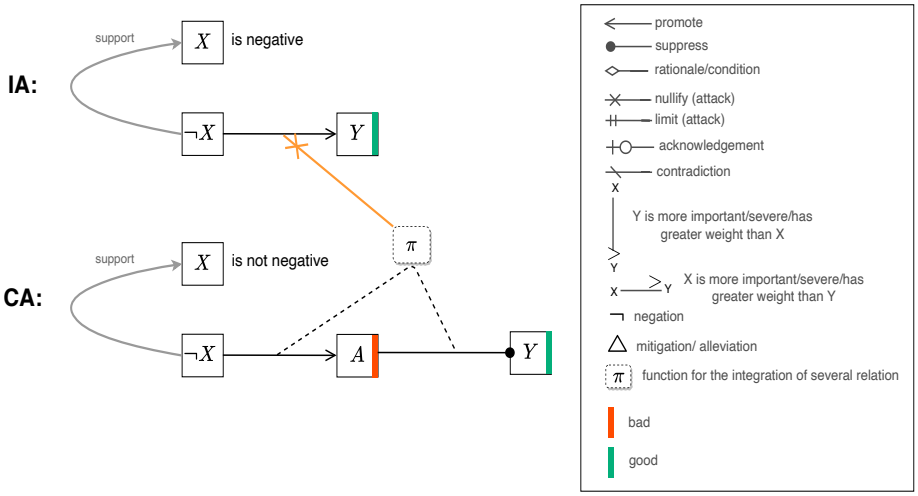

Figure 4.8: Annotation example of logic pattern of attack of a debate

4.6 Conclusion

We proposed LPAttack, a feasible annotation scheme for capturing the underlying logic pattern of attacks in arguments. LPAttack is designed to capture the common strategic moves, assumptions and value judgments during attacks in arguments. Our annotation study showed that even with a limited set of relations and attributes, we could capture the logic pattern of a wide range of attacks (90%) in a debate corpus of multiple, diverse debate themes. The results also showed a moderate inter-annotator agreement (Cohen's $\kappa = 0.49$) between two annotators, verifying the feasibility of the proposed scheme.

Chapter 5

Automatic Identification of Logic Patterns in Argumentation

5.1 Introduction

People use arguments in everyday life either to persuade others to adopt a position or belief, or to prevent others from adopting a certain position or belief (Walton et al., 2010). In argumentative discourse, persuasion is often achieved by refuting or attacking others' arguments.

Attacking an argument is not always straightforward and often consists of complex rhetorical moves in which arguers may agree with a logic of an argument while attacking another logic. In addition to such complexities in *Attacks*, arguments generally consist implicit sentiments, assumptions or value judgements which also contribute to the logical structure of attacks in arguments. Consider the following example of a debate that consists two argumentative speeches, conveyed by each opposing team of the debate:

(1) **Initial Argument (IA)**
Death penalty should be abolished (Conclusion)
because it deprives the chance of rehabilitation of the criminals. Criminals have no chance to reflect on their wrong-doing. (Premise)

Counterargument (CA)
Rehabilitation fails in comparison with the death penalty. While death penalty

ensures criminals never offend again, rehabilitation can't give that guarantee.

In Example 1, CA does not deny the *premise* of IA, instead she implicitly agrees with it while she denies the *conclusion* of the IA by giving more importance or value to the "death penalty" than the "rehabilitation of the criminals". Although this value judgement is implicit in the CA speech, CA explicitly provides a reason behind her value judgement (underlined text in CA).

Automatically identifying such reasoning patterns of complex rhetorical moves in attacks can help a wide range of natural language processing (NLP) applications, such as generating attacks in decision support or debating systems where a human and machine are engaged in a debate, producing logic-based abstractive summary, generating counterarguments by finding counterevidence to statements which can help diagnose learners' arguments and provide feedback to the learners in educational domain. Besides, recognizing such underlying logic patterns would lead to better understanding of arguments and their relations which would help us build more interpretable machine learning systems for argument mining tasks.

In spite of the broad benefits of automatic identification of logic patterns of attacks in arguments, less attention has been paid to this problem. Most of the existing studies in NLP that address *attacks* in arguments mainly focus on the classification of argumentative relations (e.g., support, attack, neutral), identifying attackable points in arguments, or counterargument generation Stab and Gurevych (2014a); Deguchi and Yamaguchi (2019); Kobbe et al. (2019); Jo et al. (2021a); Walton et al. (2008); Jo et al. (2020); Wachsmuth et al. (2018); Hua et al. (2019); Reisert et al. (2019); Alshomary et al. (2021); Jo et al. (2021b).

Although some recent studies (Reisert et al., 2018; Jo et al., 2021a) developed annotation schemes and logical mechanisms to capture the logic behind support and attack relations where they exploited implicit causal links and sentiments, these studies did not capture other implicit information, e.g., presupposition or value judgments in arguments that also contribute to the underlying logical structure of attacks. Furthermore, none of these studies capture the modes of *attacks* (e.g, whether the counterargument denies the conclusion or the premise of the attacked argument) and the complex rhetorical moves (e.g., agreeing with a premise while attacking the conclusion) in them.

To address these gaps, in our previous study (discussed in Chapter 4) we have introduced LPAttack (**L**ogic **P**attern of **Attack**) annotation scheme that captures common modes and complex rhetorical moves in attacks along with the implicit information, pre-

