ARGUMENT MINING FROM BIOMEDICAL LITERATURE WITH STRUCTURAL FEATURES

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2024

Student id: 10791765

Department of Computer Science

Contents

Word Count: 42761

List of Tables

List of Figures

Abstract

ARGUMENT MINING FROM BIOMEDICAL LITERATURE WITH STRUCTURAL FEATURES

Boyang Liu

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2024

Argument mining (AM), the process of automatically extracting argument structure from an argumentative document, has emerged as a crucial area of research within natural language processing (NLP) recently. It has a high impact on real-world applications such as decision-making, information retrieval, fact-checking and fake news detection. Correctly understanding an argument is challenging because it requires understanding not only each sentence, but also the global structural features of the document that these sentences make up. However, most previous AM models only pay attention to a single argument component (AC) to classify the type of AC or a pair of ACs to identify and classify the argument relation (AR) between the two ACs. These models tend to focus more on local features rather than global structural features and ignore the fact that argumentative texts of the same genre such as biomedical literature have global similarities in text organisation and argument structure due to the implicit rule of scientific paper writing. We refer to these similarities as genre-specific structural features and argument-specific structural features, respectively, in this thesis. Therefore, this thesis aims to explore the impact of structural features in argument mining.

Firstly, we explore the impact of text zoning labels as the representation of genre-specific structural features on AM. One type of text zoning schemes, argumentative zoning, considered as a forerunner of AM, can be regarded as a coarse-grained argument structure at the sentence level. It shows the common organisation of an abstract in biomedical literature. To leverage such structural features, we propose a method based on multi-head attention for argument component identification and classification subtasks. The results on two biomedical argument mining datasets demonstrate the positive impact of the genre structural features on AM.

Further, with the successful trend of treating other NLP tasks as machine reading comprehension (MRC) tasks to mimic the logical process that how humans do recently, which is similar to the reasoning logic of an argument, we propose a multi-turn MRC model that generates the argument structure incrementally to exploit graph-level argument-specific structural features. Specifically, at each turn, all ACs related to the query AC are generated simultaneously, such that the interaction that reveals the structural features of a group of ACs between the answer ACs is considered. In addition, the partially constructed graph is used as sub-graph level argument-specific structural features through a graph convolutional network to support the extension of the graph with additional ACs. Experiments performed on two biomedical argument mining corpus demonstrate the effectiveness of our method in terms of improving the model performance.

Finally, since the multi-turn MRC model suffers from the problem of error propagation, we propose a generative multi-hop MRC model to alleviate error propagation while exploiting argument-specific structural features. This multi-hop MRC model learns the path-level structural features and extracts the argument structure at the same time. We validate the proposed method on the same two datasets, showing that even the path information is enough as a representation for the structural features to improve the performance.

Overall, we illustrate that exploring the use of global structural features is an important step towards improving argument mining. The use of structural features is a promising way for argument mining and should receive more attention from researchers.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's policy on presentation of Theses

v. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the University of Manchester's products or services. Internal or personal use of this material is permitted. If

interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Acknowledgements

The journey of my doctoral career is a process of continuous growth and degradation. During this period, I encountered many difficulties and obstacles. It is impossible for me to reach the end of the journey without the help of others. I am honored to meet you who have helped me on my scientific research journey.

The first person I would like to express my gratitude to is my supervisor Prof. Sophia Ananiadou. Thank you to her for choosing me as her student and giving me the opportunity to start this journey. Especially in the early stage of my scientific research, her detailed guidance, selfless dedication and encouragement helped me integrate into my doctoral life faster. When I encountered difficulties, her trust and encouragement in me made me feel warm and motivated me to step out of the shadows and keep moving forward. I feel very lucky to have her as my supervisor.

At the same time, I would like to thank my co-supervisors Dr. Viktor Schlegel and Dr. Riza Theresa Batista-Navarro. Viktor gave me valuable suggestions and feedback throughout the entire research process, helping me to continuously improve my research work. He helped me revise my papers many times, and I learned a lot of paper writing skills from him. Riza provided me with important inspiration and guidance in the early stages of my research, allowing me to better understand the direction of my research. Their professional support has made me more confident and determined in my academic exploration.

I would also like to thank my parents, who are my biggest supporters in life. Their dedication and support allowed me to focus on my studies and face challenges with confidence. Their love is the source of my continuous efforts.

An additional thank you goes to my roommate Tianlin. In the moments outside of the research, we shared laughter, frustration, and growth together, and this experience will be a treasured memory for me.

I would like to express my gratitude to all my current and past colleagues at NaCTeM from whom I received a lot of help and pleasure: Laura, Jake, Chenhan, Zheheng, Kailai,

Hassan, Meizhi, Maolin, Paul, Nhung.

Finally, I would like to thank all other people who have helped and supported me in my academic career, i.e., the reviewers, the staff in the university. Your generous help and selfless sharing have made my research path more colorful. My gratitude is beyond words, and I will continue to work hard to repay the society and make more positive contributions to academia and society.

List of Abbreviations

AC Argument Component

ACC Argument Component Classification

ACI Argument Component Identification

ACIC Argument Component Identification and Classification

AM Argument Mining

AR Argument Relation

ARC Argument Relation Classification

ARI Argument Relation Identification

ARIC Argument Relation Identification and Classification

AZ Argumentative Zoning

BART Bidirectional Auto-Regressive Transformers

BERT Pre-training of Deep Bidirectional Transformers for Language Understanding

CEL Cross-Entropy Loss

CNN Convolutional Neural Network

CRF Conditional Random Fields

DT Decision Tree

EBM Evidence-Based Medicine

FFN Feed Forward Network

GCN Graph Convolutional Network

GRU Gate Recurrent Unit

LR Logistic Regression

LSTM Long-Short Term Memory

MRC Machine Reading Comprehension

NB Naive Bayes

NLP Natural Language Processing

RF Random Forests

RNN Recurrent Neural Network

RST Rhetorical Structure Theory

SVM Support Vector Machines

Chapter 1

Introduction

1.1 Motivation

Argumentation is a reasoning process reaching conclusions based on evidence that supports or attacks a claim with the intent of influencing others (Micheli, 2012). It has a long historical foundation that can be traced back to the time of Aristotle, involving multiple fields such as logic, philosophy, psychology, and linguistics (Habernal and Gurevych, 2017). In our daily life, we often encounter various forms of arguments in our daily life. For example, politicians use arguments and logic to support their policies and positions; advertisers use data and cases to promote products or services; users on the forum support their opinions by citing authoritative sources; scholars cite previous research in academic papers to support their hypotheses and conclusions.

This argument-based communication method is one of the important methods of information exchange and opinion exchange in modern society. With the development of information technology, the number of argumentative texts on the Internet has exploded. Making good use of these argumentative texts contains huge commercial value. Therefore, how to automatically extract argument structures from unstructured texts is an urgent problem to be solved. Although opinion mining (Xia et al., 2021; Liu et al., 2015) and sentiment analysis (Deng et al., 2023; Bao et al., 2023) have been extensively studied and can be used to tell us what these opinions are used for, they cannot explain why people hold these opinions. Therefore, the emerging field of argument mining (AM) has emerged.

Argument mining (also known as argumentation mining) is a branch of natural language processing (NLP), aiming to extract argument structures including argument components (AC) such as claims and premises, and argument relations (AR) between

ACs, i.e., support and attack. It has been proven beneficial in various tasks, such as information retrieval (Stab et al., 2018), automated essay scoring (Ke et al., 2018), legal decision support (Walker et al., 2018), peer reviewing (Fromm et al., 2021; Chen et al., 2022b) and fact-checking (Wührl and Klinger, 2022).

The field of biomedicine is a scientific domain that is constantly advancing and evolving, relying on a vast array of literature to document and disseminate knowledge. With the explosive growth in the number of scientific publications, traditional methods of literature retrieval and reading are no longer sufficient to meet researchers' demands for efficiency and accuracy in information acquisition. Consequently, biomedical literature mining has become a critically important area of research, aiming to extract valuable information from a vast array of literature through automated methods to support scientific research and clinical decision-making.

In medical practice, clinical decision-making is an indispensable aspect of a physician's daily work. Clinical decision-making refers to a series of decisions that a physician needs to make when faced with a patient's condition, medical history, and clinical examination results, in order to formulate the most appropriate treatment plan. This process involves the physician's professional knowledge, experience, and a profound understanding of the patient's situation. Physicians need to constantly update their medical knowledge, and stay informed about the latest research findings, treatment methods, and diagnostic technologies.

In this context, physicians must navigate through extensive medical information while tailoring optimal treatment plans based on individual patient differences, making it an exceptionally challenging task.

To better address the complexity of clinical decision-making, we use Evidence-Based Medicine (EBM) (Sackett, 1997). EBM combines the latest scientific research evidence with clinical experience to formulate optimal medical decisions. The rise of EBM has transformed the traditional model of medical practice. It encourages physicians to not only focus on the specific circumstances of patients but also to incorporate the latest research findings into consideration, enabling the formulation of treatment plans in a more comprehensive and scientific manner. Through systematic reviews and synthesis of a large body of literature, EBM helps physicians gain a more accurate understanding of the effects, side effects, and indications of different treatment methods, thus providing patients with more personalized and optimized medical services.

Although EBM provides a more scientific approach to decision-making for physicians, the exponential growth of knowledge in the field of medical research poses new challenges. In the current era of knowledge explosion, scientific research output is experiencing a massive and rapid increase. Particularly in addressing global health crises like COVID-19, medical research institutions and scientists have accelerated research on pathogens, vaccines, and treatment methods, resulting in a flood of new research findings. This abundance of research outcomes is both a valuable information resource and a substantial information burden for physicians. Doctors not only need to stay constantly updated on the latest research developments but also must sift through vast literature to identify evidence relevant to specific patient conditions. Manually searching, evaluating, and integrating this extensive literature is an extremely time-consuming and tedious task, especially considering the enormous workload physicians face every day, making it nearly impractical. Although Google and ChatGPT are already good tools, they have some drawbacks in extracting evidence from biomedical literature. The results returned by Google are too coarse, while ChatGPT suffers from the problem of hallucination, which can cause serious consequences.

With the development of deep learning, neural networks, and pre-trained language models, training an effective argument mining model to automatically extract argument structures seems to be an effective solution. As it enables physicians to efficiently obtain key information and comprehensively understand viewpoints and conclusions in biomedical literature (Mayer et al., 2021). By applying this technology on a large scale in literature, physicians can more quickly locate evidence relevant to patient conditions, alleviating the burden of manually retrieving literature. The introduction of argumentation mining technology makes the acquisition of medical information more automated and intelligent, providing physicians with a powerful intelligent assistant.

Although argument mining from the biomedical literature has many benefits, research in this domain is scarce and existing argument mining models are not good enough for this task. Specifically, previous work has mostly focused on local features without much use of global features, such as the genre-specific structural features and argument-specific structural features of argumentative texts. These global features are of great help to AM. Specifically, biomedical abstracts usually start from conclusion statements that are claims of other publications as the background information of a biomedical abstract often contains conclusions of previous literature as background information or motivation of the new research. Such statements may be mistakenly recognised by previous argument mining models as claims of the current literature

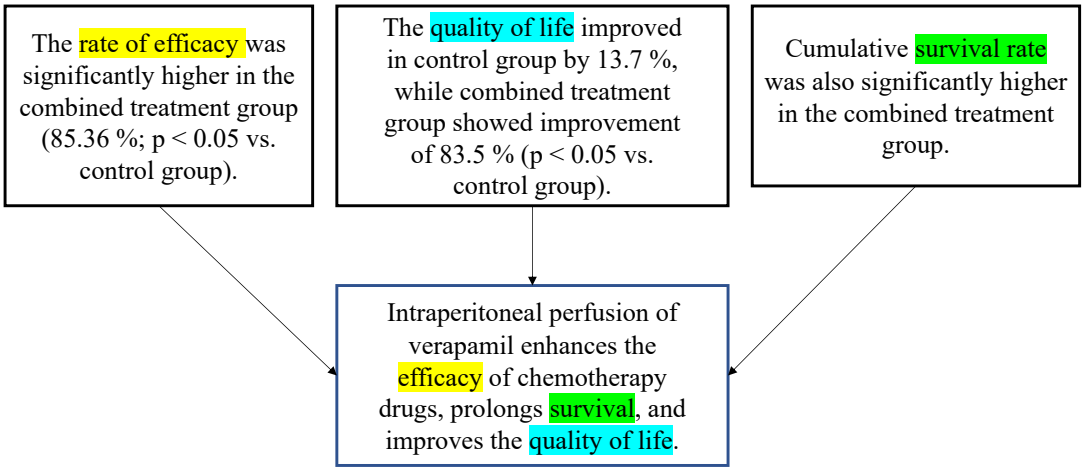


Figure 1.1: Part of the argument structure of the abstracts from PubMed 23589316 (Jia et al., 2013). The text in the blue rectangle represents a claim and texts in black rectangles denote premises. Two ACs connected with an arrow means that there is a support relation between them. Texts in yellow, green, and cyan represent aspects mentioned in argument components. Different mentions of the same aspect are highlighted with the same colour.

if they only focus on local features. For example, in *"Older breast cancer survivors (BCSs) are at risk for late and long-term treatment effects on quality of life (QOL), including lower physical functioning and fear of recurrence."*, the first sentence is used to describe the background information and it can be used as a claim in the abstract. However, the real claim of this paper is *"The combination of self-directed movement and mindfulness, as tested here, may be a valuable tool for promoting health and well-being in older long-term survivors of breast cancer"*, which is highly related to the background sentence. If the global genre-specific structural features are included and the model has learned that claims are more likely to appear at the end of an abstract rather than at the beginning, the model will recognise the background sentence as non-argumentative sentence in the current abstract. Another example is shown in Figure 1.1. In this example, the claim in the blue rectangle is designed with three aspects, named *efficacy*, *survival rate* and *quality of life*. If the model only focuses on the claim and one of the premises when learning the argument relations, the model can only learn part of the semantic information of the argument because a single premise is not enough to support the entire argument, and the relations between the three premises will be ignored.

However, pre-trained language models cannot effectively learn such structural features, especially argument-specific structural features, which demonstrate the multi-step reasoning process from premises to claim that authors of argumentative texts employ. Many prior researchers have demonstrated the difficulty language models face in handling such multi-step reasoning, particularly in the argument mining domain where training data is limited (Rae et al., 2021; Bommasani et al., 2021; Valmeekam et al., 2022). A good illustration of this difficulty is that ChatGPT requires a manually designed chain-of-thought to improve its performance, which indicates it cannot effectively process the reasoning steps for a specific task (Wang et al., 2023a; Zhao et al., 2023). In order to fill this gap, this thesis studies how global structural features can help the model to better identify the argument structure of a paper.

Specifically, we employ text zoning features to incorporate genre-specific features of biomedical literature. Text zoning involves segmenting a text into zones (Gnehm, 2018), each characterized by its unique function and content. This method captures the structural organisation of the text, offering crucial contextual insights for interpreting arguments within their respective genres. As for argument-specific structural features, with the successful trend of treating other NLP tasks as machine reading comprehension (MRC) tasks (Liu et al., 2023c; Malhas and Elsayed, 2022; Zhang et al., 2022), we plan to treat the AM task as an MRC task to leverage such features. It is because the multi-turn MRC task (Zhou et al., 2022) and multi-hop MRC task (Jiang et al., 2019) can be regarded as good ways to mimic the logical process that how humans do recently, which is similar to the reasoning logic of an argument.

1.2 Research Questions, Hypotheses and Objectives

In this section we address the key components of this thesis, including the formulation of our research questions, hypotheses, and objectives. Each of these elements is systematically examined and presented individually to provide a thorough understanding of the underlying framework guiding our research.

RQ_1 Does the identification of genre-specific structural features of biomedical literature facilitate the mining of argument components in biomedical literature abstracts?

H_1 The genre-specific structural features of biomedical literature are helpful for mining argument components in biomedical literature abstracts as it can be used to locate the argumentative parts.

$O_{1.1}$ Investigate and select available schemes to describe genre-specific structural features of biomedical literature. (C1)

$O_{1.2}$ Analyse the relation between genre-specific structural features of biomedical literature and argument components in biomedical literature abstracts.(C1)

$O_{1.3}$ Design a model that can benefit from the genre-specific structural features of biomedical literature. (C2)

$O_{1.4}$ Validate the proposed approach on argument mining datasets and compare them with state-of-the-art approaches. (C2)

$O_{1.5}$ Analyse the effect of zoning labels as structural features on AM. (C2)

RQ_2 Can the predicted argument (sub-)graphs be used as argument-specific structural features to improve the performance of the model?

$H_{2.1}$ AM can be modeled as a graph generation process by transferring it into a task such as muti-turn machine reading comprehension.

$O_{2.1}$ Develop a multi-turn MRC-based argument mining model to incorporate graph-level argument-specific structural features. (C3,C4)

$H_{2.2}$ The graph generation process allows the utilisation of graph-level argument-specific structural features through graph neural network.

$O_{2.2}$ Validate the proposed approach on argument mining datasets and compare them with state-of-the-art approaches. (C3,C4)

$O_{2.3}$ Analyse the effect of graph-level argument-specific structural features on different AM subtasks. (C3,C4)

RQ_3 Can each argument be viewed as a chain of reasoning within MRC so that the reasoning paths can be used as argument-specific structural features?

$H_{3.1}$ The argumentative process can be viewed as a chain of reasoning path.

$O_{3.1}$ Cast the process of argumentation as a reasoning path. (C5)

$H_{3.2}$ The AM task can be transfered as a multi-hop MRC task to explicitly learn this reasoning path.

$O_{3.2}$ Enhance the model's perception of the argument-specific structural features by treating the AM task as a multi-hop MRC task so that the model can explicitly learn this chain. (C5)

$H_{3.3}$ This reasoning path can improve the performance of AM models.