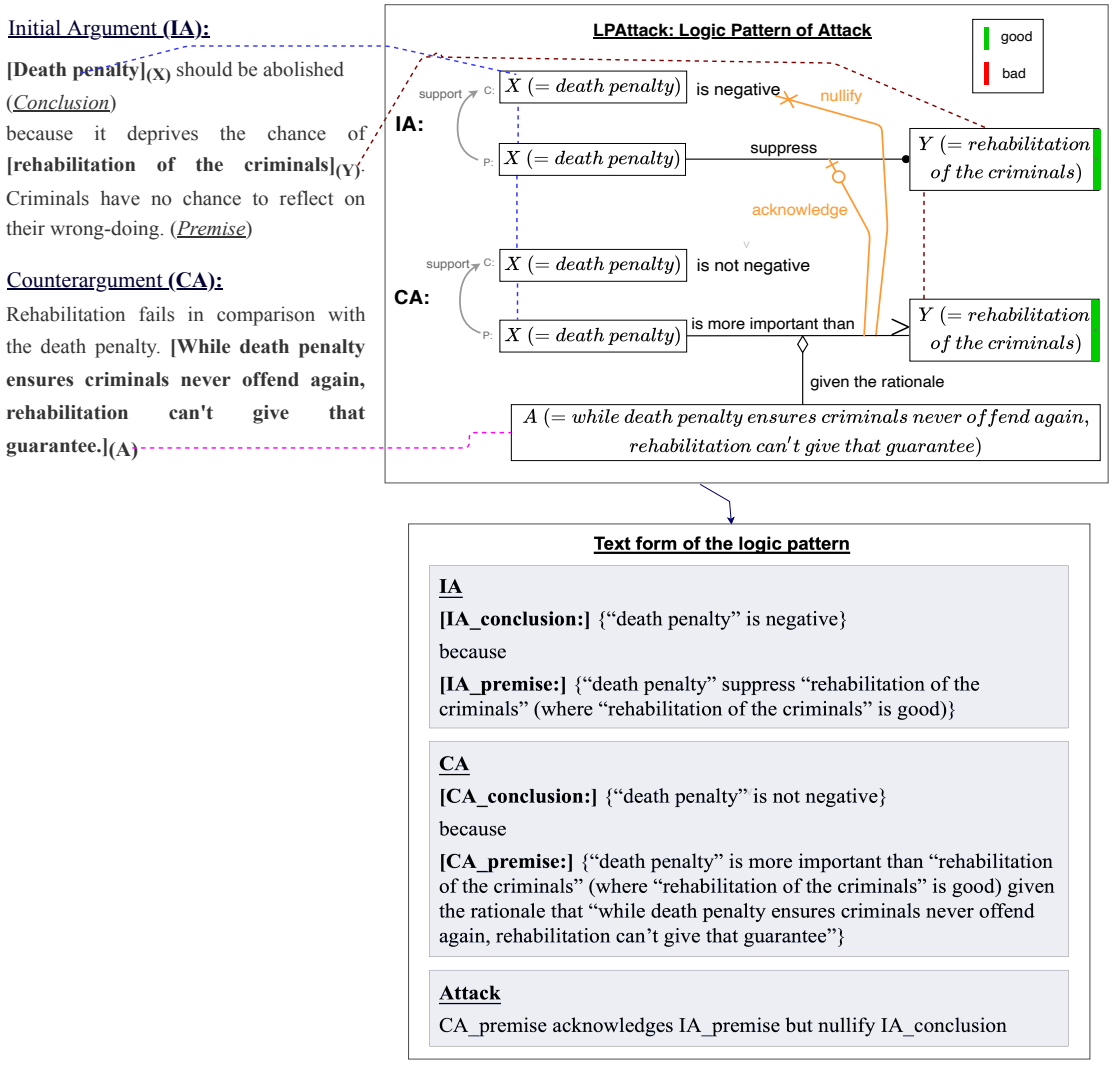

Figure 5.1: An example of logic pattern of attack of a debate captured by the LPAttack annotation scheme and the text form of the logic pattern.

suppositions, or value judgments (Mim et al., 2022). The conducted annotation study using the LPAttack scheme resulted in the construction of a corpus of logic patterns of attacks of 250 debates. Fig. 5.1 shows how the logics of Example 1 are represented using the LPAttack scheme and the text form of the logic pattern.

In this chapter, we formulate the task of automatic identification of logic pattern of attacks (captured by the LPAttack scheme) from given arguments and counterarguments. Most of the existing argument mining tasks including reasoning patterns identification use hand-crafted features such as auxiliary verbs (e.g. should, ought), part-of-speech tags, lemma, n-grams, punctuation marks, word overlap and sentiment agreement between two statements, discourse markers Feng and Hirst (2011); Rinott et al. (2015); Persing and Ng (2016a); Habernal and Gurevych (2017); Stab and Gurevych (2017);

Reisert et al. (2018). Recently many argument mining and related tasks (e.g., stance detection and classification, classification of support, attack, or neutral argumentative relations) used pre-trained deep language representation models (e.g., BERT, GPT2, BART, T5) (Devlin et al., 2018; Radford et al., 2019; Lewis et al., 2019; Raffel et al., 2020) and achieved state-of-the-art results (Durmus et al., 2019; Chakrabarty et al., 2020; Kobbe et al., 2020; Al Khatib et al., 2021; Saha et al., 2021).

The text form of the logic patterns (annotated by the LPAttack scheme) can be seen as an abstractive summary of the given argument and counterargument (as shown in Fig. 5.1). In this work, we treat the task of automatic identification of logic patterns of attacks as a logic pattern generation or summarization task and use a state-of-the-art language model which has been pre-trained on the abstractive summarization task. We demonstrate that the model yields moderate performance for the logic pattern generation task, setting a baseline for this challenging task.

5.2 Background

Computational analysis of argumentation has gained considerable attention in recent years because of its importance in many NLP applications such as essay scoring, argumentative writing support systems, and educational feedback. Common lines of work in this area include stance detection (Durmus et al., 2019; Xu et al., 2019; Allaway and McKeown, 2020), argumentative units (e.g., claim, premise) identification Levy et al. (2014); Rinott et al. (2015); Stab and Gurevych (2014a), argumentative relations (e.g., support, attack, neutral) classification Peldszus and Stede (2015); Cocarascu and Toni (2017); Niculae et al. (2017); Stab and Gurevych (2014a); Deguchi and Yamaguchi (2019); Kobbe et al. (2019); Jo et al. (2021a), qualitative assessment of arguments Persing et al. (2010); Persing and Ng (2013, 2014, 2015, 2016b); Rahimi et al. (2015); Wachsmuth et al. (2016); Habernal and Gurevych (2016); Wachsmuth et al. (2017); Mim et al. (2019b,a, 2021) and retrieval or generation of counterarguments Hua and Wang (2018); Wachsmuth et al. (2018); Hua et al. (2019); Reisert et al. (2019); Alshomary et al. (2021); Jo et al. (2021b).

Towards automatically identifying the reasoning patterns in argumentation, Feng and Hirst (2011) created a computational model using hand-crafted features (e.g., sentiment of a statement, if an argumentation is linked or convergent) to classify Walton's argumentation schemes (Walton et al., 2008) in the Araucaria (Reed, 2006) corpus where argumentative texts are annotated with Walton's schemes. Recently, Reisert et al. (2018)

created a computational model using handcrafted rule and features like auxiliary verbs (e.g. should, must, ought), negated auxiliary verbs (e.g. should not, must not), lemma, part-of-speech tags to identify and represent argument templates that capture underlying reasoning. In another recent work, Jo et al. (2021a) composed a set of rules (e.g., if there are contradictory facts in statement S_1 and S_2 , then there is an *attack* relation between S_1 and S_2) that specify the logical mechanisms in argumentation and used such logical mechanisms to signal support or attack relation.

Since the argument mining datasets typically differ with respect to their annotations depending on the task, designing features or rules manually for each new corpus becomes a challenge. Because of that reason, researchers started to use neural model architectures which do not require any manual feature engineering and achieved substantial improvement in argument mining tasks (Eger et al., 2017; Xu et al., 2018; Chen et al., 2018; Kobbe et al., 2020).

Recently, Transformer based (Vaswani et al., 2017) pre-trained deep language representation models (e.g., BERT, GPT2, BART, T5) (Devlin et al., 2018; Radford et al., 2019; Lewis et al., 2019; Raffel et al., 2020) achieved state-of-the-art results in various tasks of NLP, including argument mining tasks. Durmus et al. (2019) used a language model to determine the stance of a claim and which claim is more specific between two claims in a newly created dataset. They obtained substantial accuracy for their tasks. Chakrabarty et al. (2020) predicted the argumentative relations (i.e., support, attack or neutral) using a language model where they leveraged contextual information and discourse relations during fine-tuning. Their approach obtained significant performance gain compared to the existing state-of-the-art models. Jo et al. (2020) classified attackable sentences using language model and obtained substantial accuracy.

Language models have been extensively used for the generation tasks in argument mining as well. Gretz et al. (2020) employed language model to generate coherent claims where they added contextual information in the training data. Alshomary et al. (2021) ranked premises to find the weakest premise and generated counterargument for that premise using language model. Al Khatib et al. (2021) integrated causal knowledge from knowledge graphs using language model to generate arguments for a given prompt. Saha et al. (2021) used language model to predict of if an argument counters or supports a belief and then to generate a commonsense reasoning graph that provides explanation for the predicted stance.

Since the logic patterns annotated by the LPAttack scheme can be seen as an abstractive

summary of the argumentation, we treat the automatic identification of the logic patterns of attacks as a logic pattern generation or summarization task and use a language model that has been previously pre-trained on the task of generating abstractive summary.

5.3 Experimental Setup

We consider the automatic identification of logic patterns of attacks as a logic patterns generation or summarization task since the logic patterns annotated by the LPAttack scheme can be seen as an abstract summary of the given argument and counterargument.

5.3.1 Model

For all of our experiments, we use T5 (Text-to-Text-Transfer Transformer) (Raffel et al., 2020) which is a Transformer-based (Vaswani et al., 2017) encoder-decoder model. In this framework, the encoder is fed an input sequence and the decoder produces a new output sequence. T5 model follows a text-to-text approach i.e., the model is fed some text for context or conditioning which is concatenated with the input text and then the model is asked to produce the output text. These texts for context or conditioning are referred to as task-specific *"prefix"*. For example, to summarize some particular "text", the input will be *"summarize: text"*, where "summarize: " is the task prefix.

T5 uses a *causal masking* attention in its encoder for the input text with a *fully-visible masking* attention applied to the prefix of the input text. A "Causal" attention masking refers to the mechanism where to produce the ith entry of the output sequence, it prevents the model from attending to the jth entry of the input sequence for $j > i$. This mechanism is used during training so that the model can't "see into the future" as it produces its output. A "fully-visible" attention masking allows a self- attention mechanism to attend to any entry of the input when producing each entry of its output.

T5 model is pre-trained with denoising objective which is also known as masked language modeling objective (Devlin et al., 2018) where the task is to predict the masked tokens in the text. T5 has then been pre-trained on an abstrative summarization task where it produced state-of-the-art performance.

5.3.2 Data

We use the corpus created using the LPAttack annotation scheme (Mim et al., 2022) consisting logic pattern of attacks of 250 debates. In this corpus, there are 8 initial

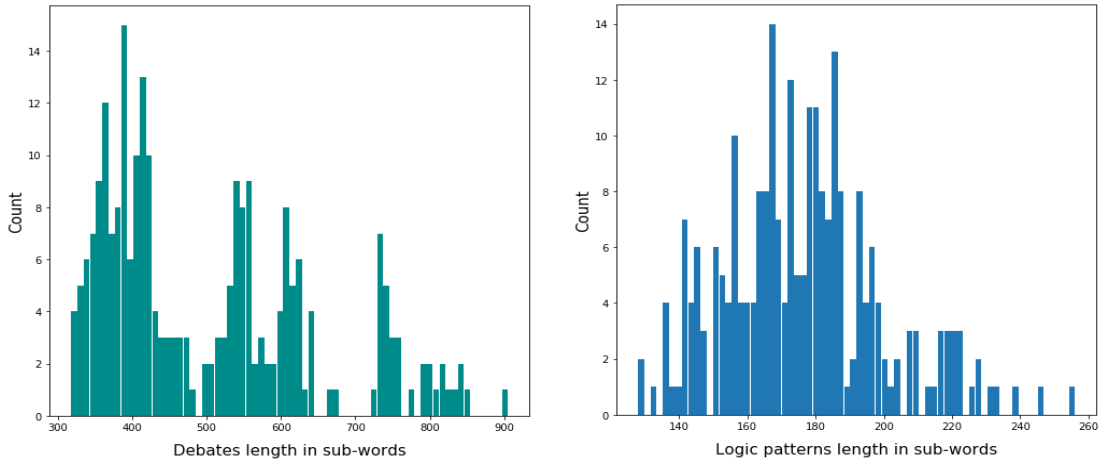

Figure 5.2: Histogram of lengths of Debates and Logic patterns annotated by the LPAttack scheme

arguments (IAs), 4 for the topic "homework" and 4 for the topic "death penalty", and 250 counterarguments (CAs) which correspond to these initial arguments. An initial argument and its corresponding counterargument refer to a debate. The average number of tokens per debate and logic pattern are 502 and 176 respectively. The longest debate have 905 tokens and the longest logic pattern have 256 tokens. The histograms of the length of the debates and logic patterns are shown in Fig 5.2.

Among the 250 debates, logic pattern of 5 debates have been annotated as "Not Applicable" or "Incorrect" by both of the annotators. Therefore, we use the 245 debates with the correct logic patterns for our experiments.

5.3.3 Task Setting

For the logic patterns generation task, the input is a debate (i.e., initial argument and a corresponding counterargument) and the output is the text form of the logic pattern of attack of that debate (as shown in Fig. 5.1).

We explore two settings for this task:

1. **In-domain setting:** In this setting, there is topic overlap between training and test data (debates from both of the topic "homework" and "death penalty" are in training as well as test data). This setting has two variations and both of the variations have same test dataset:

    - *IA-overlap:* In this setting, there is initial argument overlaps between training and test data.

- *No-IA-overlap:* There is no initial argument overlap between training and test data in this setting.

Since we remove the overlapping debates from the training data for *No-IA-overlap* setting, this setting has less training data compared to *IA-overlap* setting (16 debates less).

2. **Out-of-domain setting:** In this setting, there is no topic overlap between training and test data. It also has two variations:

   - *Train-on-HW:* In this setting, we train the model on the debates from the topic "homework" and test it on the debates from the topic "death penalty"

   - *Train-on-DP:* In this setting, we train the model on the debates from the topic "death penalty" and test it on the debates from the topic "homework"

   In the corpus, there are more debates under the topic "homework" than the topic "death penalty". Hence, the *Train-on-HW* setting has more training data (17 debates more) compared to the *Train-on-DP* setting.

5.3.4 Evaluation Procedure

We fine tune the pre-trained T5 model on our logic-pattern generation task. For all of the *In-domain* settings, we use 20% of the data as our test set. From the remaining data, we use 20% for validation and the rest as the training data. For all of the *Out-of-domain* settings, we use all the debates of a particular topic as our test set. Then, from the other topic, we use 20% data for validation and the rest as the training data.

We perform automatic evaluation for this task and use *ROUGE* (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) as our evaluation metrics. ROUGE metrics are generally used for evaluating automatically generated summaries. It compare the n-grams of the generated summary to the n-grams of the actual summary. The higher the ROUGE scores, the better the generated summary and ROUGE score 1.0 means that the generated summary is identical to the original summary.