$O_{3.3}$ Validate the proposed approach on argument mining datasets and compare them with state-of-the-art approaches. (C5)

$O_{3.4}$ Analyse the effect of reasoning paths as structural features on different AM subtasks. (C5)

1.3 Contributions and Publications

1.3.1 Contributions

The contributions (C) of this thesis include the following points:

C_1 We employ argumentative zoning as a scheme to model the genre-specific structural features of biomedical literature abstracts. We show that the argumentative zoning labels are highly related to ACs. (Chapter 3)

C_2 We utilise zoning information in the tasks of argument component identification and classification. We propose a direct yet effective method for exploiting regularities in the writing style of biomedical abstracts, to verify the effectiveness of zoning information and minimise the impact of changes in model complexity. Experimental evaluation shows that zoning information is helpful in both token-level and sentence-level argument mining tasks. (Chapter 3)

C_3 We propose a two-step model to explicitly utilise global information to solve the AM task. During the first stage, we propose a top-down multi-turn QA-based model which is used to solve the ARI subtask to get the initial graph as the graph-level global information. Then we propose a GCN-based model in the second stage to predict the type of AR and AC based on the whole initial argumentative graph. Experimental results show that the use of global information has a significant positive impact on the performance of all these tasks. (Chapter 4)

C_4 We propose a method that transforms the ARI subtask into a multi-label classification task so that the model is enabled to implicitly learn the correlations among multiple ACs that are related to the same AC. (Chapter 4)

C_5 We transfer the argument mining task as a multi-hop generative MRC task, which gives a way to leverage the "chain of thought" of an argument in a generative

manner for the argument mining task. The extensive experimental results and detailed analysis demonstrate the positive impact of the "chain of thought" as the argument-specific structural features. (Chapter 5)

1.3.2 Publications and Author Contribution Statement

A significant portion of the work presented in this thesis has already been published in peer reviewed journal and conference papers. The existing, improvements or additional details related to the publications are included in this thesis, which are outlined in the respective chapters. Author contribution statements are also included according to the CRediT author statement guidelines [1].

⋆ **Liu, B.**, Schlegel, V., Batista-Navarro, R. T., & Ananiadou, S. (2022). Incorporating zoning information into argument mining from biomedical literature. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 6162-6169).

CRediT: **Boyang Liu**: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. Viktor Schlegel: Methodology, Supervision, Writing – review & editing. Riza Theresa Batista-Navarro: Methodology, Supervision, Writing – review & editing. Sophia Ananiadou: Methodology, Funding acquisition, Supervision, Writing – review & editing.

⋆ **Liu, B.**, Schlegel, V., Thompson, P., Batista-Navarro, R. T., & Ananiadou, S. (2023). Global information-aware argument mining based on a top-down multi-turn QA model. Information Processing & Management, 60(5), 103445.

CRediT: **Boyang Liu**: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. Viktor Schlegel: Methodology, Supervision, Writing – review & editing. Paul Thompson: Writing – review & editing. Riza Theresa Batista-Navarro: Supervision, Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

⋆ **Liu, B.**, Schlegel, V., Batista-Navarro, R. T., & Ananiadou, S. (2023). Argument mining as a multi-hop generative machine reading comprehension task. In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 10846-10858).

CRediT: **Boyang Liu**: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. Viktor Schlegel: Methodology, Supervision, Writing – review & editing. Riza Theresa Batista-Navarro: Supervision, Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review &

[1] https://www.elsevier.com/authors/policies-and-guidelines/credit-author-state ment

editing.

Additional Publication

The following articles have been completed during my PhD. These articles are not included in this thesis.

⋆ **Liu, B.**, Schlegel, V., Batista-Navarro, R., & Ananiadou, S. (2023). Entity Coreference and Co-occurrence Aware Argument Mining from Biomedical Literature. In Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023) at ACL 2023 (pp. 54-60).

CRediT: **Boyang Liu**: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. Viktor Schlegel: Methodology, Supervision, Writing – review & editing. Riza Theresa Batista-Navarro: Supervision, Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

⋆ **Liu, B.**, Schlegel, V., Batista-Navarro, R., & Ananiadou, S. (2023). Write2understand: Argument Mining with Argumentative Text Reconstruction. Submitted to *Knowledge-Based Systems*.

CRediT: **Boyang Liu**: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. Viktor Schlegel: Methodology, Supervision, Writing – review & editing. Riza Theresa Batista-Navarro: Supervision, Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

⋆ Liu, Z., **Liu, B.**, Thompson, P., Yang, K., & Ananiadou, S. (2024). ConspEmoLLM: Conspiracy Theory Detection Using an Emotion-Based Large Language Model. In ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024), volume 392 of Frontiers in Artificial Intelligence and Applications, pages 4649–4656. IOS Press.

CRediT: Zhiwei Liu: Methodology, Software, Formal analysis, Validation, Visualisation, Writing – original draft. **Boyang Liu**: Methodology, Software, Writing – review & editing. Paul Thompson: Writing – review & editing. Kailai Yang: Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

⋆ Zhang, T., Yang, K., Alhuzali, H., **Liu, B.**, & Ananiadou, S. (2023). PHQ-aware depressive symptoms identification with similarity contrastive learning on social media. Information Processing & Management, 60(5), 103417.

Tianlin Zhang: Methodology, Software, Data Curation, Validation, Formal analysis, Visualization, Writing – original draft. Kailai Yang: Methodology, Writing – review & editing. Hassan Alhuzali: Visualization, Writing – review & editing. **Boyang Liu**: Methodology, Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

⋆ Zhang, T., Yang, K., Ji, S., **Liu, B.**, Xie, Q., & Ananiadou, S. (2024). Suicidemoji: Derived emoji dataset and tasks for suicide-related social content. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024, pages 1136–1141. ACM.

Tianlin Zhang: Methodology, Software, Data Curation, Validation, Formal analysis, Visualization, Writing – original draft. Kailai Yang: Methodology, Writing – review & editing. Shaoxiong Ji: Visualization, Writing – review & editing. **Boyang Liu**: Writing – review & editing. Sophia Ananiadou: Funding acquisition, Supervision, Writing – review & editing.

1.4 Thesis Structure

There are six chapters in total in this thesis, with one introductory chapter, one background chapter, three main content chapters and one conclusion chapter. The first chapter, which is the present one, serves as the introduction. The remainder of the thesis reads as follows:

In Chapter 2, we first introduce some technical background related to this thesis, aiming to provide readers with a comprehensive understanding of the key concepts in the current field of NLP and neural networks. We introduce a series of common neural network models in detail, covering their structures, characteristics, and applications in natural language processing tasks. Then we focuses on a detailed introduction to the argument mining task, and a comprehensive overview of different approaches to the task. While detailing the task and methods, we also introduce the datasets used in AM. Finally, we conclude the limitations of previous models, which provide insights into this thesis.

In Chapter 3, we explore **RQ1**. We first introduce our motivation and then review a method used to represent genre-specific structural features of biomedical literature, namely text zoning. Further, we analyse the relationship between zoning labels and

AC types, and propose two models to use zoning information to help solve token-level and sentence-level AC-related subtasks respectively. Finally, we conduct detailed experiments and analysis on two datasets. This chapter is mainly based on our peer-reviewed publication Liu et al. (2022).

In Chapter 4, We first introduce our motivation regarding the importance of argument-specific structural features. Then, We describe our two-step top-down multi-turn QA-based model to explicitly utilise graph-level argument-specific structural features to solve the AM task. In this regard, we study **RQ2**. At the same time, we also model the ARI task as a multi-label classification task so that the model can implicitly learn the relationship between multiple ACs related to the same AC. Then, we compare our model with other baseline models on two datasets. Finally, detailed analysis is performed to demonstrate the benefits of our proposed model for the AM task. It is worth mentioning that the contents presented in this chapter have been published in Liu et al. (2023b).

Chapter 5 is related to **RQ3**. In this chapter, we first analyse the path-level argument-specific structural features that is similar to the concept named chain-of-thought contained in the argument and the connection between the multi-hop machine reading comprehension task and the chain-of-thought. Then we explain our method, which is to model the AM task as a multi-hop machine reading comprehension task to allow the model to implicitly learn path-level argument-specific structural features in the argument. Further, we conduct thorough evaluations and analyses of the proposed approach, illustrating its advantages for the task of AM. The content of this chapter is drawn from our publication (Liu et al., 2023a)

The final chapter is Chapter 6. In this chapter we first review the findings of each previous individual chapter and draw conclusions based on the entire study. We then discuss the limitations of this thesis and end the whole thesis with future work.

Chapter 2

Background

In this chapter, we firstly present the technical background for the remainder of this thesis, focusing on neural networks, fundamental information about basic network architectures and evaluation metrics, from which this thesis benefits. Secondly, we offer a comprehensive overview of argument mining. Specifically, we outline the task and subsequently organise the various subtasks into structured groups based on multiple perspectives. The chapter then extensively explores existing methodologies utilising structural features to offer insights into their capabilities and constraints. Furthermore, we elucidate prevalent datasets and typical argumentation models. It is pertinent to note that while our primary focus lies on argument mining within biomedical literature, our examination encompasses argument mining across all domains, given the scarcity of studies within the biomedical realm.

2.1 Neural Networks

Neural networks, also known as artificial neural networks, are mathematical models based on network topology theory to simulate the process of the human brain receiving external stimuli and performing complex calculations on these stimuli. A neural network can be regarded as a graph, in which the basic unit of the neural network is a neuron, that is, a node in the graph, and each neuron includes a series of parameters that can be trained. There can be an edge between two neurons for the transmission of information. In this section, we will introduce the neural network models used in this thesis.

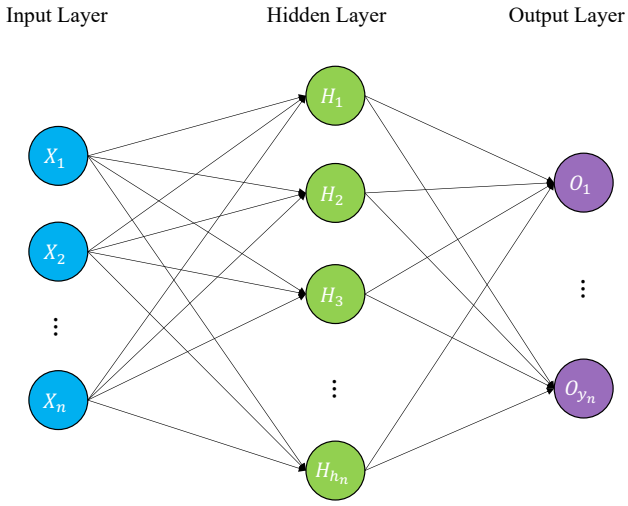


Figure 2.1: An illustration of a Feed-forward Neural Network with one hidden layer.

2.1.1 Feed-forward Neural Network (FNN)

Feed-forward neural networks (Yagawa and Oishi, 2021) are simple and basic neural networks. The architecture of FNNs usually consists of three parts: the input layer, hidden layer and output layer. The input layer is the first layer of a feed-forward neural network, which is used to receive the input of the network. The hidden layer is usually located between the input layer and the output layer, and can include one or more layers. Each layer receives the input of the previous layer, performs a weighted sum of the inputs, and then uses the activation function to generate output.

$$y = f(Wx + b) \tag{2.1}$$

where f is an activation function.

The hidden layer is used for feature pattern extraction. The last layer of a feed-forward neural network is the output layer, which computes the predictions of the model. Usually, the number of neurons contained in the output layer depends on the task being solved. For example, in classification tasks, the number of neurons is usually equal to the number of classes. A feed-forward neural network with one hidden layer is shown in Figure 2.1. It can be seen that the characteristic of feed-forward neural networks is that each neuron is only connected to the neurons of the previous layer. Therefore, in a feed-forward neural network, information flows in one direction from the input layer to the output layer. Specifically, each neuron receives the output of the previous layer and

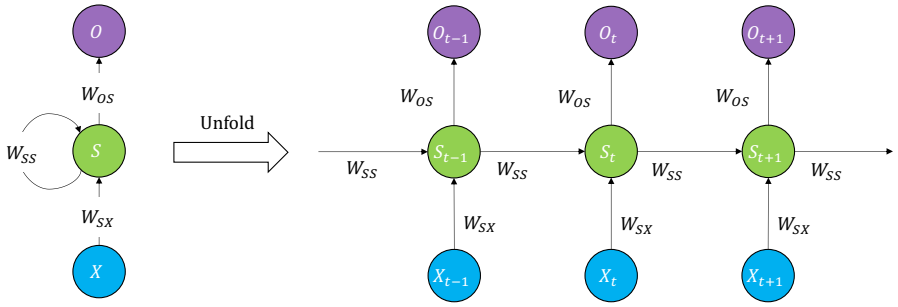


Figure 2.2: Vanilla RNN

outputs to the next layer, and there is no information transmission between neurons in the same layer.

2.1.2 Recurrent Neural Network (RNN)

2.1.2.1 Vanilla RNN

Although traditional neural networks can effectively model the global or local characteristics of data, when the input data of the neural network is in sequence form or has temporal dependence, it is difficult for traditional neural networks to process it effectively. To solve this problem, recurrent neural networks (Elman, 1990) are proposed for sequence and timing modelling. Unlike traditional feed-forward neural networks that process individual inputs independently, RNNs possess internal memory mechanisms that allow them to maintain a sense of context and capture dependencies within a sequence. The specific structure of RNN is shown in Figure 2.2.

RNNs have a chain structure, connecting the hidden layers of each time step, and inputting data in sequence according to the time sequence. The hidden representation h_t at the current time step t is composed of the input information x_t at the current time step t and the historical information h_{t-1} from previous time step $t-1$, and the calculation method of h_{t+1} of the next time step $t+1$ is the same. The calculation is repeated until the end of the last time step to obtain the complete feature matrix of the sequence. For each time step, its corresponding calculation is described in Equations 2.2-2.3:

$$o_t = g(W_{oh} h_t) \tag{2.2}$$

$$h_t = \sigma(W_{hx} x_t + W_{hh} h_{t-1}) \tag{2.3}$$

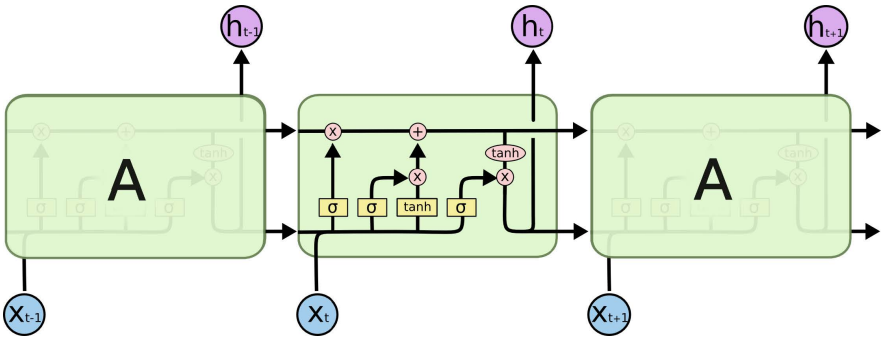


Figure 2.3: Long short-term memory cell

where o_t represents the value of the output layer, σ is an activation function; W_{hx}, W_{hh} and W_{oh} are weight matrices. RNNs use the same set of weights across all time steps, allowing them to share information throughout the sequence.

This mode makes the output of the current time step of the model depend on the output of the previous time step and the input of the current step. Therefore, the RNN model stores the historical information of the previous time step and establishes a contextual connection with the input information of the current time step to effectively solve sequence modeling.

2.1.2.2 Long Short-term Memory

Although traditional RNNs can effectively model sequence context, they suffer from vanishing gradient problems, which can hinder their ability to capture long-term dependencies. This issue arises when gradients, used for adjusting the network's parameters during training, become too small as they propagate backwards through time, resulting in diminished learning for distant elements in long sequences. To address this, gating mechanisms (Bengio et al., 1994) are combined with RNN to enhance the control of information. One of the earliest methods of gating is the Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), which can be regarded as a specialized type of RNN architecture. LSTMs are particularly well-suited for tasks involving sequential data where retaining context over longer periods is essential. They achieve this by incorporating a more sophisticated memory mechanism compared to traditional RNNs. This memory mechanism consists of memory cells (c_t) and three gating mechanisms: the forget gate (f_t), input gate (i_t), and output gate (o_t), as shown in Figure 2.3. The computational steps of LSTM are shown in Equations 2.4-2.9:

$$i_t = \sigma(W_{xi}x_t + b_{xi} + W_{hi}h_{t-1} + b_{hi}) \tag{2.4}$$

$$f_t = \sigma(W_{xf}x_t + b_{xf} + W_{hf}h_{t-1} + b_{hf}) \tag{2.5}$$

$$o_t = \sigma(W_{xo}x_t + b_{xo} + W_{ho}h_{t-1} + b_{ho}) \tag{2.6}$$

$$\tilde{c}_t = \tanh(W_{x\tilde{c}}x_t + b_{x\tilde{c}} + W_{h\tilde{c}}h_{t-1} + b_{hc}) \tag{2.7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.9}$$

where h_t is hidden state, x_t is the current input to compute an input gate value between 0 and 1, $\tilde{c}_t$ represents candidate cell state that calculates a candidate value for the new content that could be added to the memory cell, all the Ws and bs with different subscripts represent weights and biases respectively, $\odot$ is Hadamard Product (Neudecker et al., 1995) .

In summary, LSTMs are adept at capturing long-range dependencies in sequential data due to their ability to selectively remember, update, and output information from memory cells. This makes them highly effective for applications where predicting patterns over extended sequences are crucial.

2.1.2.3 Gated Recurrent Unit

Gated Recurrent Unit (GRU) is another type of recurrent neural network architecture that addresses the limitations of traditional RNNs in capturing long-range dependencies while also simplifying the structure compared to LSTM networks.

Introduced by Cho et al. (2014), GRU's architecture allows it to adaptively control the flow of information based on the context of the sequence. It simplifies LSTMs' architecture by merging the memory cell and hidden state, and by using only two gating mechanisms instead of three. This leads to faster training and potentially fewer parameters.

The update gate (z) determines how much of the previous hidden state should be combined with the candidate activation (proposed new activation). It decides which information to carry forward to the next time step and which information to update. It takes into account the previous hidden state and the current input.

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \tag{2.10}$$

The reset gate (r) decides how much of the previous hidden state should be ignored when computing the candidate activation. It controls the amount of past information

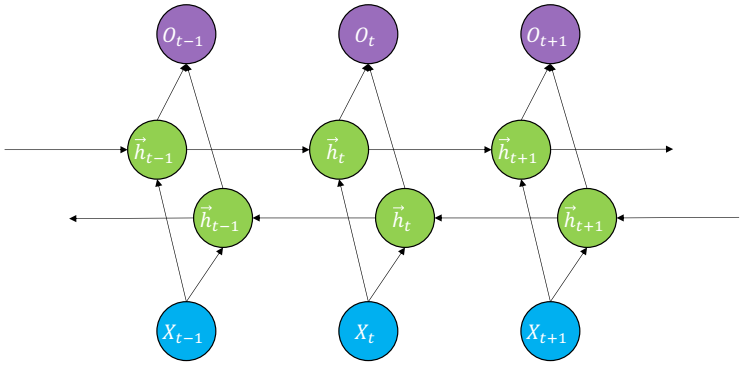


Figure 2.4: Bi-RNN (Vaswani et al., 2017).

that is relevant for the current time step.

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \tag{2.11}$$

Candidate activation ($\tilde{h}$) is the proposed new activation for the current time step. It considers the reset gate and the current input, producing a candidate that might be added to the hidden state.

$$\tilde{h}_t = \sigma(W_{hx}x_t + W_{hr}(r_t \odot h_{t-1}) + b_h) \tag{2.12}$$

The hidden state (h_t) represents the memory of the network at the current time step. It is a combination of the previous hidden state, the candidate activation, and the update gate.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{2.13}$$

2.1.2.4 Bi-directionality

RNN-based models can only predict the output of the next time step based on the timing information of the previous time steps. However, in some tasks, the output of the current time step is not only related to the past state, but also to the future state. For example, predicting a missing word in a sentence not only needs the previous text, but also the content behind it. To solve this problem, bidirectional recurrent neural networks (BiRNN) are introduced by (Schuster and Paliwal, 1997). As shown in Figure 2.4, BiRNN consists of two stacked RNNs, one is a forward RNN and another is a backward RNN. The output of BiRNN is determined by the states of the two RNNs. In practical

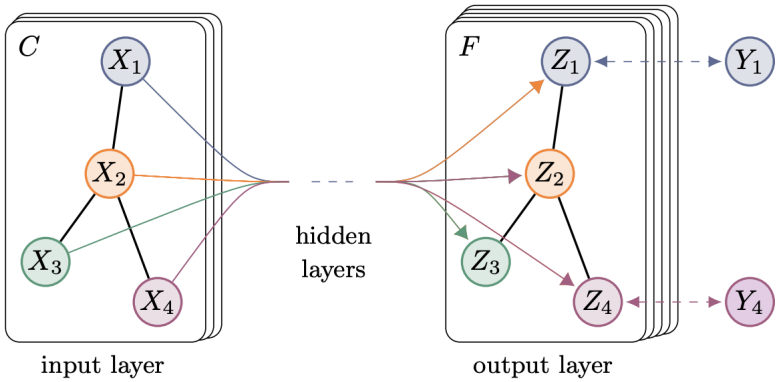


Figure 2.5: Graph Convolutional Network (Kipf and Welling, 2017)

applications, bi-directionality is implemented by employing two separate neural models (such as LSTM and GRU) that process the input sequence in both left-to-right and right-to-left directions. Usually, the results produced by these two networks are combined by concatenation to create a definitive representation for each sequence output, as depicted below:

$$y_t = [\overrightarrow{o}_t; \overleftarrow{o}_t] \tag{2.14}$$

where $\overrightarrow{o}_t$ and $\overleftarrow{o}_t$ correspond to the output of the left-to-right and right-to-left network, respectively; y_t is the final output of the bi-directional network and ";" indicates a concatenation operation.

2.1.3 Graph Convolutional Network (GCN)

Traditional neural networks are designed for grid-like data, such as images or sequences, and cannot naturally handle graph-structured data. Graph Convolutional Networks (Kipf and Welling, 2017), also known as GCN, are proposed to work directly with graph data, making them suitable for tasks like node classification, link prediction, and community detection. GCNs extend the idea of convolutional layers from traditional Convolutional Neural Networks (Kim, 2014) to graphs. The core idea of GCNs is to obtain the feature representation of the node by performing a convolution operation on the graph. Specifically, each node uses convolution operations to obtain information from its surrounding nodes to update its own representation. Since a single GCN layer can only obtain information from its neighbours, in applications, GCNs typically stack multiple layers to capture increasingly complex information from the graph, as shown

in Figure 2.5. As the number of GCN layers increases, the model can combine the node information of higher-order neighbours to obtain richer global information. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes and a feature matrix (X) for these nodes, the message passing process of a GCN layer is shown below:

$$H^{(l+1)} = \sigma(\tilde{D}^{(-1/2)}\tilde{A}\tilde{D}^{(-1/2)}H^{(l)}W^{(l)}) \tag{2.15}$$

$$\tilde{A} = A + I \tag{2.16}$$

$$\tilde{D}_{ii} = \sum j\tilde{A}_{ij} \tag{2.17}$$

where $H^{(l)}$ represents the node representation of the l-th layer and $H^{(0)} = X$, $W^{(l)}$ is a weight matrix for the l-th layer, A is the adjacency matrix, I is the identity matrix, $\tilde{D}$ is the degree matrix of $\tilde{A}$, σ is a non-linear activation function.

Here, $\tilde{A}$ where a self-loop is added to each node in the graph is used instead of A because A_{ii} are all 0, which will cause the model to forget the node's own information and only pay attention to its neighbours when updating the information of a given node. In addition, the normalization operation is usually performed by $\tilde{D}^{(-1/2)}\tilde{A}\tilde{D}^{(-1/2)}$ in Equation 2.15 during the GCN operation, because the number of neighbours nodes and weights are different for different nodes. After the graph convolution operation, the final eigenvalue of the node with many neighbours will be large.

GCNs have a large number of applications, such as node classification, graph classification and link prediction. For node classification, the output of the last layer is often used. For graph classification, the average value or maximum value of all nodes is commonly used.

2.1.4 Attention Mechanisms

Although RNN-based models can deal with natural language effectively, LSTM can only alleviate the long-distance dependency problem in RNN to a certain extent, and its information 'memory' ability is not enough, especially when facing long text. And RNN-based models are time-consuming since the calculation of these types of models needs to be completed serially. To solve these problems, attention mechanisms (Galassi et al., 2021a) are proposed. Attention mechanisms are derived from the phenomenon that when humans observe a picture or read a text, they will first be attracted by a certain part of the picture or some key words of the text and ignore other parts that are not of interest. This mechanism can help humans handle a large amount of information

effectively. Similarly, the core of attention mechanisms is to let the model only pay attention to a small part of important information when a large amount of data is received and processed. Attention mechanisms are widely used in computer vision and natural language processing and have achieved good results.

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, key, value, and output are all vectors. Most current attention functions can be summarized into two processes: the first process is to calculate the weight coefficient based on the query and key, and the second process is to perform a weighted sum of the value based on the weight coefficient. The first process can be subdivided into two stages: the first stage calculates the attention score matrix $A = [a_{ij}]$ through a score function between the query vector q and key vector k; the second stage normalizes the attention score matrix of the first stage by a softmax function. Here, the normalized result is the attention distribution of the query vector on each element of the key vector. The calculation formula of the first stage is as follows:

$$a_{ij} = softmax(s(q_i, k_j)) = \frac{exp(s(q_i, k_j))}{\sum_{j=1}^{n_k} exp(s(q_i, k_j))} \tag{2.18}$$

where q_i is the i-th element in q and k_j it the k-th element in k, s is a score function used to calculate the attention scores between q and k. There are many different methods for the score function. We will introduce some of the most important ones in this section later.

After the second stage, the representations of the context c can be obtained:

$$c_i = \sum_j a_{ij} v_j \tag{2.19}$$

where c_i is the i-th element of the context vector c. c_i contains information about the token in location i and the contextual information by combining queries, keys and values. In this way, the attention mechanism allows the model to dynamically focus on and select relevant information in the context, thereby improving the performance and expressiveness of the model.

Next, we will introduce the common used score function methods, such as the additive attention as well as dot-product attention.

Additive Attention. The additive attention is one of the first attention mechanisms proposed by Bahdanau et al. (2015). For each query vector, the additive attention calculates the similarity between each query and key using additive operations. This means that for each query vector q and key vector k, the score function s is shown

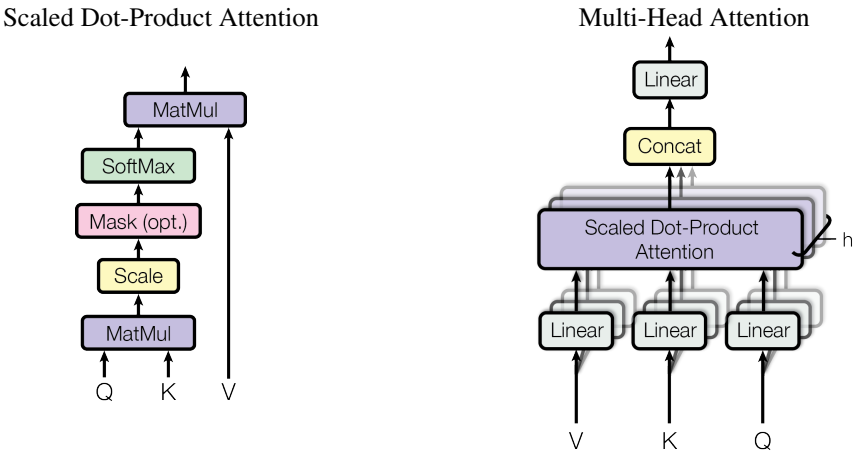


Figure 2.6: (left) Scaled Dot-Product Attention. (right) Multi-head attention consists of several attention layers running in parallel (Vaswani et al., 2017).

below:

$$s(q_i, k_j) = W_1 q_i + W_2 k_j \tag{2.20}$$

where W_1 and W_2 are learnable parameters.

Dot-Product Attention. For each query vector, calculate the dot-product (inner product) between it and all key vectors. The dot product operation is a way of calculating similarity. Typically, for each query vector q and key vector k, their attention score e is calculated as follows:

$$s(q_i, k_j) = q_i k_j \tag{2.21}$$

In addition, when the dimension of the input vector is relatively high, the dot product model usually has a relatively large variance, resulting in a relatively small gradient of the Softmax function. Therefore, the scaled dot product model smoothes the fractional value by dividing by a square root term, which is also equivalent to smoothing the final attention distribution, alleviating this problem.

$$s(q_i, k_j) = \frac{q_i k_j}{\sqrt{D}} \tag{2.22}$$

Self-attention. Previous attention mechanisms are often used to handle relationships between sequences, where queries and keys come from different sequences. Self-attention mechanisms (Vaswani et al., 2017), on the other hand, are used to handle relationships between elements within a single sequence, where the query, key, and value all come

from the same sequence, which allows the model to capture the interrelationships between information at different positions in the sequence to better leverage the context and generate meaningful representations. Therefore, the self-attention mechanism is widely used in natural language processing tasks. It is worth mentioning that the self-attention mechanism does not propose a new score function, but a method of applying the attention mechanism to a single sequence. Therefore, both additive attention and dot product attention can be used as score functions.

Multi-head Attention. Multi-head attention is an extended and improved version of the self-attention mechanism proposed by Vaswani et al. (2017), which allows the model to simultaneously focus on different subspaces or feature representations in the input. The basic idea of the multi-head attention mechanism is that different attention heads can learn to capture different relationships and features of the input. By computing multiple attention heads in parallel, the model can obtain a more comprehensive representation, thereby improving performance.

2.1.5 The Transformer

As the majority of previous sequence-to-sequence models, the Transformer (Vaswani et al., 2017) follows the encoder-decoder structure. In the encoder-decoder structure, given a input sequence $x = (x_1, ..., x_n)$, an encoder maps it to a sequence of continuous representations $z = (z_1, ..., z_n)$ and a decoder is used to generate an output sequence $y = (y_1, ..., y_m)$ based on z.

As for the Transformer shown in Figure 2.7, the left part is the encoder of the Transformer and the right part is the decoder of it. From Figure 2.7, it is clear that the encoder of the Transformer stacks N encoder layers and the decoder also has N decoder layers. Here, N is six.

The encoder layer contains two parts, a multi-head attention sub-layer and a position-wise fully connected feed-forward network. The latter sub-layer is performed independently and identically for each position in the sequence. It involves two linear transformations separated by a ReLU activation function.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2.23}$$

In addition, a residual connection (He et al., 2016) is applied around each of the two sub-layers, followed by the application of layer normalization (Ba et al., 2016). The structure of the decoder layer is quite similar to that of the encoder layer, with a

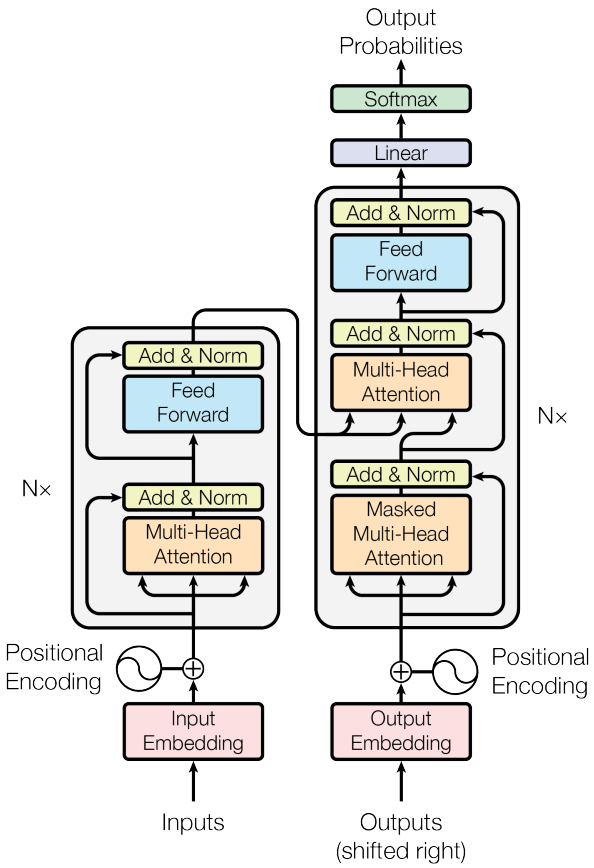


Figure 2.7: The Transformer - model architecture (Vaswani et al., 2017).

main difference part named the masked multi-head attention sub-layer. It introduces a masking mechanism based on the multi-head attention to limit the model to only consider previous positions when calculating attention weights. To be specific, the decoder for token position t should not have access to token position $t + 1$. This is accomplished before the Softmax stage by adding a mask matrix M that is negative infinity at entries where the attention link must be cut, and zero at other places. This mask mechanism is used to ensure that the model does not use future information when generating predictions to avoid information leakage.

One problem of the Transformer structure is that it does not have sequence order information because the self-attention mechanism does not care about the order of input. In order for the Transformer to process sequence data (such as natural language text) and capture their order information, positional encoding is introduced. As shown in Figure 2.7, it is clear that besides the input sequence x, the Transformer also adds a positional encoding at the bottoms of the encoder and decoder stacks. The positional

encodings are generated by a combination of sine and cosine functions. The formulas of the positional encoding are shown in Equation 2.24 and 2.25:

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}}) \tag{2.24}$$

$$PE(pos, 2i) = cos(pos/10000^{2i/d_{model}}) \tag{2.25}$$

where pos is the position and i is the dimension. This encoding allows the model to capture information with different frequencies at different locations to distinguish the relative positions between words.

2.1.6 Pre-trained Language Model

Due to its excellent performance, pre-trained language models are widely used in many NLP tasks. In this section, we will introduce the pre-train language models. Based on the Transformer structure, there are mainly three types of pre-trained language model structures, encoder-only models such as BERT, decoder-only models such as GPT and encoder-decoder models like BART. As the decoder-only models are not used in this thesis, we only introduce the other two types of pre-trained language models in the following part of this section.

2.1.6.1 BERT

BERT, also known as Bidirectional Encoder Representation from Transformers, is a pre-trained language model based on the Transformer structure. As mentioned in Section 2.1.5, BERT only uses the encoder part of the Transformer and abandons the decoder part. Therefore, it processes input text in a bidirectional manner (i.e. left to right and right to left), which helps BERT achieve more accurate language representation by comprehensively considering information in context. In addition, because the Transformer architecture allows BERT to perform parallel computations at all positions in the sentence, it also overcomes the disadvantages of RNNs that rely on the order of the sequence, resulting in computational inefficiency.

As a language model, the first step for BERT is to process the input text. An example is shown in Figure 2.8. The input text is first segmented into discrete tokens, usually words or subwords. BERT uses a word segmentation method called WordPiece, which divides text into multiple subwords or roots to capture more lexical information. In addition to the tokens in the input text, there are usually two special tokens ([SEP]

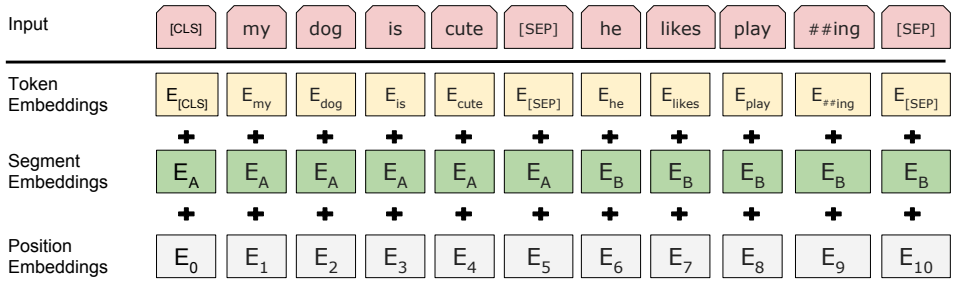


Figure 2.8: BERT input representation. Taken from (Devlin et al., 2019).

and [CLS]) added into the sequence of tokens. The [SEP] token is used to distinguish sentences while the [CLS] token is usually the first token to represent the whole input text in case of the text classification task.Then each token is mapped to a high-dimensional token embedding vector. Besides the token embeddings, the input of BERT also contains other two parts, position embeddings and segment embeddings. The former allows the model to identify the position of each token in the text, and the latter helps the model distinguish segment A from segment B when the task input is a sentence pair, i.e., the ARI and ARC subtasks. Then, the concatenation of these three embeddings is used as the final input of BERT.

Next, we will introduce the pre-training step of BERT, where two pre-training tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP) are conducted. MLM is a token-level task that aims to allow BERT to learn the relationship between words and the structure of the context. In this task, the token in each training sequence is randomly replaced with a mask token ([MASK]) with a probability of 15%, and then BERT needs to predict the original token at the [MASK] position. In contrast, the purpose behind NSP is to learn the relations at sentence-level. Specifically, BERT accepts a pair of sentences as input and predict whether the two sentences are semantically consecutive.

The performance of the BERT model depends largely on the text used in pre-training. Therefore, in order to allow the model to better handle tasks in specific domains, some models trained with text in specific fields have been proposed. For example, BioBERT (Lee et al., 2020) for the biomedical domain and LEGAL-BERT (Chalkidis et al., 2020) for the legal domain.