ROUGE1 and ROUGE2 compare the uni-grams and bi-grams between the machine-generated summary and the human reference summary respectively. ROUGE-L doesn't compare n-grams, instead treats each summary as a sequence of words and then looks for the longest common subsequence ignoring sentence boundary. Please note that we

Setting	Pattern Type	ROUGE-1			ROUGE-2			ROUGE-L		
		average	max	min	average	max	min	average	max	min
IA-overlap	Overall	0.78	1.0	0.55	0.71	1.0	0.44	0.76	1.0	0.50
	IA	0.85	1.0	0.46	0.80	1.0	0.33	0.85	1.0	0.46
	CA	0.64	1.0	0.38	0.55	1.0	0.24	0.62	1.0	0.36
	Attack	0.86	1.0	0.55	0.78	1.0	0.46	0.81	1.0	0.45
NO-IA-overlap	Overall	0.76	0.98	0.52	0.64	0.95	0.37	0.74	0.98	0.50
	IA	0.81	1.0	0.48	0.72	1.0	0.33	0.81	1.0	0.48
	CA	0.61	0.97	0.37	0.50	0.94	0.23	0.59	0.97	0.34
	Attack	0.84	1.0	0.45	0.72	1.0	0.3	0.80	1.0	0.45

Table 5.1: Results of logic patterns generation for In-domain settings (topic overlap between training and test data).

do not use ROUGE-Lsum as our evaluation metric since it considers sentence boundaries in the generated text and there are no sentence boundaries in the logic patterns.

For *In-doamin* settings, we calculate ROUGE scores for initial argument (IA), counter-argument (CA),and Attack patterns separately for each of the generated logic pattern. We also average the scores of IA, CA and Attack patterns and report it as an *"Overall"* pattern score. For *Out-of-domain* settings, we do not calculate ROUGE scores for IA, CA, and Attack patterns separately but combinedly since the generated instances in Out-of-domain settings often have missing CA or Attack patterns.

5.3.5 Preprocessing

We lowercase the tokens of the debates and logic patterns except for the tokens *"IA", "CA", "Attack"* in both of them and the *"IA_conclusion", "IA_premise", "CA_conclusion", "CA_premise"* tokens in the logic patterns. We add the token *"summarize:"* to the beginning of all the debates since for simmarization with T5 model, it is needed to add such token to the beginning of the texts that is needed to be summarized.

5.3.6 Implementation Choices

We use *T5-base* model for our logic pattern generation task. Adam optimizer and batch sizes of 2 is used. The learning rate is set to $1e-4$ for fine-tuning the model. We use early stopping with patience 5, and train the network for 20 epochs. To select hyper-parameters, we monitor performance on the validation set and choose the model that yields the highest ROUGE-1 score. We specify the input sequence length (in tokens) to be 1024 and the output (summary) sequence length to be 512.

Setting	ROUGE-1			ROUGE-2			ROUGE-L		
	average	max	min	average	max	min	average	max	min
Train-on-HW	0.63	0.98	0.39	0.50	0.95	0.26	0.60	0.98	0.35
Train-on-DP	0.54	0.94	0.25	0.42	0.89	0.12	0.51	0.94	0.23

Table 5.2: Results of logic patterns generation for Out-of-domain settings (trained on one topic, tested on another topic).

5.4 Results

Table 5.1 and 5.2 show the performance of the logic patterns generation task. We report the average (average performance on all test debates), maximum (maximum performance among all test debates) and minimum (minimum performance among all test debates) performance.

5.4.1 Results of In-domain settings

Table 5.1 lists the results of the *In-domain* settings (topic overlap between training and test data). If we look at the "average" under all of the ROUGE scores, we see that moderate performance is obtained for this task ("Overall" pattern) in both of the *IA-overlap* and *NO-IA-overlap* settings.

We also see that the *IA-overlap* setting yields better performance than the *NO-IA-overlap* setting. This could be because of the fact that in the training phase of *IA-overlap* setting, the model is exposed to the initial arguments similar to those in the testing phase or the fact that *IA-overlap* setting has slightly more training data (16 debates more).

We also see that among *IA, CA*, and *Attack* pattern generation, poor performance is achieved for the *CA* pattern generation. Furthermore, we observe that in the *IA-overlap* setting, sometimes the generated logic pattern is identical to the human annotated logic pattern (max = 1.0).

5.4.2 Results of Out-of-domain settings

Table 5.2 show the results of the *Out-of-domain* settings (trained on one topic, tested on another topic). We see that to some extend, the model can learn some logic from one topic data and can apply it for a different topic .

We observe that between two Out-of-domain settings, *Train-on-HW* performs better.

Pattern of Identical Predictions

IA

[IA_conclusion:] {"X" is negative}

because

[IA_premise:] {"X" promotes “Y” (where “Y" is bad)}

CA

[CA_conclusion:] {"X" is not negative}

because

[CA_premise:] {{"X" doesn't promote “Y” (where “Y" is bad)} given rationale/condition “Z”}

Attack

CA_premise nullify IA_premise

Example of Identical Prediction (ROUGE scores = 1.0)

IA

[IA_conclusion:] {"death penalty" is negative}

because

[IA_premise:] {"death penalty" promotes "executioner's suffering" (where "executioner's suffering" is bad)}

CA

[CA_conclusion:] {"death penalty" is not negative}

because

[CA_premise:] {{"death penalty" doesn't promote "executioner's suffering" (where "executioner's suffering" is bad)} given rationale/condition "the executioner can feel peace of mind, knowing that he has served a part in bringing justice to the victim and the victim's family, by seeing to it that the guilty party can never hurt anyone again”}

Attack

LO_premise nullify PM_premise

Figure 5.3: Pattern of predictions identical to the human annotation and an example of identical prediction.

That could be because of the reason that *Train-on-HW* has more data for training (17 debates more) or *Train-on-HW* data have more generalizable logics.

5.5 Analysis

For analyzing the results, we select the In-domain setting *IA-overlap*, since it is a commonly used setting and has the best performance. Please note that, logic patterns are

	ROUGE-1			ROUGE-2			ROUGE-L		
	average	max	min	average	max	min	average	max	min
IA	0.91	1.0	0.5	0.82	1.0	0.0	0.91	1.0	0.5
CA	0.80	0.97	0.54	0.71	0.94	0.12	0.79	0.97	0.52
Overall	0.85	0.99	0.52	0.76	0.97	0.06	0.85	0.99	0.51

Table 5.3: Results of text spans match between generated and human annotated logic patterns

made of *"relations and attributes"* as well as *"text spans"* chosen from the given arguments and counterarguments. We refer relations and attributes to *"pattern"* here for the purpose of analysis.

5.5.1 Identical predictions

To explore what sort of logics the model is able to predict successfully, we look into the predictions identical to the human annotation. We find three such identical predictions and observe that all the identical predictions have the same pattern (when the text spans of the patterns are replaced with variables). The pattern of identical predictions and one such identical prediction are as shown in Fig 5.3.