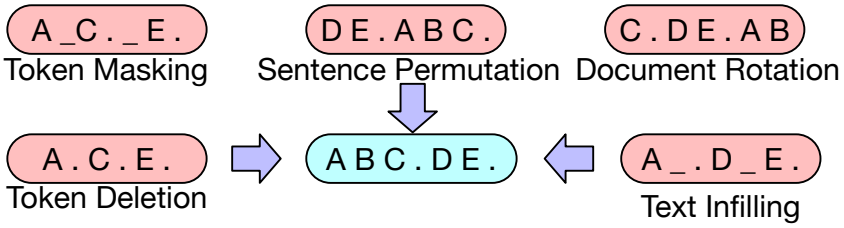


Figure 2.9: Transformations for noising the input in the pre-training step of BART. Taken from Lewis et al. (2020).

2.1.6.2 BART

BERT has achieved brilliant results in many NLP tasks. However, because it only uses the encoder in the Transformer structure, it is not suitable for solving text generation tasks. Therefore, BART (Bidirectional Auto-Regressive Transformers (Lewis et al., 2020)), a neural network model that consists of a bidirectional encoder as well as an autoregressive decoder is proposed. The training process of BART is to reconstruct the original text given a corrupted text as input and then optimise a reconstruction loss between the reconstructed text and the original text. There are mainly five noising schemes for document corruption, as shown in Figure 2.9. The details are introduced as follows:

Token Masking. In line with BERT, random tokens are selected and substituted with a special token ([MASK]).

Token Deletion. Random tokens are deleted. Different from token masking, the model also needs to predict the positions of the missing tokens.

Text Infilling. Several text spans are randomly selected, and the lengths of these spans are generated from a Poisson distribution. Each of these spans is substituted with a single [MASK] token. Spans with a length of 0 signify the insertion of [MASK] tokens. The text infilling task instructs the model to estimate the number of tokens that are absent from a given span.

Sentence Permutation. A document is segmented into sentences according to full stops, and then these sentences are rearranged randomly.

Document Rotation. A token is selected in a uniformly random manner, and the document is rotated so that the chosen token is the first token of the document. This task instructs the model to recognize the document's starting point.

Same as BERT, BART can also be used directly or after fine-tuned. And there is also a BioBART (Yuan et al., 2022) for biomedical NLP tasks.

2.2 Evaluation Metrics

As mentioned in Section 2.3.1, the argument mining task consists of four subtasks, and depending on the granularity of the argument mining task, it can be divided into two types: token-level and sentence-level. This leads to a variation in the task modelling methods, such as sequence labelling, span classification, sentence classification, or span (sentence) pair classification. However, all four subtasks can be considered as classification tasks. Therefore, like other NLP tasks, the evaluation metric widely used is based on the F1-Score. In this section, we first introduce the formula for the F1-Score.

We introduce the evaluation metrics with a binary classification task. In a binary classification task, the label of each instance belongs to one of two categories (here we use positive or negative to represent these two categories). According to the relationship between the model's prediction result for an instance and the true label of the instance, the following four statistics are defined: *True Positives (TP)* indicate the number of positive examples that the model correctly predicted as positive; *True Negatives (TN)* indicate the number of negative examples that the model correctly predicted as negative examples; *False Positives (FP)* indicate the number of negative examples that the model incorrectly predicted as positive examples; *False Negatives (FN)* indicate the number of positive examples that the model incorrectly predicted as negative examples. A confusion matrix of binary classification based on these four statistics is shown in Table 2.1, which provides the basis for defining commonly used metrics such as precision, recall, accuracy, and F1-score.

		Ground truth	
		Positive	Negative
Prediction	Positive	**TP**	**FP**
	Negative	**FN**	**TN**

Table 2.1: Confusion matrix of binary classification.

Accuracy. Accuracy (ACC) is the most straightforward indicator for measuring classification models, it represents the proportion of correctly classified instances to the total number of instances.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.26}$$

Precision & Recall. Precision (P) indicates the proportion of actual positive samples among the samples whose predicted results are positive; conversely, Recall (R) indicates

the proportion of actual positive samples among the samples whose predicted results are positive to the total number of positive samples.

$$P = \frac{TP}{TP + FP} \quad (2.27) \qquad R = \frac{TP}{TP + FN} \quad (2.28)$$

F1-score. F1 score is a weighted average of precision and recall, which can be regarded as a special situation of F_β score ($\beta = 1$), The calculation formula of F_β score is as follows:

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \quad (2.29)$$

Micro-, Macro- and Weighted-averaged Metrics. The evaluation matrix mentioned earlier only considers the binary classification task. Now we need to extend it to the multi-class classification task. Consider a three-class classification task for the ARIC subtask. Given a pair of ACs, the model outputs one of three labels (support, attack, none). The confusion matrix for this three-class classification task is shown in Table 2.2. It can be seen that the main difference between the multi-class and binary classification tasks is how to calculate the errors. For instance, when the model predicts the label of a pair of ACs is "attack" but the true label is "support", we need to count two types of errors: a FP for category "attack" due to misclassification and a FN for category "support" due to prediction failure. In situations where the prediction is labelled as "None", it is common to disregard FP errors and solely assess the FN errors for each overlooked relation category. Likewise, when the true label is "None", only FP errors are taken into account for each incorrectly predicted category.

		Ground Truth		
		Support	Attack	None
Prediction	Support	**TP**(S)	**FP** (S) & **FN** (A)	**FP** (S)
	Attack	**FP** (A) & **FN** (S)	**TP**(A)	**FP**(A)
	None	**FN**(S)	**FN**(A)	**TP**(N)

Table 2.2: Confusion matrix of multi-class classification. Here, S, A, N represent Support, Attack and None respectively.

There are many different calculation methods for F1-Score for multi-classification tasks, such as micro-, macro- and weighted-averaged F1-Scores. Since the main

difference between these methods is the calculation of Precision and Recall, we mainly introduce these differences below.

After obtaining Precision and Recall, the final F1-Score is calculated by Equation 2.29. Micro-averaged metrics for Precision and Recall are calculated globally by counting the total TP, FN and FP, as shown below, where C is the total number of categories:

$$P_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i} \tag{2.30}$$

$$R_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FN_i} \tag{2.31}$$

Macro- and weighted-averaged metrics are both weighted averages of each category. Therefore, the equations used to calculate P and R for both metrics can be uniformly expressed as:

$$P_\alpha = \sum_{i=1}^{C} \alpha_i P_i \tag{2.32}$$

$$R_\alpha = \sum_{i=1}^{C} \alpha_i R_i \tag{2.33}$$

The difference is that the weights of macro-averaged metrics are all the same (the reciprocal of the number of classes):

$$\alpha_i^{macro} = \frac{1}{C} \tag{2.34}$$

while the weights of weighted-averaged metrics are based on the distribution of instance numbers in each category.

$$\alpha_i^{weighted} = \frac{support_i}{\sum_{i=1}^{C} support_i} \tag{2.35}$$

2.3 Argument mining: An Overview

2.3.1 Task Definition

Argument mining (also known as argumentation mining) is a task that aims at analysing the argument structure of discourse. Differing from the majority of other NLP challenges, it is not a single and well-demarcated task but it constitutes a collection of

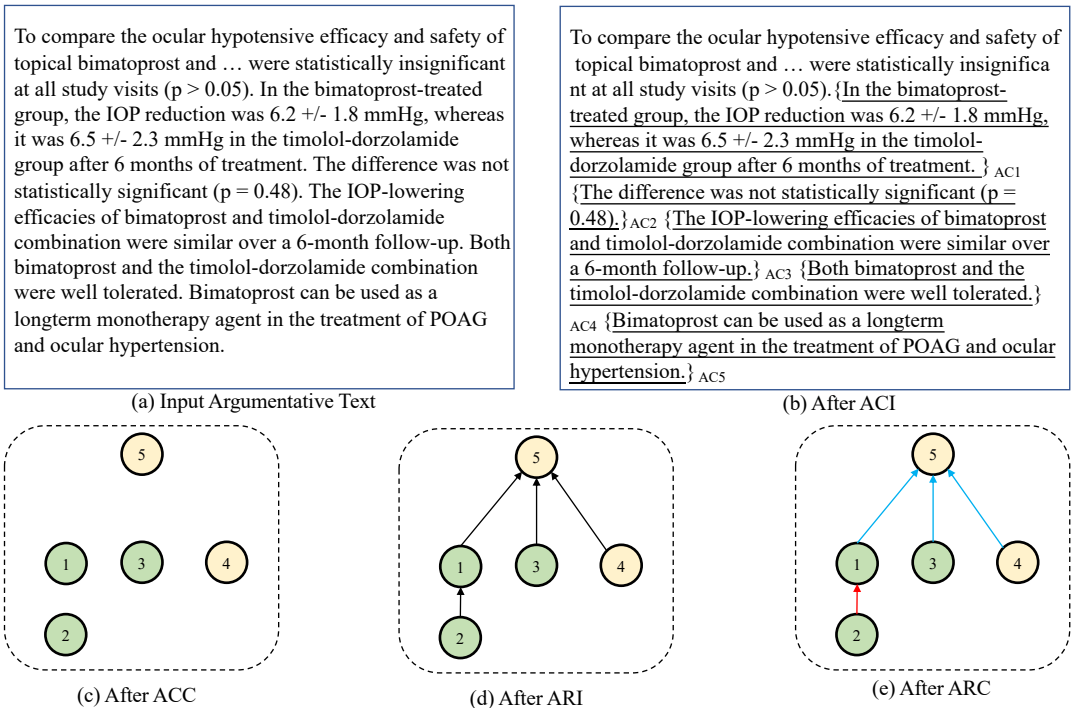


Figure 2.10: An example of the pipeline of the four AM subtasks. The green nodes represent *premises*, the yellow nodes are *claims*, the black arrows denote the existence of an AR between two ACs, and the blue and red arrows respectively represent that the type of AR is *support* and *attack*.

subtasks, the combinations of which can be utilised for particular applications. Here we choose the most popular definition from Eger et al. (2017) where the task can be divided into four subtasks: (1) **Argument Component Identification (ACI)**: separating argumentative units from non-argumentative units. (2) **Argument Component Classification (ACC)**: classifying argument components (AC) into specific types. (3) **Argument Relation Identification (ARI)**: determining which ACs are related to each other, and the direction of these argument relations (AR).(4) (o_t): labelling the type of each AR.

An example is shown in Figure 2.10. Given the input argumentative text shown in subfigure (a), the ACI subtask aims to separate argumentative units from non-argumentative content. In subfigure (b), there are five argumentative units that are underlined. After the ACC subtask, these five argumentative units are classified into claims (yellow nodes in subfigure (c)) and premises (green nodes in subfigure (c)). After the ARI subtask, four argument relations are identified as shown in subfigure (d). And finally, after the ARC subtask, these four argument relations are classified into support

(blue arrows in subfigure (e)) and attack (red arrows in subfigure (e)) classes.

2.3.2 Task Classification

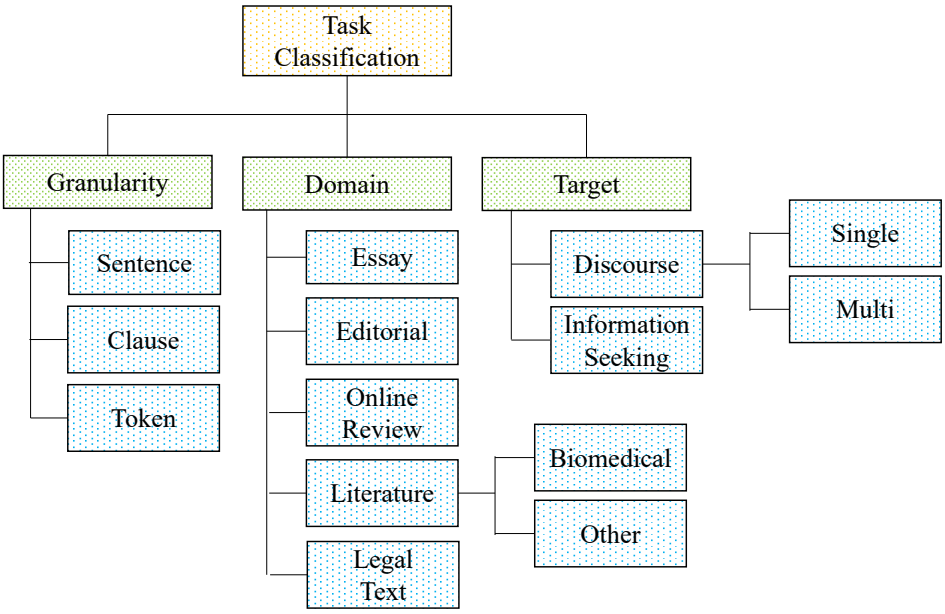


Figure 2.11: Categories of the argument mining task on different aspects.

There exist two main categories of AM tasks based on the target (Stab et al., 2018), as shown in Figure 2.11. The first category is discourse-level argument mining, which entails the model extracting the structure from provided discourses. This category can be further subdivided based on the number of discourses involved. Single discourse argument mining (Ajjour et al., 2017) focuses on extracting arguments from individual discourses, whereas multi-discourse argument mining (Cheng et al., 2021; Sun et al., 2023) deals with extracting arguments from multiple interconnected discourses, often observed in online discussions where users engage in debates through posts. The second category is the information-seeking argument mining task (Trautmann, 2020; Cheng et al., 2022), which essentially functions as an information retrieval task. In this task, the model is tasked with locating all arguments pertinent to a given contentious topic and determining the stance of each argument towards the topic.

In this thesis, we only pay attention to the single discourse-level argument mining task, as it is the foundation of understanding an argument, and the multi-discourse and information-seeking argument mining tasks can be regarded as an extension of this task.

Another differentiation can be established by considering target domains. Initially, methods for extracting relationships were primarily applied to the legal domain (Mochales and Moens, 2011), Gradually, researchers became interested in argument mining in other fields, such as student essays (Eger et al., 2017), scientific literature (Accuosto et al., 2021), online reviews (Niculae et al., 2017), editorials (Al-Khatib et al., 2017). Less attention was paid to the biomedical domain.

Additionally, argument mining can be categorised into sentence-level, clause-level, and token-level tasks depending on the granularity of the annotation schemes. In sentence-level and clause-level argument mining, the boundary of an argument component aligns with either a sentence or a clause, respectively. In contrast, token-level argument mining allows for the length of argument components to vary, spanning from less than a clause to several sentences. However, due to the limited availability of clause-level datasets, we will primarily concentrate on sentence-level and token-level tasks in the subsequent sections, disregarding clause-level annotation for the time being.

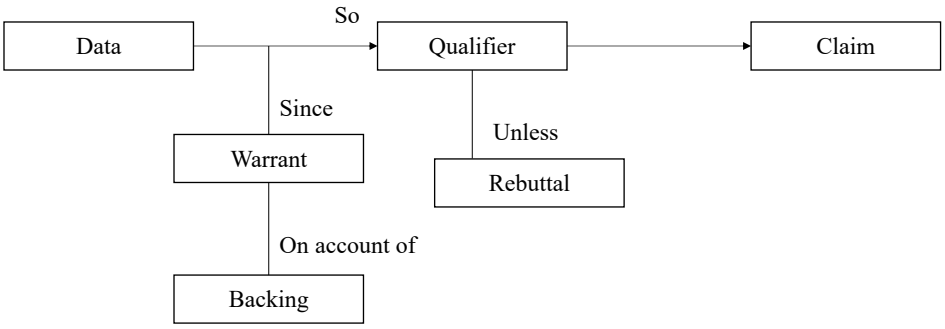
2.3.3 Argumentation Models and Datasets

In this section, we introduce some widely used AM datasets. The representation of argument structures in different datasets varies depending on the specific requirements of each application (Accuosto, 2021). Therefore, before discussing the datasets, we first present some argumentation models that serve as the foundation for many annotation schemes.

2.3.3.1 Argumentation models

Argumentation models can be divided into two broad categories: abstract argumentation and structured argumentation (Lippi and Torroni, 2015). The former treats each argument as an atomic unit, without concerning itself with its internal structure, while the latter proposes an internal structure for each argument. Since argument mining requires consideration of both argument components and their relationships, the majority of annotation schemes are based on structured argumentation models. Some influential structured argumentation models include the following:

- **Toulmin Model**: Toulmin model (Toulmin, 1958) is a classic argumentation model used to describe the inner structure of an argument. It is a well-formed argument structure that contains six different necessary parts: (1) *Claim*: The central statement or viewpoint of an argument. (2) *Data*: Specific facts, examples,



{Harry was born in Bermuda.}$_{\text{Data}}$ *Since* {A man born in Bermuda will generally be a British subject.}$_{\text{Warrant}}$ *On account of* {The following statuses and other legal provisions: (...)}$_{\text{Backing}}$ *So*, {presumably}$_{\text{Qualifier}}$ *Unless* {Both his parents were aliens}$_{\text{Rebuttal}}$ {Harry is a British subject.}$_{\text{Claim}}$

Figure 2.12: An example of Toulmin model (Toulmin, 1958).

or information used to support the claim. (3) *Warrant*: The reasoning or logical step that connects the claim and the data. (4) *Backing*: Additional support or explanatory information to strengthen the warrant in the argument.(5) *Rebuttal*: Viewpoints or opposing opinions that may challenge the argument. (6) *Qualifier*: Words used to express the degree, applicability, or conditions of the claim, adjusting the strength or scope of the claim. An example of Toulmin model of an argument is shown in Figure 2.12.

- **Freeman Model**: Freeman (2011) introduces a theory outlining the structure of arguments, which examines a hypothetical dialectical exchange between a proponent defending (supporting) a claim and an opponent questioning (attacking) it. It consists of five elements: premise, conclusion, modality, rebuttal, and counter-rebuttal. Among these, premise and conclusion serve as the fundamental elements, whereas the other three may or may not be present in an argument.

- **Walton Model**: Walton model Walton et al. (2008) assumes that an argument comprises a set of statements (propositions), consisting of three components: a conclusion, a set of premises, and an inference from the premises to the conclusion. The model includes various templates (argumentation schemes) designed to capture typical structures of arguments employed in everyday reasoning. These schemes serve the dual purpose of guiding and evaluating the reasoning/argumentation process.

While there are several argumentation models available, directly applying these models for annotation may not be applicable in all cases because the representation of argument structures often varies based on specific needs and domains, as we mentioned earlier. For instance, Habernal and Gurevych (2017) try to annotate the argument structures of user-generated documents based on Toulmin model to test its suitability. They find that in most cases Qualifiers and Warrants are not included in the documents. Moreover, they also find that users often attack the presented rebuttals by offering counter-rebuttals to maintain the overall consistency of the argument.

Therefore, in practical applications, some annotation schemes are modified versions of existing argumentation models. For example, an annotation scheme based on Freeman model is proposed by Peldszus and Stede (2013). In this scheme, each argument structure is represented as a graph where nodes denote ACs, with one AC identified as the central claim. ARs can be established among ACs or between an AC and another AR. The model enables the representation of scenarios in which two (or more) ACs participate in a joint support AR with another one.

There are also some researchers who design their own schemes based on the needs of downstream applications. For example, in the legal domain, Habernal et al. (2023) propose an own designed annotation scheme in which 16 AC types are included to enable a more thorough analysis of the Court's motivational itinerary.

2.3.3.2 Datasets

Dataset	Gran.	#Instances	Level	#ACT	#ART	Scheme
BioScheme	sen	4	full-text	2	-	adopted Walton
CF	sen	29	full-text	5	-	own
PCRC	sen	259	abstract	2	-	own
SciARK	sen	300	abstract	2	-	adopted Toulmin
BioClaim	sen	1500	abstract	1	-	claim
AbstRCT	token	659	abstract	3	2	adopted Freeman
SciARG	sen	285	abstract	11	6	own

Table 2.3: Argument mining dataset in the biomedical domain. Here, "Gran" means granularity. #ACT and #ART represent the number of argument component type and argument relation type respectively.

In this section, we introduce argument mining datasets in the biomedical domain, which is shown in Table 2.3. There are several conclusions that can be drawn from this table.

Firstly, most of the datasets only contain the annotations of ACs. For instance, Achakulvisut et al. (2019) proposed a dataset called BioClaim that contains 1500 biomedical abstracts indicating whether the sentence presents a scientific *claim*. Alamri and Stevenson (2016) further divided claims in cardiovascular research abstracts into two subcategories based on the characteristics of claims in biomedical literature, namely *evaluative claims* and *causal claims* when annotating the PCRC dataset. The former refers to the author's expression of judgment on the value of a biomedical concept, and the latter indicates a connection between two concepts and states that one concept exerts influence on the other. Other datasets also include more AC types besides the claim. Blake (2010) introduced a new annotation scheme named Claim Framework that contains five AC types (*explicit claims*, *implicit claims*, *correlations*, *comparisons*, and *observations*) and used it to annotate the CF dataset that consists of 29 full-texts of biomedical literature. In SciARK, Fergadis et al. (2021) annotated 300 biomedical abstracts about good health and well-being with two types of ACs, i.e., *claim* and *evidence*. Similarly, BioScheme Green (2015a) used *claim* and *premise* as the two AC types. Here, the *premise* and *evidence* are two similar concepts. However, only two datasets (RCT (Mayer et al., 2020) and SciARG (Accuosto et al., 2021)) contain both the AC and AR annotation. Compared with the RCT dataset which has three types of ACs (*major claim*, *claim* and *premise*) and two types of ARs (*support* and *attack*), SciARG is annotated with a fine-grained scheme in which the number of categories of AC is 11 (*proposal*, *proposal-implementation*, *observation*, *result*, *result-means*, *conclusion*, *means*, *motivation-problem*, *motivation-hypothesis*, *motivation-background*, *information-additional*) and the number of categories of AR is 6 (*support*, *elaboration*, *by-means*, *info-required*, *sequence*, *info-optional*).

Secondly, it is obvious that only two datasets focus on full-text annotation (BioScheme and CF) and the number of instances in these two datasets are small as annotating the full biomedical literature is a very time-consuming job. Therefore, most of datasets only focus on abstracts so that the size could be larger.

Finally, from the perspective of the annotation scheme, only the own-designed and adopted argumentation model-based schemes are used for annotation. This indicates that the argument structure in biomedical literature does not fit perfectly with existing argumentation models, which may lead to the difficulty of argument mining in biomedical literature.

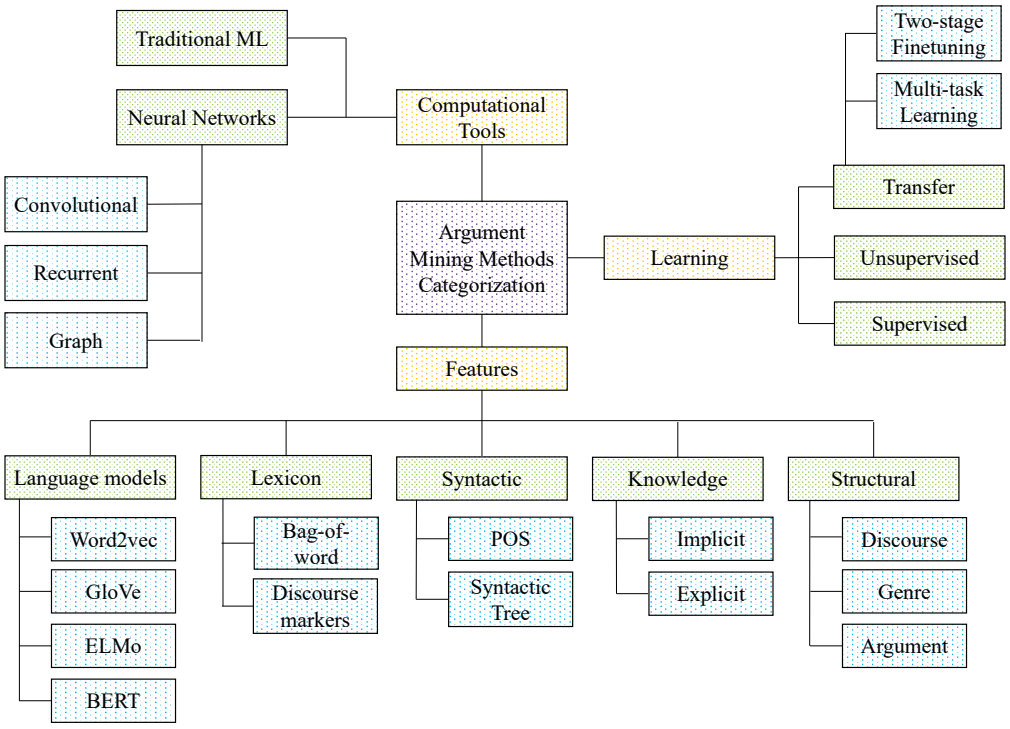


Figure 2.13: Taxonomy of AM approaches.

2.3.4 Structural Features in Argument Mining Models

Generally, an essential feature in NLP tasks is text embedding, as it represents one of the most general methods for vectorising text information. However, due to the complexity of the argumentation task, many argumentation models use not only textual embeddings, but many additional features. There are mainly four types of features used in previous argument mining models as shown in Figure 2.13: lexicon features (including bag-of-word (Potash et al., 2017; Kuribayashi et al., 2019; Bao et al., 2021), discourse markers (Eckle-Kohler et al., 2015; Burhan ud Din Tahir, 2017; Du et al., 2017; Aker et al., 2017; Achmadeeva et al., 2019; Gemechu and Reed, 2019; Opitz and Frank, 2019; Dutta et al., 2022; Melie et al., 2023)), syntactic features (Morio and Fujita, 2019; Mayer et al., 2018; Wan et al., 2022; Mushtaq and Cabessa, 2022), knowledge features (such as implicit knowledge (Zhuang et al., 2020; Rodrigues and Branco, 2022; Morio et al., 2022) and explicit knowledge (Gemechu and Reed, 2019; Alhamzeh et al., 2021; Stylianou and Vlahavas, 2021; Fromm et al., 2019; Saint-Dizier, 2017; Paul et al., 2020; Kobbe et al., 2019; Li et al., 2021; Saadat-Yazdi et al., 2023)) and structural features.

Considering that the main focus of this thesis is to use structural information, we only illustrate how existing models harness structural features in the rest part of this section. Given the limited research on argument mining from biomedical text, the methods we discuss in this section are based on argument mining from monolingual text.

Since argument mining itself is the extraction of text structure information, many previous works have explored the help of other structural information in argument structure. Broadly, two categories of structural information are employed for the argument mining task: the discourse structure and the features of a specific genre. In this section, we will introduce the discourse structure first.

Discourse Structure. Discourse structure refers to the organisation and arrangement of a written or spoken expression, and the way a speaker uses it to convey information, ideas, or stories between different parts. It involves how language elements (such as sentences and paragraphs) are organised to form a coherent and logical whole. One way to describe the discourse structure is to connect discourse units (sentences, clauses or nominalisations (Webber et al., 2012)) with discourse relations. Given that discourse units can serve as ACs, and some discourse relations are highly related to argument relations, many researchers are committed to exploring the correlation between discourse and argument relations (Green, 2010, 2015b; Galitsky et al., 2018; Kononenko et al., 2020).

Early works aimed at exploring the argumentative nature of discourse relations. For example, an early study (Azar, 1999) defined five discourse relations of Rhetorical Structure Theory (RST), namely Antithesis, Concession, Evidence, Justify and Motivation as argument relations. He argued that such five discourse relations could be used to determine four types of arguments (Evidence for a supportive argument, Justify for a justifier argument, Motivation for an incentive argument and, Antithesis and Concession for persuader arguments). Villalba and Saint-Dizier (2012) conducted a further examination into the argumentativeness of rhetorical relations. Through a comprehensive analysis of online textual reviews, the study revealed patterns in presenting persuasive arguments in the support function via specific cases of rhetorical relations such as ELABORATION, JUSTIFICATION, RESTATEMENT, and COMPARISON. Additionally, it illustrated the expression of argumentative elements in the attack function through the rhetorical relation CONTRAST.

In contrast, certain researchers argue that relying solely on discourse relations to represent argument relations may be insufficient. Stab et al. (2014) observed that

in scientific literature and student essays, discourse relations failed to encompass all discourse relationships, and such relations were limited to adjacent text units. Similarly, Green (2015a) demonstrated that these two types of relations serve distinct purposes and that each did not fully encompass the other. Nevertheless, they acknowledged that discourse relationships can be beneficial in representing discursive relationships.

Furthermore, some researchers explore whether argument structures can be aligned with discourse structures. The groundwork for this research is laid by the work of Stede et al. (2016) who expands the Microtexts corpus with two additional annotation layers of discourse structure, namely Rhetorical Structure Theory and Segmented Discourse Representation Theory (Asher and Lascarides, 2003). In order to directly compare the two structures, they standardised the segmentation rules of the underlying minimal discourse units. Leveraging this annotated dataset, they first conducted research at the single edge level aiming to study whether there is a consistent mapping between each single relation of the two schemes. Their findings revealed that overall, 60% of the edges are shared between both structures. For instance, the most prevalent class of SUPPORT relations in argumentation primarily corresponds to REASON, albeit with some instances of CAUSE and EVIDENCE edges. However, discourse graph and argumentative graph derived from one text typically exhibit non-isomorphic characteristics, and the numbers of relation types vary across different schemes. Therefore, they also studied mapping relationships at the subgraph level. As for the shared components in both theories, approximately 43% of all 3-node argument subgraphs could be aligned with RST subgraphs, and vice versa, accounting for 46%. The majority of these alignments involved parallel structures, such as having two SUPPORTs for a claim on the argumentation side and two parallel REASONs on the RST side. Utilising the same corpus, Peldszus and Stede (2016) conducted experiments to derive ACs and ARs from RST trees. They compared three different models, namely a transformation model, a structure aligner on the subgraph level and an evidence graph model (Peldszus and Stede, 2015). They found that the graph-based model had a significant improvement in the prediction of ARs and central claims. One data mining method, Redescription Mining (Galbrun and Miettinen, 2012), was also leveraged to align subgraphs extracted from the argument and RST structures (Huber et al., 2019). However, their method failed to take into account whether these subgraphs covered the same textual vertices. Huber et al. (2020) proposed to take advantage of the AOC-Poset structure to understand how the subgraph alignments occur in a small corpus annotated in argument and RST structures. The use of AOC-Posets was to observe and explain how subgraphs coming

from two parallel graph-based views of objects are aligned.

Based on these findings, some models have been proposed to leverage the discourse structure as a type of feature to enhance the performance of the argument mining task. Still on the Microtext dataset, Hewett et al. (2019) tested whether the predicted discourse relations can improve the effect of AM. They used the golden segment boundaries from the annotation as segmentations of element discourse units. Then, they compared multiple previous parsers through manual evaluation of the results, ultimately selecting the one proposed by Feng and Hirst (2014). They added the predicted discourse features as an additional feature to the evidence graph model (Peldszus and Stede, 2015) with a minimum spanning tree decoder. They found that incorporating discourse parser features provides valuable information, especially for the classification of the function and attachment subtasks in the AM task.

In order to investigate whether discourse-level annotation is also beneficial for argument mining from scientific literature, Accuosto and Saggion (2019a) extended additional argument structure annotations on a subset (60 abstracts) of a scientific literature dataset (738 abstracts) named SciDTB (Yang and Li, 2018), which already contained RST annotations. They found that regardless of using a CRF or BILSTM model, adding additional RST features could improve the model's performance in the argument mining task. However, this method relies on having golden discourse labels as input to predict the argument structure of an unseen text, which is expensive to obtain in real-world settings. To address this issue, they devised two methods based on transfer learning. The first one was the sequential transfer learning method (Accuosto and Saggion, 2019b). Specifically, they trained an encoder for discourse analysis based on the RST labels, and then used the output of this encoder as an extra features to the input of the argument mining task. The second approach was a multi-task learning framework (Accuosto and Saggion, 2020), where the argument mining task is concurrently learned alongside a discourse parsing task. Based on their experimental results, they discovered that transferring discourse knowledge through representations acquired in discourse parsing tasks can enhance the performance of argument mining models. However, they observed that the multi-task framework dropped the performance on the AM task. They argued that it was due to the large gap in the amount of data between the two tasks, resulting in the regularisation effect introduced in the auxiliary discourse parsing task being too strong.

In the social media domain, Chakrabarty et al. (2019) leveraged the RST features by combining the RST classifier with a fine-tuned BERT model through ensembling. Their

analysis showed that the RST features can serve as an indicator of argument relations, i.e., Antithesis relation in RST for the attack relation in the argument structure.

Genre-specific Structural Features. Many researchers find that some texts of specific genres often have some relatively fixed structures (Igari et al., 2012), which we call the genre-specific structural features in this thesis. It is proved that these genre-specific structural features can play a guiding role in argument mining. One area where this type of structural information is highly exploited is the student essay (Eger et al., 2017). One simple way to use such information is to number the location of each sentence in a text because claims are more likely to appear at the beginning of essays and paragraphs. Song et al. (2020) proposed structural sentence positional embeddings to explicitly encode sentence positions. Specifically, they introduced three types of position embedding, the first one was global position, where the index of a sentence that is used to describe its position in a whole essay. In addition, as an essay has multiple paragraphs, the paragraph number was also used as they assume this type of information is also important. The last position embedding was the local position, which is used to reveal the position of a sentence in its paragraph. From the experimental results, they found that encoding paragraph position and local position largely improves the performance. Not only in the student essays, this kind of position information is also widely used in other types of argumentative text, such as legal texts (Jasim et al., 2019), scientific documents (Accuosto and Saggion, 2019a) and social media (Li et al., 2017).

There are also other ways to leverage the genre-specific structural information of student essays. For example, rather than numbering all the sentences, Potash et al. (2017) only distinguished whether or not an AC is the first AC in a paragraph. In addition, they also added a learnable embedding for each type of paragraph and concatenated the structural embedding with the AC embedding as they found the major claims mainly exist in the first and last paragraphs of an essay. Another way to leverage the paragraph type information was proposed by Bao et al. (2021), who adds one special token for each paragraph type in a BERT tokenizer and the BERT can learn the representation of such tokens during fine-tuning.

Wang et al. (2020) believed that the ACC task on the student essay dataset was not scale-independent because major claims were used to express the point of view of the entire essay, while each claim was used to express the point of view of each paragraph in the essay. Therefore, when extracting major claims, the entire essay was used as BERT's input, while for claims, BERT's input only contained one paragraph. Experimental results showed that using genre-specific structural features through this

multi-scale method can improve the performance of the model.

Argument-specific Structural Features. Due to the design of the annotation schemes, the argument structure of an argumentative text in many datasets (Stab and Gurevych, 2017) is a tree. Therefore, some models have been designed to exploit this argument-specific structural feature to improve the accuracy of argument mining.

The basic idea behind one type of method is to let the model first learn local features and then perform global decoding of the local probability distribution to achieve constraints on the overall structure. For example, Persing and Ng (2016) and Stab and Gurevych (2017) utilised Integer Linear Programming (ILP) over basic argument mining classifiers for the joint prediction of ARs and AC types, incorporating various structural constraints to guarantee the tree structures. In addition, the minimum spanning tree algorithm was leveraged as a decoding mechanism by Peldszus and Stede (2015) so that argument structures extracted by the model are trees. Specifically, they first used an evidence graph model to generate a graph for each argumentative text and then decoded such graphs to trees by the minimum spanning tree algorithm.

Another type of method that helps the model to learn the tree structure features is based on the multi-task learning setup. To be specific, Putra et al. (2021) extended a biaffine attention argument mining model with an auxiliary task that concerns node depth prediction for the tree-structured representation of argument structure. They found that such an auxiliary task is helpful for the ARI subtask.

2.4 Limitations

This section provides an extensive discussion of previous work on argument mining tasks. Based on the above discussion, we found that despite the numerous approaches proposed, a series of limitations remain regarding the exploration of utilising structural features in biomedical abstracts, which this thesis intends to address. First, genre-specific structural features have been shown to be helpful in the AM task on student essays, but in other domains, the role of genre-specific structural features has not been explored. This includes how to model the genre-specific structural features of biomedical abstracts, explore the relationship between these features and argument structures, and how to leverage these features in an AM model. Moreover, previous methods often model the AR-related tasks as a pair classification task, which causes them to only focus on local features and ignore the connections between different ACs associated with the same AC. In fact, it is common that a claim is supported by several

experimental results in the biomedical domain. Finally, the application of argument-specific structural features is limited to the geometric structure formed by the argument (such as a tree), but does not consider from the semantic perspective. We argue that the argument-specific structural features of biomedical abstracts written following implicit rules have certain similarities at the semantic level.

2.5 Summary

In this chapter, we introduced two parts, namely technical background and an overview of argument mining. The former is the foundation in the following chapters. Briefly, in Chapter 3, we used FFN, BiGRU, BERT and the multi-head attention mechanism to exploit the genre-specific structural features. In Chapter 4, the model we proposed includes multiple neural network components, such as FFN, BERT as well as GCN. Our model in Chapter 5 is mainly based on BART. We will introduce these models in detail in the corresponding chapters. In the second part, we reviewed the definition of AM problems and task classification methods. Then we explored existing models that leveraging structural features. Further, we described commonly used argumentation models. Finally we conclude the limitations of previous models. This review is helpful for understanding the advance and limitations of existing argument mining models and therefore inspire our further research in Chapter 3, Chapter 4 and Chapter 5.