In order to investigate what enabled the model to generate such identical predictions, we looked into the training data and found 20 identical patterns in the training data. We think that being able to be trained on a number of identical patterns helped the model to learn the logics of this pattern and generate such pattern correctly along with the correct text spans.

5.5.2 Pattern and Text span matching

We investigate if there are some cases where the model is able to predict the relations and attributes correctly but the text spans are not correct. We identify how many generated patterns as well as text spans match to the human annotation separately (we exclude identical predictions during this investigation).

For pattern matching, we replace all the text spans with a fixed variable and calculate only pattern match. We find that 25% (11/44) of the generated patterns (IA+CA+Attack pattern) match exactly to the human annotation.

We also find that for all the IA patterns, the IA_conclusions are predicted correctly

Human annotated logic pattern

IA

[IA_conclusion:] {"homework" is negative}

because

[IA_premise:] {no "homework" suppresses "be obliged to study by their teachers or parents" (where "be obliged to study by their teachers or parents" is bad)}

CA

[CA_conclusion:] {"homework" is not negative}

because

[CA_premise:] {{no "homework" doesn't suppress "be obliged to study by their teachers or parents" (where "be obliged to study by their teachers or parents" is bad)} given rationale/condition "some students will still be forced to study, because thought of their teachers or parents, hope to put their children into advanced schools"}

Attack

CA_premise nullify IA_premise

Generated logic pattern (ROUGE-2 = 0.68)

IA

[IA_conclusion:] {"homework" is negative}

because

[IA_premise:] {no "homework" suppresses "problems between family" (where "problems between family" is bad)}

CA

[CA_conclusion:] {"homework" is not negative}

because

[CA_premise:] {{no "homework" doesn't suppress "problems between family" (where "problems between family" is bad)} given rationale/condition "though thought of their teachers or parents, hope to put their children into advanced schools will not be changed on government paradigm"}

Attack

CA_premise nullify IA_premise

Figure 5.4: Generation example where the pattern (relations and attributes) match to the human annotation but text spans do not match.

and there is 61% (27/44) exact IA pattern match. For the rest of the (39%, 17/44) IA patterns, IA_premises are predicted incorrectly.

We then calculate how many CA pattern match but attack pattern do not match when IA pattern match exactly. We see that there is no such matching. It means that if the model can predict IA and CA pattern correctly, then it can also predict Attack pattern

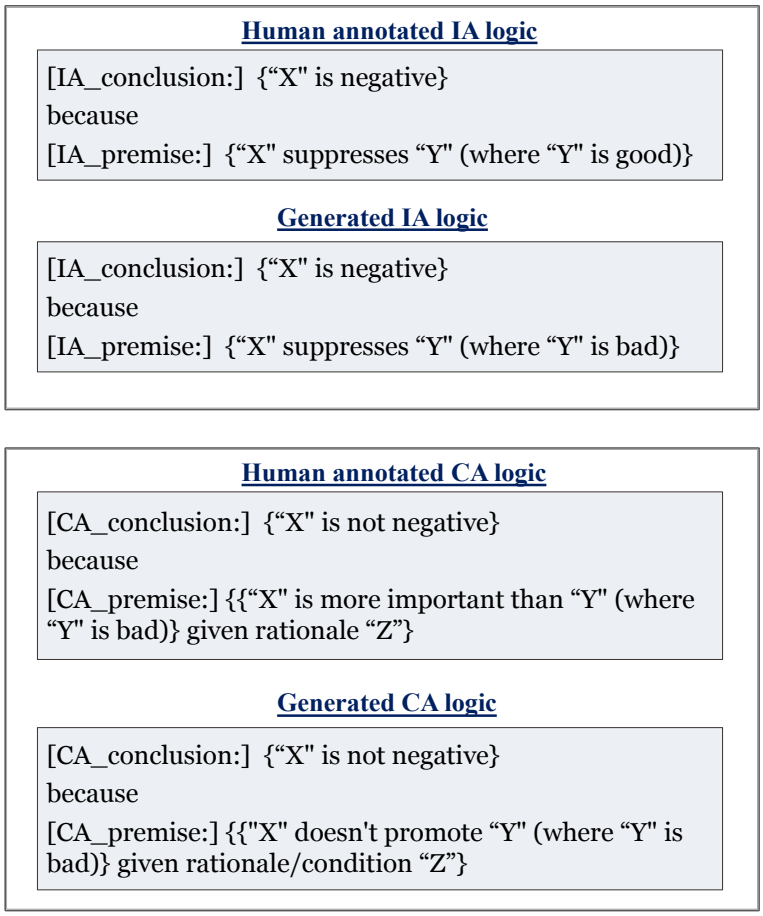

Figure 5.5: Incorrectly predicted IA and CA patterns

correctly. We find that there are 16 cases where the IA pattern match but CA patterns do not match.

We also calculate how much the text spans match when the patterns match exactly [1]. We observe that 82% (9/11) IA text spans match exactly and there is no exact matching for CA text spans. To investigate if the text spans generated are closer to the human annotation, we calculate the ROUGE scores for the text spans. Table 5.3 shows the results and we see that for CA, although there are no exact text span match, the model generates some good text spans close to the human annotation (good semi-exact matching, ROUGE-2 score = 0.94). One example of generated logic pattern where the pattern match with the human annotation but IA and CA text spans do not match is shown in Fig 5.4.

[1]Please note that only IA and CA patterns have text spans.

[CA_conclusion:] {“X" is not negative}
because
[CA_premise:] {{"X" doesn't promote “Y” (where “Y" is bad)} given rationale/condition “Z”}

[CA_conclusion:] {“X" is not negative}
because
[CA_premise:] {{“X" promotes “Y" (where “Y" is bad)} can be mitigated given rationale “Z”}

Figure 5.6: Correctly predicted CA patterns

5.5.3 Errors in pattern prediction

We investigate where the model fails to generate correct patterns and we see that while all the IA and CA *conclusions* are predicted correctly (which is basically sentiment prediction), the model struggle to predict the reasoning *premises*. Examples of reasoning error in IA and CA are shown in Fig 5.5. As we see in the Figure, while generating IA patterns, the model could not learn that *if “X” suppresses something bad, then it becomes a good or positive thing*.

For the mismatched CA patterns shown in Fig 5.5, their IA pattern match and the IA pattern is *[IA_conclusion:] {“X” is negative} because [IA_premise:] {“X” promotes “Y” (where “Y” is bad)}*. We find that model could not predict any complex CA patterns that represent value judgement or contradiction. Only two CA patterns have been predicted correctly which are not as complex as value judgements (shown in Fig 5.6).

5.6 Conclusion and Future Work

In this work, we consider the task of automatic identification of logic patterns as a logic pattern generation or summarization task and use a pre-trained language model for generating the logic patterns. Our results from the automated evaluation show that the model yields moderate performance, setting a baseline for this task. Further analysis of the results exhibits that the model struggles to generate reasoning patterns, providing future direction for designing a more sophisticated model to generate proper reasoning for the logic patterns.

Chapter 6

Conclusion

Argumentation can help students improve their critical thinking skills, decision making skills or writing skills. However, students often struggle to construct well-defined arguments and it is necessary to guide them by providing feedback so that they can improve their argumentation. The difficulty here is that providing feedback manually is an extremely time consuming task and requires lots of human efforts. Therefore, the importance of building an automated feedback system is enormous and assessing the quality of argumentation and capturing its underlying reasoning patterns are two of the crucial tasks to reach this ultimate goal.

For the precise assessment of argumentation quality, incorporating its discourse information is critical. Existing studies use discourse annotations based parsers or pe-trained language models to encode such information but discourse annotations are costly and long-range discourse dependencies are not well captured by language models. In addition, while quality assessment of argumentation enables us to provide feedback about how good or bad an argumentation is, it does not indicate the issues why the quality is good or bad. For such deeper understanding, capturing the underlying reasoning patterns of argumentation is necessary but it is relatively less explored and no existing studies capture complex strategic moves in argumentation.

Given this background, in this thesis, we explored the following research issues:

How to capture discourse structure in argumentation in an unsupervised way? From a series of investigations and experiments, we found that we can capture long-range (i.e., paragraph level) discourse dependencies in an unsupervised way by cor-

rupting argumentative texts (e.g., shuffling or dropping the some paragraphs) automatically and training a model to learn to distinguish between the original and corrupted argumentative text. We also found that capturing discourse in this way improves the argumentation assessment performance. We hope that these findings will facilitate discussions on unsupervised ways of capturing discourse structure in argumentation.

What are the common reasoning patterns in argumentation? We conducted a preliminary study where we identified the common reasoning patterns (e.g., agreeing with and denying a premise at the same time which leads to denying the conclusion) of complex strategic moves in argumentation.

How to capture the common reasoning patterns in argumentation? Based on the insights of preliminary study, we created an annotation scheme comprising two base patterns and fourteen relations and attributes (e.g., promote, acknowledge, nullify) which can capture the common underlying logic patterns in argumentation.

Is it possible to automatically identify the reasoning patterns in argumentation? We conducted baseline model experiments and found that if we have annotated data, it is possible to automatically identify the logic patterns in argumentation but existing models struggles to predict the reasonings in the logic patterns. We hope that these findings will facilitate future research on building sophisticated models for the task of automatic identification of reasoning patterns.

The key contributions of this thesis are summarized as follows:

Establishing an unsupervised approach to capture long-range discourse dependencies in argumentation: We proposed a novel unsupervised pre-training approach to capture long-range discourse dependencies in argumentation that does not require any discourse parsers or annotations. We then used our unsupervised pre-training method for the quality assessment of argumentation. We demonstrated that our method is effective in capturing discourse structure of argumentation by achieving state-of-the-art performance on the assessment task.

Designing an annotation scheme to capture the reasoning patterns in argumentation: We analyzed the internal structure of how one argument attacks or agrees with another argument which provided insights into how to represent the strategic moves in argumentation so that human annotation is plausible. Based on these insights, we designed a novel annotation scheme, defined the annotation guidelines and formulated the task of capturing logic pattern of attacks in argumentation.

Construction of a corpus using the invented annotation scheme: We conducted an annotation study and created a corpus comprising logic pattern of attacks using our proposed scheme. Our annotation study yielded moderate agreement between two annotators indicating the feasibility of the human annotation for the scheme.

Baseline model experiments for the automatic identification of reasoning patterns: We considered the automatic identification of reasoning patterns as a reasoning patterns generation task and used a pre-trained language model for the generation purpose. The model achieved moderate performance, setting a baseline for this task.

To conclude, first of all, this thesis demonstrated that it is possible to capture discourse dependencies in argumentation in an unsupervised way. However, the thesis in hand focuses only on the argumentation from educational domain. Therefore, the potential future work would be investigating how such unsupervised strategy performs in other domains and if there is any difficulty in adopting this strategy in other domains.

This thesis also showed that capturing reasoning patterns in argumentation is a challenging task which requires a well-defined scheme and detailed guidelines for human annotation. Designing a well-defined annotation scheme itself is quite challenging because a single argumentation can be interpreted in different ways (especially if the argumentation is long) and even if the interpretation is same, it can still be represented or described in different ways. Another difficulty here is that, the more complex the annotation is, the more costly the expert annotation would be which hinders the creation of large scale datasets. Therefore, the future directions regarding this task would be refining the annotation scheme to make it as simple and as uncomplicated as possible and crowdsourcing the annotations to reduce the cost and increase the number of annotated data.

The thesis in hand empirically confirmed that if we have the annotations of the reasoning patterns, automatic identification of such patterns is plausible. However, the existing pre-trained language models struggles to predict the reasons in the logic patterns. Besides, we performed automated evaluation for this task which can not identify if there are any predicted patterns which do not match with the gold human annotation but are correct. Therefore, the future direction would be performing manual evaluation for this task and creating a more sophisticated model that can understand and predict the reasons in logic patterns properly.

Overall, in this thesis, we explored two tasks which are important to achieve the ultimate goal of providing automated feedback to students so that they can improve their

argumentation. One of these tasks is assessing the quality of argumentation by capturing its discourse and the other is capturing the underlying reasoning patterns of complex strategic moves in argumentation. For the former, this thesis has presented successful unsupervised strategies that improved the argumentation assessment performance by capturing its discourse. For the latter task, this thesis has built the foundational ground and established the task by creating a scheme, a dataset and a baseline model and provided future directions for improving it.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*, 2019.

Stergos Afantenos and Nicholas Asher. Counter-argumentation and discourse: A case study. CEUR Workshop Proceedings, 2014.

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.

Khalid Al-Khatib, Yufang Hou, Henning Wachsmuth, Charles Jochim, Francesca Bonin, and Benno Stein. End-to-end argumentation knowledge graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7367–7374, 2020.

Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. Employing argumentation knowledge graphs for neural argument generation. 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 4744–4754, 2021.

Tariq Alhindi, Smaranda Muresan, and Daniel Preoţiuc-Pietro. fact vs. opinion: The role of argumentation features in news classification. In *Proceedings of the 28th international conference on computational linguistics*, pages 6139–6149, 2020.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*, 2016.

Emily Allaway and Kathleen McKeown. Zero-shot stance detection: A dataset and model using generalized topic representations. *arXiv preprint arXiv:2010.03640*, 2020.

Milad Alshomary, Shahbaz Syed, Arkajit Dhar, Martin Potthast, and Henning Wachsmuth. Argument undermining: Counter-argument generation by attacking weak premises. *arXiv preprint arXiv:2105.11752*, 2021.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational linguistics*, 34(4):555–596, 2008.

Yigal Attali and Jill Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

Maria Becker, Katharina Korfhage, and Anette Frank. Implicit knowledge in argumentative texts: an annotated corpus. *arXiv preprint arXiv:1912.10161*, 2019.