Chapter 3

Argument Mining with Genre-specific Structural Features of Biomedical Literature

In this chapter, we address our first research question (**RQ1**), as outlined in Chapter 1, focusing on the relationship between genre-specific structural features of biomedical literature [1] and argument components. Initially, we adopt a biomedical text zoning scheme to represent the genre-specific structural features. The objective of text zoning is to segment a text into zones (i.e., Background, Conclusion) that serve distinct functions. Argumentative zoning, a specialised text zoning framework tailored for the scientific domain, is widely considered as a precursor to argument mining by many researchers. Surprisingly, however, little work is concerned with exploiting zoning information to improve the performance of argument mining models, despite the relatedness of the two tasks. To address this gap, we propose the integration of zoning information serving as genre-specific structural features, into argument component identification and classification tasks through the utilisation of two transformer-based models. One model is tailored for the sentence-level argument mining task and the other is tailored for the token-level task. In particular, we add the zoning labels predicted by an off-the-shelf model to the beginning of each sentence, inspired by the convention commonly used biomedical abstracts. Moreover, we employ multi-head attention to transfer the sentence-level zoning information to each token in a sentence. Based on experiment results, we find a significant improvement in F1-scores for both sentence- and token-level tasks. It is worth mentioning that these zoning labels can be obtained with high

[1] In this rest of this thesis, we use genre-specific structural features for short.

accuracy by utilising readily available automated methods. Thus, existing argument mining models can be improved by incorporating zoning information without any additional annotation cost. This chapter is drawn from our published work (Liu et al., 2022).

3.1 Motivation

The majority of previous argument mining models have primarily focused on local features (Mayer et al., 2020; Accuosto et al., 2021), such as individual argument component, for tasks like classifying the type of AC. Consequently, they have often disregarded the influence of genre-specific structural features within documents. This oversight is particularly significant in highly structured genres like biomedical abstracts, where text organisation adheres to implicitly defined rules specific to biomedical literature.

In biomedical abstracts, for instance, the organisation of text is relatively fixed, governed by established conventions (Sollaci and Pereira, 2004; Hopewell et al., 2008). These conventions dictate the placement and arrangement of various sections, such as introduction, methods, results, and conclusions. However, existing AM models tend to overlook the informative value embedded within these genre-specific structural features.

By integrating zoning information into AM tasks, as proposed in our approach, we aim to bridge this gap. Zoning information encapsulates the structural organisation of text, providing valuable context for understanding arguments within their respective genres. Through the incorporation of zoning labels predicted by off-the-shelf models and leveraging multi-head attention mechanisms, our proposed models effectively capture and utilise genre-specific structural features. This departure from solely focusing on local features enables our models to better discern argumentative contents within highly structured genres like biomedical abstracts.

Text zoning aims at segmenting a text into zones (Gnehm, 2018). Here, each zone differs from others and consists of text parts in terms of a particular function. For example, email zoning (Repke and Krestel, 2018) segments an email into five zones including *body*, *header*, *signoff*, *signature* and *greetings*. A job advertisement can be divided into eight zones (i.e., *company description*, *reason of vacancy*...)(Gnehm and Clematide, 2020). As for scientific literature, there are several zoning schemes which have been proposed (Teufel et al., 1999; Liakata et al., 2010; Kim et al., 2011; Dernoncourt and Lee, 2017). Among them, argumentative zoning (Teufel et al., 1999) is considered as the antecedent for argument mining in scientific literature in previous

Background: *We have recently suggested that bolus 5-fluorouracil (5-FU) may work via a RNA directed mechanism while ...*

Patients and methods: *Two hundred fourteen patients from nineteen Italian centers were randomized to the control arm ...*

Results: {*Nine CR and twenty-seven PR were obtained on one hundred eleven evaluable patients treated in experimental arm (RR = 32%, 95% confidence interval (95% CI): 24%-42%), while two CR and eleven PR were observed among one hundred three evaluable patients in control arm (RR = 13%, 95% CI: 7%-21%)* }$_{premise1}$. ... {*Eighty percent of patients receiving second-line chemotherapy in control arm were treated with continuous infusion 5-FU* }$_{premise5}$.

Conclusions: *Alternating,* [*schedule-specific biochemical modulation of FU is more active than ...*]$_{claim1}$. [*However, the overall survival was similar suggesting that alternating bolus and infusional 5-FU upfront may be as effective as giving them in sequence as first- and second-line treatment*]$_{claim2}$.

Figure 3.1: An abstract from PubMed 11142481. We remove several sentences for brevity. The sequences in curly brackets are pieces of evidence and those in square brackets are claims.

research (Lawrence and Reed, 2020; Accuosto and Saggion, 2020). It is a sentence-level scheme used in the classification of sentences by their functions within a scientific paper. For example, a sentence belongs to the *Background* zone if it is used as a description of generally accepted background knowledge and it belongs to the *Aim* zone if it is a statement of a research goal. To date, there has been a notable absence of research exploring the influence of zoning information on the specific tasks related to argument component identification and classification.

In this thesis, we work towards closing this gap by performing a fine-grained analysis of the impact of zoning information on the tasks of argument component identification and classification. We choose the PubMedRCT (Dernoncourt and Lee, 2017) as the zoning scheme used in our paper. This scheme consists of five zones, namely *Background*, *Objective*, *Method*, *Result* and *Conclusion* (see Figure 3.1 for

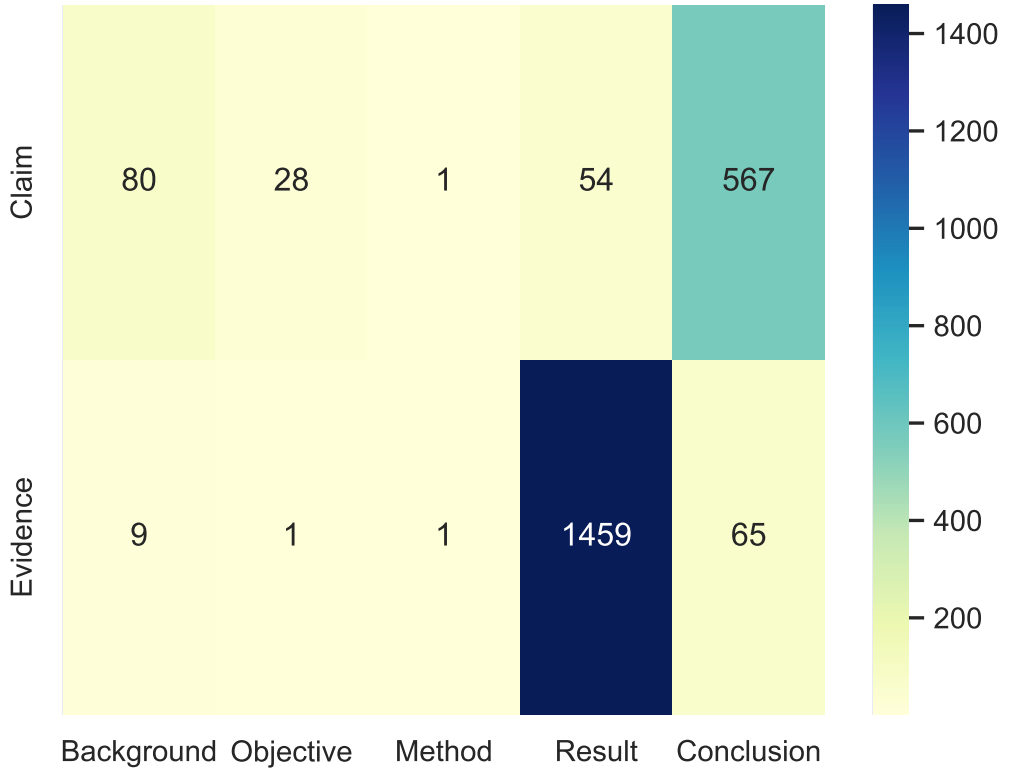


Figure 3.2: Distribution of argument components and zoning information within the training subset of AbstRCT dataset. Zoning labels are predicted labels using a tool named HSLN (Jin and Szolovits, 2018)

some examples). By doing our own frequency analysis on AbstRCT (Mayer et al., 2020) dataset, we find that argument components mainly exist in the *Result* and *Conclusion* zones, as shown in Figure 3.2. Specifically, evidence is more likely to occur in the *Result* zone and claims are more likely to occur in the *Conclusion* zone. Based on these findings, our hypothesis is that relying on zoning information, a model can mine argument components more accurately. We investigate the impact of zoning information on both token-level and sentence-level argument mining tasks.

Our contributions, elaborated upon in Chapter 1, are mentioned below:

- Our study represents a pioneering effort in integrating zoning information into argument component identification and classification tasks.

- Additionally, we introduce a straightforward yet efficient approach to leverage the inherent structural patterns present in biomedical abstracts. This method not

only validates the efficacy of zoning information but also mitigates any potential complications arising from changes in model complexity.

- Moreover, through rigorous experimental evaluation, we demonstrate the utility of zoning information across both token-level and sentence-level argument mining tasks, underscoring its beneficial impact on the overall performance.

The rest of this chapter is structured as follows: In Section 3.2, we introduced methods to utilise zoning information as genre-specific structural features at the sentence-level and token-level argument mining tasks. The details of experimental settings are described in Section 3.3. The main results of our model are displayed in Section 3.4. Analysis of the results is presented in Section 3.5, followed by the related work in Section 3.6. Finally, we conclude this chapter in Section 3.7.

3.2 Methodology

We propose two models (depicted in Figure 3.3): a sentence-level argument mining model (SLAM) and a token-level argument mining model (TLAM), for the sentence-level and token-level tasks, respectively. We will provide a formal definition of the two tasks in Section 3.2.1. SLAM is based on Accuosto et al. (2021) while TLAM is based on Mayer et al. (2020). The main difference between their models and ours is the utilisation of zoning information in such a way that changes to the models are minimal, and that they directly assess the effect of zoning information. Before introducing our proposed method, we first give a formal definition of the task solved in this thesis.

3.2.1 Task Definition

In this chapter, we solve the ACIC subtask which combines the ACI and ACC subtasks. According to the granularity of the task, it can be divided into sentence-level ACIC subtask and token-level subtask.

Sentence-level. For the sentence-level task, the smallest unit of annotation is each sentence and each sentence is annotated as either one type of AC or non-argumentative sentence. We formally define an argumentative document $DOC = \{x_1, x_2, ..., x_m\}$ as a sequence of m sentences, where m_i represents the i-th sentence in the document. The goal of the sentence-level ACIC task is to predict a label L_C for each $x_i \in DOC$. Here, $L_C \in ACTYPE \cup non - arg$, where $ACTYPE$ denotes the

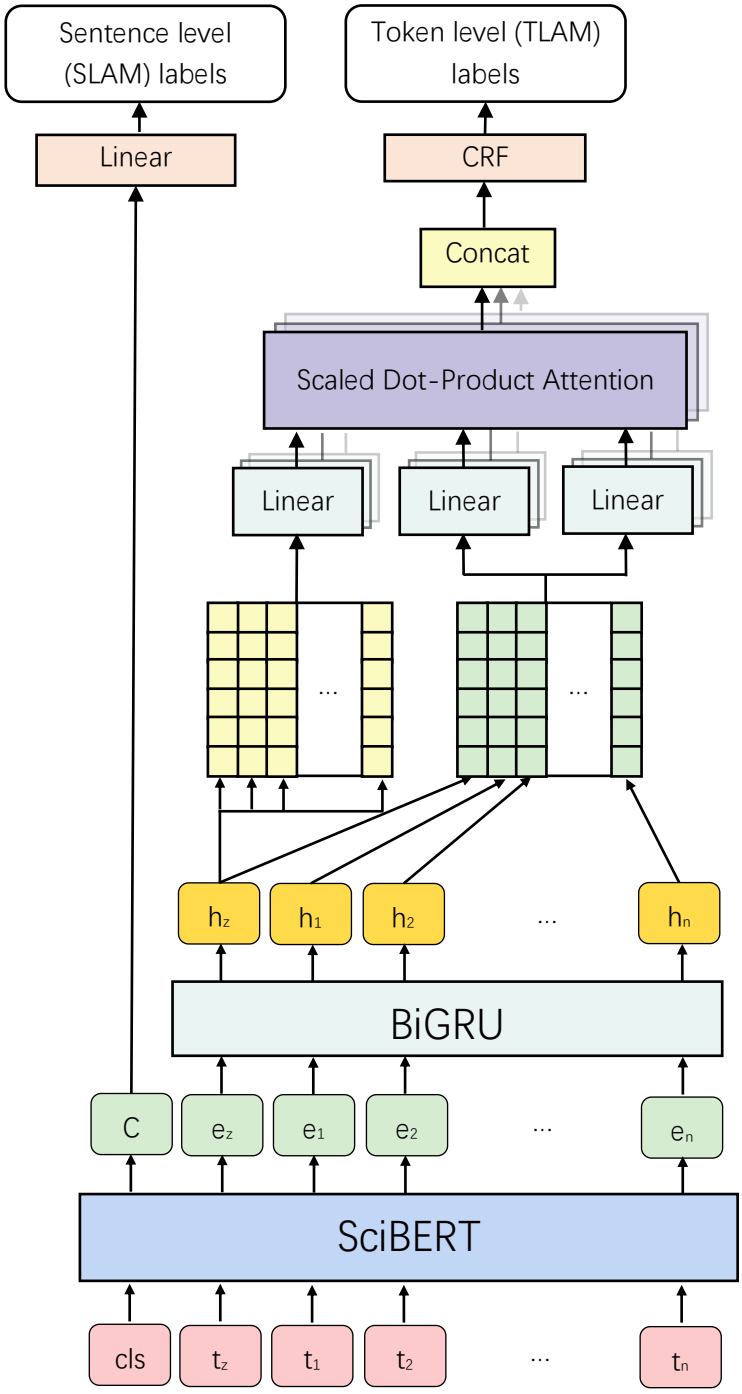


Figure 3.3: Overview of our model. t_z represents zoning labels, and cls is the special token [CLS] in SciBERT

classes of AC types and depends on the specific annotation scheme used. $non-arg$ represents non-argumentative sentences. It is worth mentioning that in some annotation schemes (Niculae et al., 2017), all sentences are annotated as AC, so there are no non-argumentative sentences .

Token-level. For the token-level task, the smallest unit of annotation is each token and each AC consists a span of continuous tokens. Each token either belongs to an AC span or a non-argumentative span. We formally define an argumentative document $DOC = \{t_1, t_2, ..., t_n\}$ as a sequence of n tokens, where t_i represents the i-th token in the document. The goal of the token-level ACIC task is to predict a label L_C for each $t_i \in DOC$. Here, $L_C \in \{B-ACT_i, I-ACT_i, O|ACT_i \in ACTYPE\}$. ACT_i is the i-th type of AC in $ACTYPE$. $B-$ and $I-$ represents beginning and inside token in an AC, respectively. O represents the non-argumentative token.

In the following subsections, we describe how we combined zoning information with argument mining and present the details of our two models.

3.2.2 Utilisation of Zoning Information

To the best of our knowledge, no dataset currently exists annotated with both zoning and argument component labels. Consequently, we undertake the task of predicting zoning labels for both the AbstRCT and SciARG datasets. In selecting the zoning scheme, we opt for PubMedRCT (Dernoncourt and Lee, 2017) for two primary reasons. Firstly, this scheme is utilised in annotating the largest dataset, PubMedRCT200k. Secondly, a high-accuracy (F1 Score 92.6) tool named HSLN (Hierarchical Sequential Labelling Network (Jin and Szolovits, 2018)) is available, facilitating the process of zoning label prediction,[2] which performs best on this dataset.

Given an abstract X that consists of m sentences:

$$X = (x_0, x_1, ..., x_m) \tag{3.1}$$

we first apply the HSLN model on each sentence to obtain the zoning label.

$$z_i = HSLN(x_i) \tag{3.2}$$

As discussed above and shown in Figure 3.1, the convention typically used in biomedical abstracts is to explicitly place the zoning labels at the beginning of each zone

[2]https://github.com/jind11/HSLN-Joint-Sentence-Classification

(e.g. Background, Method, etc.). Here, each zone consists of one or more sentences. Inspired by this, we placed the corresponding zoning label in front of each sentence in the abstract (since zoning is formally a sentence classification task). Afterwards, we used these sentences enriched with zoning information as the input to SLAM and TLAM.

$$input = concatenate(z_i, x_i) \tag{3.3}$$

We adopt a direct methodological approach to empirically confirm that the observed improvement indeed originates from the incorporation of zoning information, rather than solely from the complexity of the model design.

3.2.3 Sentence-Level Argument Mining (SLAM)

For the sentence-level argument mining task, we do not need to identify boundaries of argument components, so we treat it as a sentence classification task. We employed the pre-trained SciBERT model (Beltagy et al., 2019) to obtain sentence embeddings, drawing upon the results of Mayer et al. (2020) who showed that SciBERT yields the best results in biomedical literature argument mining. We used a linear layer as the sentence classifier.

Specifically, we followed the work of Accuosto et al. (2021) and directly used the conventionally used `[CLS]` token in BERT-based models as the representation of the class of each sentence in an abstract. The `[CLS]` token is then passed to a linear layer. Finally we employed a Softmax function to obtain the probability distribution of argument component types.

$$y_i = Softmax(W[CLS] + b) \tag{3.4}$$

In line with Accuosto et al. (2021), we chose cross entropy loss as the loss function for the sentence-level model.

3.2.4 Token-Level Argument Mining (TLAM)

We treat the token-level argument mining task as a sequence tagging problem, incorporating both the argument component identification and classification tasks. Similar to Mayer et al. (2020), we used SciBERT to obtain token embeddings and passed them to a BiGRU (Cho et al., 2014) sequence encoder. Finally we employed a conditional random field (CRF) layer to capture label dependencies. Furthermore, we added a multi-head

attention operation to transfer the sentence-level zoning labels into token-level labels. In particular, we used the embeddings of each token rather than the `[CLS]` token as the output of SciBERT:

$$e_z, e_1, e_2, ..., e_n = SciBERT(input) \tag{3.5}$$

BiLSTM has proven to be effective for the task of sequence tagging. However, Mayer et al. (2020) found that BiGRU performs better than BiLSTM on the AbstRCT dataset. Therefore, we selected BiGRU as the sequence encoder. We concatenate both forward and backward hidden state vectors $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ to obtain the encoding h_t for each token in a sequence:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{3.6}$$

Considering that the sequence labelling task is a token-level task whereas adding zoning labels in front of each sentence only provides sentence-level information, we propose a zoning attention method based on the multi-head attention mechanism used in the transformer architecture (Vaswani et al., 2017), to transform sentence-level into token-level information.