Maria Becker, Ioana Hulpuş, Juri Opitz, Debjit Paul, Jonathan Kobbe, Heiner Stuckenschmidt, and Anette Frank. Explaining arguments with background knowledge. *Datenbank-Spektrum*, 20(2):131–141, 2020.

Maria Becker, Katharina Korfhage, Debjit Paul, and Anette Frank. Co-nnect: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. *arXiv preprint arXiv:2105.03157*, 2021a.

Maria Becker, Siting Liang, and Anette Frank. Reconstructing implicit knowledge with language models. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 11–24, 2021b.

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

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. TOEFL11: A corpus of non-native English. *ETS Research Report Series*, 2013(2): i–15, 2013.

Elena Cabrio and Serena Villata. Natural language arguments: A combined approach. In *ECAI 2012*, pages 205–210. IOS Press, 2012.

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

Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. Language model pre-training for hierarchical document representations. *arXiv preprint arXiv:1901.09128*, 2019.

Di Chen, Jiachen Du, Lidong Bing, and Ruifeng Xu. Hybrid neural attention for agreement/disagreement inference in online debates. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 665–670, 2018.

Hongbo Chen and Ben He. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, 2013.

Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, 2017.

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

Robin Cohen. Analyzing the structure of argumentative discourse. *Computational linguistics*, 13:11–24, 1987.

Marcos Cramer and Mathieu Guillaume. Directionality of attacks in natural language argumentation. In *Bridging@ IJCAI/ECAI*, 2018.

Ronan Cummins and Marek Rei. Neural multi-task learning in automated assessment. *arXiv preprint arXiv:1801.06830*, 2018.

Mamoru Deguchi and Kazunori Yamaguchi. Argument component classification by relation identification by neural network and textrank. In *Proceedings of the 6th Workshop on Argument Mining*, pages 83–91, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fei Dong and Yue Zhang. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, 2016.

Fei Dong, Yue Zhang, and Jie Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, 2017.

Esin Durmus, Faisal Ladhak, and Claire Cardie. Determining relative argument specificity and stance for complex argumentative structures. *arXiv preprint arXiv:1906.11313*, 2019.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. *arXiv preprint arXiv:1704.06104*, 2017.

Roxanne El Baff, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. Challenge or empower: Revisiting argumentation quality in a news editorial corpus. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 454–464, 2018.

Roxanne El Baff, Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. Analyzing the persuasive effect of style in news editorial argumentation. Association for Computational Linguistics, 2020.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. Neural automated essay scoring and coherence modeling for adversarially crafted input. *arXiv preprint arXiv:1804.06898*, 2018.

Vanessa Wei Feng and Graeme Hirst. Classifying arguments by scheme. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 987–996, 2011.

Aris Fergadis, Dimitris Pappas, Antonia Karamolegkou, and Harris Papageorgiou. Argumentation mining in scientific literature for sustainable development. In *Proceedings of the 8th Workshop on Argument Mining*, pages 100–111, 2021.

James B Freeman. Argument structure and disciplinary perspective. *Argumentation*, 15 (4):397–423, 2001.

Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–554, 2016.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. International corpus of learner English, 2009.

Nancy L Green, E Cabrio, S Villata, and A Wyner. Argumentation for scientific claims in a biomedical research article. In *ArgNLP*, pages 21–25, 2014.

Shai Gretz, Yonatan Bilu, Edo Cohen-Karlik, and Noam Slonim. The workweek is the best time to start a family–a study of gpt-2 based claim generation. *arXiv preprint arXiv:2010.06185*, 2020.

Vivek Gupta, Ankit Saw, Pegah Nokhiz, Praneeth Netrapalli, Piyush Rai, and Partha Talukdar. P-sif: Document embeddings using partition averaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 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. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, 2016.

Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179, 2017.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. 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 1930–1940, 2018.

Miochael AK Halliday. An introduction to functional grammar 2nd edition. *London: Arnold*, 1994.

John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192, 2004.

Xinyu Hua and Lu Wang. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–230, 2018.

Xinyu Hua, Zhe Hu, and Lu Wang. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, 2019.

Ioana Hulpus, Jonathan Kobbe, Christian Meilicke, Heiner Stuckenschmidt, Maria Becker, Juri Opitz, Vivi Nastase, and Anette Frank. Towards explaining natural language arguments with background knowledge. In *PROFILES/SEMEX@ ISWC*, pages 62–77, 2019.

Radu Tudor Ionescu and Andrei Madalin Butnaru. Vector of locally-aggregated word embeddings (vlawe): A novel document-level representation. In *NAACL-HLT*, 2019.

S Ishikawa. ICNALE: the international corpus network of Asian learners of English. *Retrieved on November*, 21:2014, 2013.

Yangfeng Ji and Noah A Smith. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, 2017.

Yohan Jo, Seojin Bang, Emaad Manzoor, Eduard Hovy, and Chris Reed. Detecting attackable sentences in arguments. *arXiv preprint arXiv:2010.02660*, 2020.

Yohan Jo, Seojin Bang, Chris Reed, and Eduard Hovy. Classifying argumentative relations using logical mechanisms and argumentation schemes. *arXiv preprint arXiv:2105.07571*, 2021a.

Yohan Jo, Haneul Yoo, JinYeong Bak, Alice Oh, Chris Reed, and Eduard Hovy. Knowledge-enhanced evidence retrieval for counterargument generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3074–3094, 2021b.

Jonathan Kobbe, Juri Opitz, Maria Becker, Ioana Hulpus, Heiner Stuckenschmidt, and Anette Frank. Exploiting background knowledge for argumentative relation classification. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

Jonathan Kobbe, Ioana Hulpuș, and Heiner Stuckenschmidt. Unsupervised stance detection for arguments from consequences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 50–60, 2020.

Leah S Larkey. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. An argument-annotated corpus of scientific publications. Association for Computational Linguistics, 2018.

John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. Aifdb: Infrastructure for the argument web. In *Computational Models of Argument*, pages 515–516. IOS Press, 2012.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500, 2014.

Marcin Lewiński and Dima Mohammed. Argumentation theory. *The International Encyclopedia of Communication Theory and Philosophy*, pages 1–15, 2016.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict and analyse review helpfulness. *arXiv preprint arXiv:1707.07279*, 2017.

Jiawei Liu, Yang Xu, and Yaguang Zhu. Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744*, 2019a.

Junhao Liu, Zhen Hai, Min Yang, and Lidong Bing. Multi-perspective coherent reasoning for helpfulness prediction of multimodal reviews. 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 5927–5936, 2021.

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.

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

Daniel Marcu. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, 26(3):395–448, 2000.

Sandeep Mathias and Pushpak Bhattacharyya. Thank "goodness"! a way to measure style in student essays. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, 2018.

Mohsen Mesgar and Michael Strube. A Neural Local Coherence Model for Text Quality Assessment. In *Proceedings of the 2018 Conference on EMNLP*, pages 4328–4339, 2018.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 378–385, 2019a.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. Unsupervised learning of discourse-aware text representation. 2019b.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. Corruption is not all bad: Incorporating discourse structure into pre-training via corruption for essay scoring. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

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. *arXiv preprint arXiv:2204.01512*, 2022.