To be specific, given the representation of a sequence after the BiGRU encoder $h = (h_z, h_1, h_2, ..., h_n)$, we used a duplicate of the zoning token h_z as query Q to add the zoning information into each token, while the key K and value V are the same as in the transformer:

$$Q = [h_z, h_{z1}, h_{z2}, ..., h_{zn}] \tag{3.7}$$

$$K = V = [h_z, h_1, h_2, ..., h_n] \tag{3.8}$$

where h_{zi} is identical with h_z.

We then used scaled-dot attention to obtain the representation of each head and the final output of the multi-head attention $O_{attention}$ is calculated by concatenating all representations of each head. Finally, a CRF layer is used to learn the dependency between labels:

$$Y = CRF(O_{attention}) \tag{3.9}$$

where Y is a series of predicted labels. As with the sentence-level task, we also employed cross entropy loss as the loss function for the token-level task.

3.3 Empirical Study

In this section, we describe the experiments undertaken and subsequently provide an analysis of the results obtained from these experiments.

3.3.1 Data

We evaluate our model on two datasets on medical scientific abstracts. One dataset is used for the token-level task and the other for the sentence-level task. The statistics of these two datasets is shown in Table 3.1

	Documents	All ACs	Average ACs
Neo_train	350	2267	6.5
Neo_dev	50	326	6.5
Neo_test	100	686	6.9
Gla_test	100	594	5.9
Mix_test	100	570	5.7
SciARG_train	250	2431	9.7
SciARG_test	35	356	10.2

Table 3.1: Statistics of datasets. In order to show the difference between different test sets of the AbstRCT dataset, we report the data statistics of three test sets separately. Here, *Neo*, *Gla* and *Mix* represent neoplasm, glaucoma and mixed.

AbstRCT dataset (Mayer et al., 2020).This is a token-level dataset and there are 659 biomedical abstracts in this dataset. It consists of three types of argument components, namely *major claim*, *claim*, and *evidence*. This dataset has three parts. The biggest part is the neoplasm corpus, which is split into the training set, development set, and test set. Additionally, there are two other test sets. The glaucoma test set includes only abstracts concerning glaucoma, whereas the second one is a mixed set with 20 abstracts concerning each disease in the dataset (neoplasm, glaucoma, hypertension, hepatitis and diabetes), respectively.

SciARG dataset (Accuosto et al., 2021).The dataset utilised in this study operates at the sentence level, with annotators treating individual sentences as units for annotation. Comprising 285 biomedical abstracts, it employs a fine-grained scheme encompassing eleven distinct types of argument components for annotation purposes. The details can be seen in Table 3.2.

AC type	Description
proposal	high level description of the proposed approach/solution
proposal-implementation	processes/tools/methods that are part of the proposal
observation	data obtained from experiments
result	direct interpretation of observed data
result-means	results and the means by which they were obtained
conclusion	high-level interpretation/generalisation of results
means	secondary methods/processes not part of the proposal
motivation-problem	known problem/limitation addressed by the proposal
motivation-hypothesis	new ideas/paths for known problems/limitations
motivation-background	known information to support the proposed approach
information-additional	additional information (definitions/examples)

Table 3.2: The definition of AC types on the SciARG dataset. Adopted from Accuosto et al. (2021).

3.3.2 Baselines

In this work, we designed a rule-based heuristic method and chose two existing transformer-based models as baselines. The reason for choosing the latter two is that they are similar to our models. In this way, we can maintain comparability by minimising changes to their model architecture, thus directly testing the effect of incorporating zoning information.

	First_Token	Other_Tokens
Background	O	O
Objective	O	O
Method	O	O
Result	B-evidence	I-evidence
Conclusion	B-claim	I-claim

Table 3.3: Illustration of the heuristic method. *First_Token* denotes the first token of each sentence, and *Other_Tokens* means other tokens that are not the first token in the sentence.

Heuristic method. As depicted in Figure 3.2, zoning and argument components are strongly related. To directly assess the extent to which zoning information can help in identifying and classifying argument components, we designed a heuristic method for the token-level task, which applies the following rules: sentences labelled as *Background*, *Objective* and *Method* are classified as non-argumentative sentences, and all the tokens of non-argumentative sentences are all labelled as O (Outside). Sentences labelled as *Result* are considered evidences and labelled as *Conclusion* are

classified as claims. The first token in *Result* and *Conclusion* sentences are labelled as B-evidence and B-claim respectively, while succeeding tokens are labelled as I-evidence and I-claim.

Mayer et al. (2020) employed a fine-tuned SciBERT model with a BiGRU network and a CRF layer, which is a common method for sequence tagging tasks. We use it as a baseline for the token-level task.

Accuosto et al. (2021) used the cased version of SciBERT as a base model and feed the representation of the `[CLS]` token into a linear classifier followed by a Softmax function. We utilised this as a baseline for the sentence-level task.

3.3.3 Experimental Settings

The token-level task is a BIO sequence labelling task. Like Mayer et al. (2020), we merged major claims and claims into claims considering the negligible occurrences of major claims. Finally, the token-level task was cast as a five labels (i.e., B-claim, I-claim, B-evidence, I-evidence and outside) sequence tagging task. For this task, we used the uncased SciBERT model, and fine-tuned it with Adam optimizer (Kingma and Ba, 2015) for three epochs. The hidden dimension of a single GRU for each direction in the BiGRU sequence encoder was set to 768. We set the learning rate to 5×10^{-5}. For the sentence-level task, we used the cased SciBERT model. We used the Adam optimizer with a learning rate of 2×10^{-5}. The number of training epoch was set to 15. The hyperparameters for the SciARG dataset are selected based on five-fold cross-validation evaluations in the training set. Both uncased and cased SciBERT were downloaded from Huggingface (Wolf et al., 2020).

3.4 Main Results

We report macro-averaged (F1) and micro-averaged (f1) scores for the token-level task as Mayer et al. (2020) did, and macro-averaged F1-scores weighted by class cardinality for the sentence-level task as Accuosto et al. (2021) did. All these scores are a mean across ten different runs of the model training with different random seeds. In the sentence-level task, we also report the results of a specific task named *main unit identification* proposed by Accuosto et al. (2021), which aims at finding the sentence describing the most significant contribution of a research paper. The results of token-level and sentence-level argument mining are shown in Table 3.4 and Table 3.5, respectively. In

Table 3.4, F1 stands for macro-averaged F1-score and f1 corresponds to micro-averaged F1-scores.

Models	Neoplasm		Glaucoma		Mixed	
	f1	F1	f1	F1	f1	F1
Heuristic method	87.23	80.20	87.32	82.17	88.00	80.97
Mayer et al. (2020) [3]	90.05	83.48	91.50	85.67	90.88	84.27
TLAM_without_Att	90.82*	85.09*	91.93†	87.03*	91.68*	85.70*
TLAM	90.78*	**85.59***	92.18*	87.82*	91.63*	86.05*
TLAM_Single_B	90.12	84.64*	91.74	87.66*	90.99	85.41*
TLAM_Single_O	90.15	84.79*	91.74	87.73*	90.94	85.25*
TLAM_Single_M	90.05	84.77*	91.79	**87.99***	91.33†	85.72*
TLAM_Single_R	90.68*	85.29*	**92.19***	87.52*	**91.69***	85.98*
TLAM_Single_C	**90.94***	85.51*	91.86	87.60*	91.61*	**86.16***

Table 3.4: Results for token-level argument mining. * and †indicates statistically significant improvements over the baselines compared to our model, according to a t-test with $p<0.05$ and $p<0.1$
. Best results are highlighted in bold. TLAM is our token-level argument mining model. F1 and f1 stand for macro- and micro-averaged F1-scores, respectively. TLAM_Single_{B,O,M,R,C} means the model only exploits a single zoning label, i.e., *Background*, *Objective*, *Method*, *Result* and *Conclusion* respectively.

Method	Component	Main Unit
Accuosto et al. (2021) [4]	67.38	86.76
SLAM	**69.08***	**88.79***
TLAM_Single_B	68.32*	87.52*
TLAM_Single_O	68.39*	87.78*
TLAM_Single_M	68.87*	87.99*
TLAM_Single_R	68.95*	88.13*
TLAM_Single_C	68.88*	88.24*

Table 3.5: Results for sentence-level argument mining. Best results are highlighted in bold. SLAM is our sentence-level argument mining model. Significance tests are conducted between Accuosto et al. (2021) and the other methods. * indicates statistically significant improvements over Accuosto et al. (2021) compared to the other models, according to a t-test with $p<0.05$.

We find that the heuristic method obtains very high macro- and micro-averaged F1-scores, despite its simplicity. This result is in line with our finding in Figure 3.2 that the zoning information is highly related to the ACs. Both of them suggest that zoning information is indeed useful for argument component identification and classification,

even without the help of additional semantic information. Regarding the token-level experiment, from the obtained results we observe that the overall macro-averaged F1-score improves by 2.11, 2.15 and 1.78 percentage points on the neoplasm, glaucoma and mixed test sets, respectively. All improvements in micro-averaged F1-score are less than one percentage point, which is mainly due to the dominance of the 'O' label.

From the results reported in Table 3.5, we find that the sentence-level task benefits from zoning information as well, not only for classifying argument components, but also for the identification of main units. Interestingly, even though the number of argument component types is higher (eleven) than the number of zoning types (five), the fine-grained component type classification task can still benefit from the coarser zoning labels.

To understand the influence of multi-head attention, we ran both TLAM_without_Att and TLAM models. The difference between them is that the latter does not include multi-head attention and directly uses the output of BiGRU as the input of the CRF layer. It is evident in Table 3.4 that multi-head attention improves the macro-average F1-score by roughly 0.5 percentage points. One thing worth mentioning is that even without additional multi-head attention, the improvement using zoning information is also obvious (1.61, 1.36 and 1.43 percentage points on the neoplasm, glaucoma and mixed test sets, separately).

Furthermore, we conducted experiments to investigate the contribution of each type of zoning label. Unlike other experiments that test the contribution of one label type by removing this type to detect the degradation in the model's performance, we conducted experiments to test the results of incorporating only one type of zoning label. For instance, TLAM Single_B only adds *Background* label before the sentences that belong to *Background* zone, while the sentences that belong to other zones are sent to the SciBERT directly without any processing. The reason why we choose this way is that each sentence has a zoning label, if we remove one type of zoning labels and keep others, the model might learn that the sentences without any zoning labels belong to the same type, which cause that we cannot get the correct effect of each type of zoning label. The results are shown in Table 3.4 and Table 3.5.

For the token-level argument mining, as shown in Table 3.4, we observe that even when using only one type of zoning label, the five models perform better than the model developed by Mayer et al. (2020), which does not include zoning information. Among

[3]We downloaded their code from https://gitlab.com/tomaye/ecai2020-transformer_based_am to reproduce these results. We also directly employed their code in our evaluation.

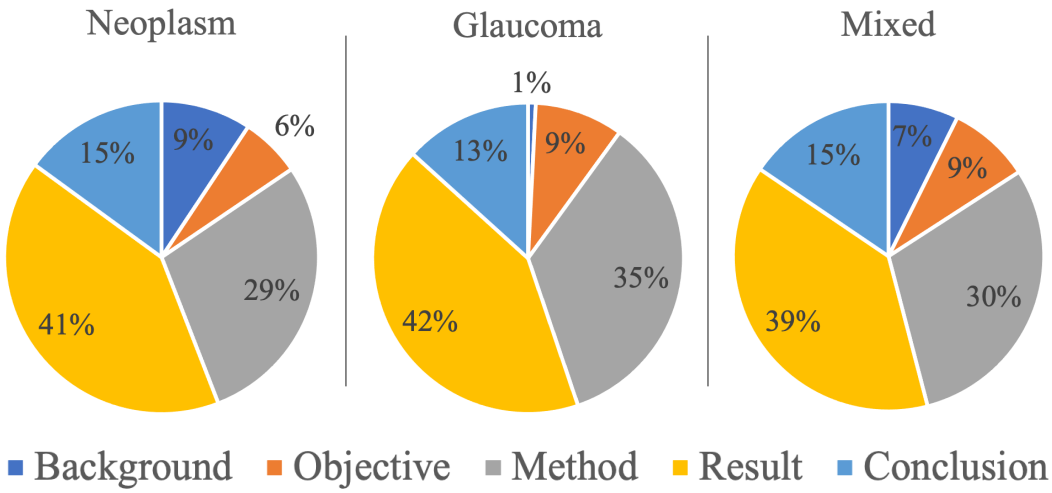


Figure 3.4: Distribution of predicted zoning labels in three different test sets within the AbstRCT dataset.

	precision	recall	F1-score
Background	77.59	78.80	78.19
Objective	74.25	67.64	70.79
Method	95.06	97.51	96.27
Results	96.44	95.51	95.98
Conclusion	97.79	96.04	96.91

Table 3.6: The performance of the HSLN model rerun by ourselves. It includes the precision (P), recall (R), and F1 score (F1) that are computed as percentages for each label.

the five labels, models using the *Result* and *Conclusion* labels alone can obtain results comparable with the model using all five types of labels, when considering all three test sets. It is noticeable that the model incorporating only the *Conclusion* label outperforms the model that uses all five types of labels in four different F1-score. It is also worth noting that using the *Method* label alone achieves the best performance in the glaucoma test set. We posit that this is due to the high proportion of *Method* sentences (30%) in abstracts, especially in the glaucoma test set (35%), and the very high performance on such label, as shown in Figure 3.4 and Table 3.6, separately. Even though this type of zoning label is least relevant to argument components, it helps to effectively exclude these non-argumentative sentences given the high frequency of occurrences.

It is clear that the information provided by *Background* and *Objective* leads to the least improvement in model performance. One possible reason is that these two labels have little correlation with the appearance of argument components, as shown

in Figure 3.2. Another reason could be that the accuracy of the predictions of these two types of labels is relatively low (78.19 F1-score for *Background* and 70.79 for *Objective*, as shown in Table 3.6), and these two labels tend to be confused by the HSLN model (Jin and Szolovits, 2018). Improvements might be obtained when using gold-standard zoning labels rather than predicted labels.

As for sentence-level argument mining, from Table 3.5, we observe similar results that our model shows improvement even when only one type of zoning label is provided. It's worth mentioning that for the main unit identification task, the impact of *Objective* is the greatest, because the parts that are labeled as the main unit in that dataset often exist within the *Objective* zone.

3.5 Analysis

3.5.1 Impact of Zoning Labels

In order to assess the efficacy of our model in detail, we conduct a comparison with the baseline method (Mayer et al., 2020) based on three examples. The outcomes are presented in Table 3.7.

Positive Impact. In the first example, the model (Mayer et al., 2020) incorrectly classified the AC into a conclusion in the absence of zoning information. However, our model correctly predicted the label for each token after utilising the zoning information "*Result*" and learning its high correlation with "evidence" .

Negtive Impact. We also analyse the examples when zoning label fails to show the possible limitations of our model. We find that there are mainly two situations. The first one is due to the incorrectly predicted zoning label, as shown in Table 5.9. In Example 2, our model predicted the AC as "evidence" because the zoning label is "*Result*". However, the correct zoning label is "*Conclusion*" and correct AC label is claim.[3] We think it could be avoid if the golden zoning labels are provided. The second issue is with the labeling itself, as some sentences in the backgrounds are actually from the previous section. The third one is that sometimes our model will rely too much on the zoning label and incorrectly predict the label of ACs, as shown in Example 3. Since most of the sentences in the objective zone are non-argumentative, the model incorrectly predicts all the labels as O. This may be caused by the model over-relying on the zoning label and ignoring the semantic information.

[3]We do not have the golden zoning labels, and we think it should be "*Conclusion*" based on the context.

Num	Example	ZL
1	Input: both patients with objective response and disease stabilisation had clearly better symptom control than those with disease progression. Pre(Z): {B-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E} Pre(NZ): {B-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C} Golden: {B-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E}	Res
2	Input: overall, global quality of life was maintained in both treatment groups. Pre(Z): {B-P, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E, I-E} Pre(NZ): {B-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C} Golden: {B-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C}	Res
3	Input: The immunosuppressive drug rapamycin may influence insulin sensitivity in insulin-responsive tissues. Pre(Z): {O, O, O, O, O, O, O, O, O, O, O, O} Pre(NZ): {B-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C} Golden: {B-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C, I-C}	Obj

Table 3.7: Some examples to show the impact of zoning labels. Given an input text(*Input*), *Pre(Z)* and *Pre(NZ)* denote the predicted labels by TLAM (with zoning labels) and Mayer et al. (2020) (without zoning labels). In contrast, *Golden* means the ground truth. *ZL* represents the zoning label for the input sentence. *Obj* and *Res* are abbreviations for *Objective* and *Result*, respectively.

3.5.2 Impact on Boundary Detection and Type Classification

For the token-level dataset, the model we proposed is used to solve the ACIC task. In Section 3.4, we can conclude that zoning labels can improve the performance of the model on this task. However, it is unclear, whether the improvement of the model comes from the improvement of Boundary Detection or Type Classification. Therefore, in this chapter we mainly discuss this issue. Our model mainly contains five labels, namely *B-claim*, *I-claim*, *B-evidence*, *I-evidence* and *O*. In order to analyze from both the boundary and type perspectives, we divide the four tags except O into two groups in two ways. As shown in Table 3.8 and Table 3.9, B-F1 is the average score of B-claim and I-claim, and I-F1 is the average score of I-claim and I-evidence. We also calculate the C-F1 and E-F1 for the performance on the claim type and evidence type in a similar

Models	Neoplasm		Glaucoma		Mixed	
	B-F1	I-F1	B-F1	I-F1	B-F1	I-F1
Heuristic method	72.28*	83.57*	75.14*	85.94*	73.47*	84.23*
Mayer et al. (2020)	69.83*	84.01*	73.26*	86.82*	70.36*	85.77*
TLAM	**73.93**	**86.11**	**77.44**	**89.01**	**74.30**	**87.05**

Table 3.8: Results for token-level argument mining according to boundary labels. B-F1 and I-F1 stand for macro-averaged F1-scores for beginning token (B-claim and B-evidence) and inside token (I-claim and I-evidence), respectively. Significance tests are conducted between TLAM and the other two methods. * indicates statistically significant improvements over the other two models compared to TLAM, according to a t-test with $p<0.05$.