Nathalie Muller Mirza and Anne-Nelly Perret-Clermont. *Argumentation and education: Theoretical foundations and practices*. Springer Science & Business Media, 2009.

Raquel Mochales and Marie-Francine Moens. Study on the structure of argumentation in case law. In *Proceedings of the 2008 conference on legal knowledge and information systems*, pages 11–20, 2008.

Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. Automated essay scoring with discourse-aware neural models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 484–493, 2019.

Shoichi Naito, Shintaro Sawada, Nakagawa Chihiro, Naoya Inoue, Kenshi Yamaguchi, Iori Shimizu, Farjana Sultana Mim, Keshav Singh, and Kentaro Inui. Typic: A corpus of template-based diagnostic comments on argumentation. *arXiv preprint*, 2022.

Muller Mirza Nathalie. Can we learn through disagreements?: A sociocultural perspective on argumentative interactions in a pedagogical setting in higher education. *Inovacije u nastavi-časopis za savremenu nastavu*, 28(3):145–166, 2015.

Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument mining with structured svms and rnns. *arXiv preprint arXiv:1704.06869*, 2017.

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, 2015.

Isaac Persing and Vincent Ng. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, 2013.

Isaac Persing and Vincent Ng. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, 2014.

Isaac Persing and Vincent Ng. Modeling argument strength in student essays. 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 543–552, 2015.

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, 2016a.

Isaac Persing and Vincent Ng. Modeling stance in student essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2174–2184, 2016b.

Isaac Persing, Alan Davis, and Vincent Ng. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, 2010.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 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, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

Zahra Rahimi, Diane Litman, Elaine Wang, and Richard Correnti. Incorporating coherence of topics as a criterion in automatic response-to-text assessment of the organization of writing. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 20–30, 2015.

Chris Reed. Preliminary results from an argument corpus. *Linguistics in the twenty-first century*, pages 185–196, 2006.

Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. Language resources for studying argument. In *Proceedings of the 6th conference on language resources and evaluation-LREC 2008*, pages 2613–2618. ELRA, 2008.

Paul Reisert, Naoya Inoue, Tatsuki Kuribayashi, and Kentaro Inui. Feasible annotation scheme for capturing policy argument reasoning using argument templates. In *Proceedings of the 5th Workshop on Argument Mining*, pages 79–89, 2018.

Paul Reisert, Benjamin Heinzerling, Naoya Inoue, Shun Kiyono, and Kentaro Inui. Riposte! a large corpus of counter-arguments. *arXiv preprint arXiv:1910.03246*, 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.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chungmin Lee. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, 2017.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. Explagraphs: An explanation graph generation task for structured commonsense reasoning. *arXiv preprint arXiv:2104.07644*, 2021.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Keshav Singh, Farjana Sultana Mim, Naoya Inoue, Shoichi Naito, and Kentaro Inui. Exploring methodologies for collecting high-quality implicit reasoning in arguments. In *Proceedings of the 8th Workshop on Argument Mining*, pages 57–66, 2021.

Wilbert Spooren and Liesbeth Degand. Coding coherence relations: Reliability and validity. 2010.

Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 1501–1510, 2014a.

Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014b.

Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, 2017.

Kenneth Steimel and Brian Riordan. Towards instance-based content scoring with pre-trained transformer models.

Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on EMNLP*, pages 1882–1891, 2016.

Christoph Unger. *Genre, relevance and global coherence: The pragmatics of discourse type*. Springer, 2006.

Teun A Van Dijk. The study of discourse. *Discourse as structure and process*, 1(34): 703–52, 1997.

FH van Eemeren, B Garssen, ECW Krabbe, AF Snoeck Henkemans, B Verheij, JHM Wagemans, et al. Handbook of argumentation theory. 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, and Gregor Engels. Modeling review argumentation for robust sentiment analysis. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 553–564, 2014.

Henning Wachsmuth, Johannes Kiesel, and Benno Stein. Sentiment flow-a general model of web review argumentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 601–611, 2015.

Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691, 2016.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. Computational argumentation quality assessment in natural language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, 2017.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, 2018.

Marilyn Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, 2012.

Douglas Walton. Objections, rebuttals and refutations. 2009.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

Douglas Walton, Katie Atkinson, et al. Argumentation in the framework of deliberation dialogue. In *Arguing global governance*, pages 230–250. Routledge, 2010.

Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuan-Jing Huang. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 791–797, 2018.

Lingfei Wu, Ian EH Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J Witbrock. Word Mover's Embedding: From Word2Vec to Document Embedding. *arXiv preprint arXiv:1811.01713*, 2018.

Chang Xu, Cécile Paris, Surya Nepal, and Ross Sparks. Cross-target stance classification with self-attention networks. *arXiv preprint arXiv:1805.06593*, 2018.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. Recognising agreement and disagreement between stances with reason comparing networks. *arXiv preprint arXiv:1906.01392*, 2019.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, 2020.

Deniz Zeyrek, Işın Demirşahin, and Ayışığı B Sevdik Çallı. Turkish discourse bank: Porting a discourse annotation style to a morphologically rich language. *Dialogue & Discourse*, 4(2):174–184, 2013.

Haoran Zhang and Diane Litman. Co-attention based neural network for source-dependent essay scoring. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2018.

Renxian Zhang. Sentence ordering driven by local and global coherence for summary generation. In *Proceedings of the ACL 2011 Student Session*, pages 6–11, 2011.

Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*, 2019.

List of Publications

Journal Papers (Refereed)

1. Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi and Kentaro Inui. Corruption Is Not All Bad: Incorporating Discourse Structure Into Pre-Training via Corruption for Essay Scoring. In IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol.29, pp.2202-2215, 2021.

International Conference Papers (Refereed)

1. 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 13th International Conference on the Language Resources and Evaluation Conference (LREC), June 2022.

2. Keshav Singh, Naoya Inoue, Farjana Sultana Mim, Shoichi Naito and Kentaro Inui. IRAC: A Domain-specific Annotated Corpus of Implicit Reasoning in Arguments. In Proceedings of the 13th International Conference on the Language Resources and Evaluation Conference (LREC), June 2022.

3. 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 Proceedings of the 13th International Conference on the Language Resources and Evaluation Conference (LREC), June 2022.

4. Keshav Singh, Farjana Sultana Mim, Naoya Inoue, Shoichi Naito, Kentaro Inui. Exploring Methodologies for Collecting High-Quality Implicit Reasoning in Arguments. Proceedings of the 8th Workshop on Argument Mining, pages 57–66,

November 10–11, 2021.

5. Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi and Kentaro Inui. Unsupervised Learning of Discourse-Aware Text Representation for Essay Scoring. In Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop (ACL SRW 2019), pp. 378–385, July 2019.

Other Publications (Not refereed)

1. Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, Kentaro Inui. Unsupervised Learning of Discourse-Aware Text Representation. 言語処理学会第25回年次大会, pp.1471-1474, March 2019.