Models	Neoplasm		Glaucoma		Mixed	
	C-F1	E-F1	C-F1	E-F1	C-F1	E-F1
Heuristic method	**73.88**	81.96*	80.04	80.93*	73.71*	83.97*
Mayer et al. (2020)	69.65*	84.17*	76.52*	83.53*	72.50*	83.62*
TLAM	73.80	**86.17**	**80.45**	**85.99**	**75.58**	**85.76**

Table 3.9: Results for token-level argument mining according to type labels. C-F1 and E-F1 stand for macro-averaged F1-scores for claim (B-claim and I-claim) and evidence (B-evidence and I-evidence), respectively. Significance tests are conducted between TLAM and the other two methods. * indicates statistically significant improvements over the other two models compared to TLAM, according to a t-test with $p<0.05$.

way.

It is obvious that the zoning information can not only benefit Type Classification but also Boundary Detection significantly. Due to the strong correlation between zoning information and AC types, the improvement stemming from zoning information for Type Classification is expected. As for the impact on Boundary Detection, one possible reason is that with zoning information, the model is less likely to predict non-argument components as ACs, especially in the *Background*, *Objective* and *Method* zone.

3.6 Related Work

Text Zoning for Scientific Literature There are different zoning schemes for different domains, including, argumentative zoning (Teufel et al., 1999) for computational linguistics, CoreSC (Liakata et al., 2010) for chemistry, MAZEA (Dayrell et al., 2012) for physical sciences and engineering, and life and health sciences, and PIBOSO (Kim et al., 2011), GENIA-MK (Thompson et al., 2011; Shardlow et al., 2018) and PubMedRCT

(Dernoncourt and Lee, 2017) for biomedicine. Argumentative zoning is the earliest schema that includes seven categories of zones, such as *Aim, Background, Contrast*. The idea of argumentative zoning is to follow the knowledge claims made by authors. For example, sentences for the description of new knowledge claims belong to *OWN* zone, while for the description of existing knowledge claims belong to *OTHER* zone. CoreSC, meanwhile, is a concept-driven scheme. It seeks to retrieve the structure of research components from a paper as generic high-level Core Scientific Concepts and thus obtains humanly-readable representations of the research process, including categories such as *Model* (to describe a theoretical model or framework) or *Conclusion* (to describe statements inferred from research results). A detailed comparison between these two schemes can be found in (Liakata et al., 2012). MAZEA and PIBOSO both consider six classes, the former includes *Background, Gap, Purpose, Method, Result and Conclusion*, while the latter includes *Background, Population, Intervention, Outcome, Study Design and Other*. GENIA-MK classifies sentences that describe bio-event into different categories based on their knowledge types (i.e., *Investigation, Observation*). All these five schemes were used for manually annotated datasets. In contrast, annotations in the PubMedRCT200k (Dernoncourt and Lee, 2017) dataset were obtained automatically based on PubMedRCT scheme designed for biomedicine. Observing that in biomedical literature, there exist zoning labels provided by publication authors themselves, Dernoncourt and Lee (2017) selected abstracts with zoning labels as the documents for their dataset. They then used a rule-based method to map author-provided labels to the 5 categories in the scheme and annotated each sentence. Adding this type of information is less laborious than adding other information such as PICO entities(Stylianou and Vlahavas, 2021) or discourse relations (Accuosto and Saggion, 2020), where labels are not readily available from publications. Although these schemes are not directly designed for argument mining, they are helpful in locating important arguments in scientific literature.

Argument Component Identification and Classification Recently, argument mining from biomedical literature has received more attention, in part due to the challenges brought about by the inherent complexity of the structure and language used in this domain (Kirschner et al., 2015). Transformer-based models have been dominant in the approaches used in the argument component identification and classification task. Mayer et al. (2020) compared different transformer-based models, demonstrating that SciBERT model (Beltagy et al., 2019) performs best on biomedical literature. Accuosto et al. (2021) employed cased SciBERT to mine argument structures from both

computational linguistics and biomedical literature. Other researchers also investigated other models. For example, Galassi et al. (2021b) designed a logic tensor network for neuro-symbolic argument mining. Galassi et al. (2023) proposed a multi-task attentive residual network for the argument mining task in different scientific domains; they made the assumption that the boundary of each argument component has already been detected correctly and focussed only on the classification task.

The methods mentioned above mainly rely on the powerful semantic processing capabilities of pre-trained language models. However, there are also some models, like our approach, that leverage additional information besides the semantic information. For instance, Stylianou and Vlahavas (2021) combined PICO information as external knowledge with argument mining and obtained significant improvement. Accuosto and Saggion (2019a) found that incorporating discourse information significantly contributes to the identification of the argumentative function. With regard to zoning more specifically, Lauscher et al. (2018) designed a tool for analysing argument and rhetorical aspects in scientific writing. This tool can be used for both argument component identification and discourse role (similar to zoning labels) classification tasks. However, it does not consider the relation between these two tasks. Differently from other work, we incorporate zoning information into whole argument component identification and classification tasks.

3.7 Summary

In this chapter, we leveraged zoning labels as genre-specific structural features and conducted an analysis of the correlation between zoning labels and argument component types. The evident correlation between the two led us to formulate a model aimed at utilising zoning labels to enhance the performance of the argument mining (AM) model on tasks related to argument components.

Consequently, we devised both a sentence-level model and a token-level model tailored for the respective tasks of sentence-level and token-level argument mining. These models incorporate the predicted zoning label at the beginning of each sentence, which is then provided as input to the encoding layer, thereby facilitating the integration of zoning information into the AM framework.

Experiment results performed at these two different levels demonstrated the effectiveness of utilising zoning information for the task of argument mining. Of particular

interest, our findings revealed that heuristic algorithms utilising zoning labels as exclusive features achieved competitive results, even when compared against approaches employing pre-trained language models. This observation underscores the significance of zoning labels as valuable features within the argument mining framework.

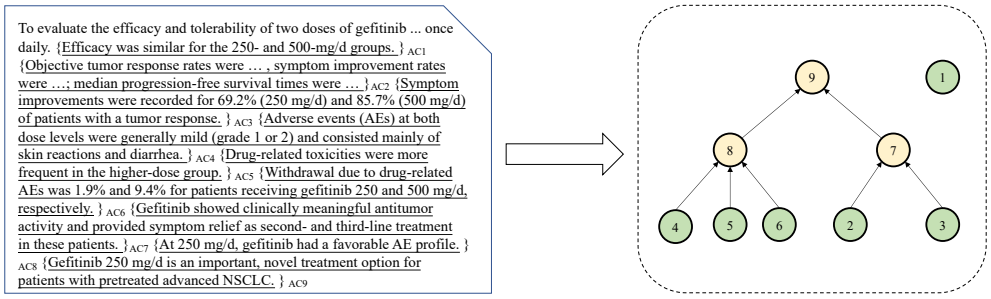
Through analysis, we observed that even if only one partition label is used, the model still gains improvement from the partition information, especially when the selected label is *Result* or *Conclusion*. This is because these two types of labels are closely related to evidence and claims. Additionally, we found that using only the *Method* label performed quite well on the glaucoma test set. We believe this is because the proportion of method labels in the dataset is high (30%), and the performance on such labels is almost perfect.

One limitation is that we only use one type of zoning schemes. However, it is possible that a bespoke zoning scheme could have been developed and have changed the results. We believe that a zoning scheme conducive to argument mining should meet two conditions. First, there should be a high correlation between argument components and zones, preferably a one-to-one or one-to-many relationship (if a zone corresponds to both some argument components (ACs) and some non-ACs, such a zone would be unhelpful or even detrimental to the argument mining task). Additionally, the amount of data annotated with this zoning scheme needs to be sufficiently large to train a high-accuracy model; otherwise, incorrect predictions of zoning labels could actually degrade the model's performance.

Chapter 4

Argument Mining with Graph-level Argument-specific Structural Features

In the previous chapter, we explored the impact of genre-specific structural features of biomedical literature, namely zoning information, on the argument component identification and classification subtasks. In this chapter, as mentioned in Chapter 1, we move to our second research question (**RQ2**), which is related to the argument-specific structural features. Because the argument structures are represented as graphs, we also model argument-specific structural features as graphs and propose a model to leverage such graph-level structural features. Specifically, we propose a novel two-stage model which leverages graph-level structural features to support AM. The first stage uses a multi-turn question-answering (QA) model to incrementally generate an initial argumentative graph that identifies relations (without AR types) among ACs. At each turn, all ACs related to the query AC are generated simultaneously, such that the sibling features between the answer ACs are considered. In addition, the partially constructed graph is used as subgraph-level structural features to support the extension of the graph with additional ACs. After the whole initial graph structure has been determined, the second stage assigns semantic types to both the ACs and ARs among them, leveraging information from this initial graph as (whole)graph-level structural features. We test the proposed method and our experiment results show that our model improves the state-of-the-art performance on two biomedical datasets for different AM subtasks. It is worth pointing that the contents of this chapter are published in Liu et al. (2023b).



(a) Argumentative Text

(b) Argumentative Graph

Figure 4.1: An example of AM from the AbstRCT dataset(Mayer et al., 2020). The argumentative text is the input and the argumentative graph is the output of AM. In (b), the nodes highlighted in yellow(7, 8, 9) are *claims*, while the remaining (green) nodes are *premises*. All of the edges in the argumentative graph constitute *support* relations.

4.1 Motivation

Building up a picture of the argument structure of a document is a complex process, which we, as humans, typically carry out in an iterative manner as we try to make sense of the document content. We firstly need to comprehend the meaning of sentences, and then link together the meanings of sentences and paragraphs in order to acquire a global meaning representation for the entire document (Landi et al., 2013). In argumentative texts, argument-specific structural features are important, as argumentative texts in the same genre usually have similar global argument-specific structural features. For example, as shown in Figure 4.1, in scientific abstracts, the bottom-level ACs (such as AC2, AC3) are usually premises about the experimental results, which are used to support aspect-based claims (AC7, AC8). The top-level AC (like AC1 in Figure 4.1) is usually a high level claim about the main conclusion of the abstract. We argue that such argument-specific structural features can be helpful for extracting argument structure. Furthermore, accurate and complete comprehension of a document's structure is likely to require more than one reading of the document; an initial scanning or skimming pass of the whole document may help to arm us with the knowledge required to better understand and make links between specific arguments that are introduced in the document (Avery and Graves, 1997; Saricoban, 2002; Toprak and Almacioğlu, 2009).

While it may be expected that the most effective computational models for AM should simulate this human behaviour, i.e., by making use of the global argument-specific structural features of the document, most existing state-of-the-art neural AM

models do not currently attempt do so. For example, several approaches tackle ARI and ARC as straightforward pair classification tasks (Galassi et al., 2018; Mayer et al., 2020; Accuosto et al., 2021). That is to say, these models take all possible pairs of ACs in a document as their input, and predict relations/labels using only information from the local context of each AC in the pair. In other words, they ignore the fact that information about the wider argument structure of the document could impact upon the most appropriate predictions. In contrast, Si et al. (2022) recognise that the use of contextual information from other related ACs can be beneficial for identifying and classifying relations. However, the only related ACs considered are those that share the the same parent in the argumentative graph. i.e., nodes which are siblings of each other. Other models make some use of global document context, in the sense that they use the entire document text as input to their joint models for ACC and ARI (Eger et al., 2017; Kuribayashi et al., 2019; Bao et al., 2021). However, the unstructured nature of the input information means that details about the argument structure of the document cannot be fully exploited. In summary, none of the existing state-of-the-art models for AM makes full use of the global argument-specific structural features of documents, suggesting that they do not perform to their fullest potential.

According to the identified shortcomings of the approaches highlighted above, in this work, we propose a novel two-stage framework for AM. The framework is inspired by previous studies into human reading comprehension behaviour, in terms of its exploitation of different types of global argument-specific structural features that are used to support the ARI, ACC and ARC subtasks. The *reading* stage carries out the task of ARI, using a multi-turn question answering (QA) model to incrementally construct an initial argumentative graph for the complete document. The graph is represented using a graph convolutional network (Kipf and Welling, 2017) (GCN). The reading stage aims to simulate the human process of skimming a document to understand its high-level argument structure. The second *post-reading* stage of the framework makes uses of the argumentative graph constructed during the reading phase to support the tasks of ACC and ARC. The stage can be considered similar to a human's second "pass" through a document, in which they are able to better understand and interpret argumentative units and their structure, using the knowledge gained from their initial skimming. Further, we compare our model with other baselines on two datasets, and analyse the impacts of global features. To summarise, our contributions are as follows:

- We propose a two-step model to explicitly utilise global argument-specific structural features to solve the AM task. During the first stage, we propose a top-down

multi-turn QA-based model which is used to solve the ARI subtask to get the initial graph as the graph-level argument-specific structural features. Then we propose a GCN-based model in the second stage to predict the type of AR and AC based on the whole initial argumentative graph.

- The sibling nodes information and the partially constructed graph (a subgraph of the initial graph) information are used as global information during the first stage.
- Experimental results show that the use of global information has a significant positive impact on the performance of all these subtasks, and allows our framework to outperform related models on most tasks when applied to two different benchmark datasets.

This chapter is structured as follows: Section 4.2 introduces the overall framework of the two-stage model and uses an example to explain how the framework works. More details of the model are described in Section 4.3. Then, the experimental settings are provided in Section 4.4. The comparison of our model with other approaches are conducted in Section 4.5. The results are analysed in Section 4.6, followed by a discussion of related work in Section 4.7. The chapter is concluded in Section 4.8.

4.2 Global Information-Aware Argument Mining Framework

4.2.1 Task Definition

Similarly to previous related efforts (Galassi et al., 2023; Ruggeri et al., 2021; Si et al., 2022; Bao et al., 2021; Kuribayashi et al., 2019), we assume that ACI has already been carried to identify all ACs in the document, and we focus on the tasks that construct and label the argumentation graph using these pre-identified ACs (i.e., ARI, ACC and ARC).

We formally define an argumentative document $DOC = \{t_1, t_2, ..., t_l\}$ as a sequence of l tokens, where t_i represents the i-th token in the document. Given a set of AC span indices $IDX = \{(s_1, e_1), ...(s_d, e_d)\}$, where s_i and e_i represent the start token and the end token of the i-th AC contained in DOC, respectively, we denote the set of all d AC spans in the DOC as $AC_{span} = \{AC_1, AC_2, ..., AC_d\}$. The goal of the ACC subtask is to predict a label L_C for each $AC_i \in AC_{span}$. Here, $L_C \in ACTYPE$, where $ACTYPE$ depends on the specific annotation scheme used. Given a pair of argument components $\{(AC_i, AC_j) \mid AC_i, AC_j \in AC_{span}\}$, the goal of the ARC subtask is to

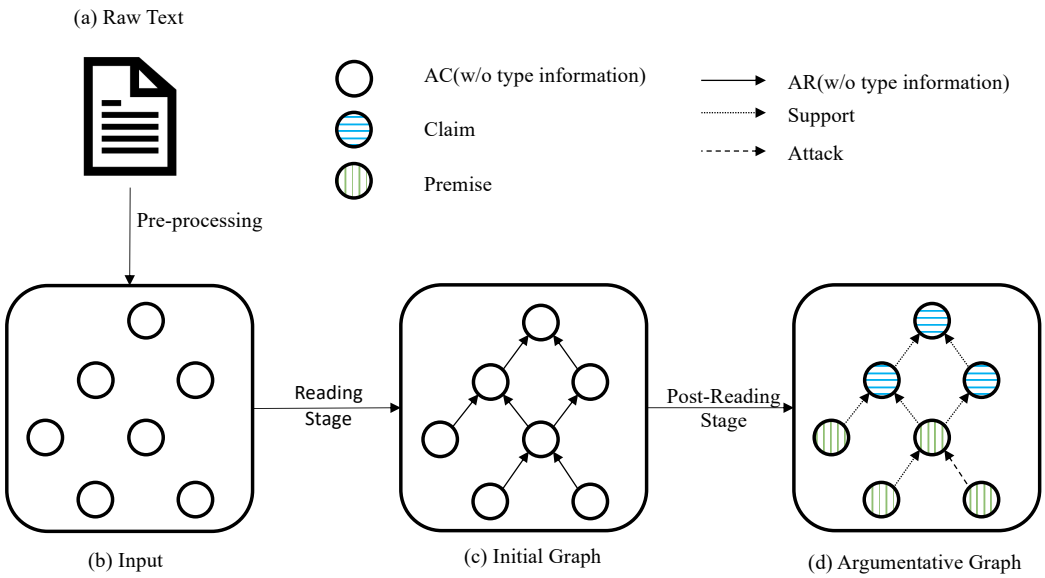


Figure 4.2: Global information-aware argument mining framework.

predict the argument relation type $L_R \in ARTYPE$ that holds between the pair, where $ARTYPE$ is also dependent on the annotation scheme.

4.2.2 Global information-aware Argument Mining Framework

In this section, we introduce our Global Information-aware Argument Mining framework (GIAM), which is shown in Figure 4.2. During the reading stage of our framework, all available information about the partially constructed graph is used to support the extension of the graph with new AC nodes. Specifically, two types of global structure information are used, the first of which is the subgraph-level argument-specific structural feature. Since the reading stage generates the argumentative graph for the document in an incremental fashion, only a partial graph (or *subgraph*) will have been constructed at each intermediate stage of the process. However, by exploiting all of the currently available subgraph information to support the extension of the graph with additional related AC nodes, the graph generation process is able to simulate the human process of comprehending a document. That is to say, as the argument structure is incrementally understood, previously digested information about the document may be used to identify and make sense of new argument components, and to appreciate how they link in with previously introduced aspects of the argument. This aspect of our model is graphically depicted in the GCN part of Figure 4.4, where the green-coloured node represents an AC whose related ACs are to be predicted, and the blue nodes represent the current

subgraph.

By using information about an AC node and the subgraph, one or more child AC nodes that are related to the AC node in question can be predicted and added to the graph. In Figure 4.4, these are depicted as the yellow nodes in the graph structure. If there are multiple child AC nodes, then these *sibling* nodes will often contain shared aspects of information. As mentioned above, previous work (Si et al., 2022) has shown that rather than trying to identify each sibling node in isolation, it can be beneficial to take advantage of the information that is shared among them when identifying new nodes in the graph. Our model thus follows a similar approach, by *jointly* predicting all sibling nodes at the same time; this *sibling-level* information constitutes the second type of global structure information used by our framework.

The result of reading stage is an initial argumentative graph structure for the whole document. Through our use of a GCN to represent the argumentative graph, the post-reading stage is able to take advantage of information from this entire graph as graph-level argument-specific structural features during the post-reading stage. These *whole graph* features support the tasks of adding classification labels to both the ACs and the identified relations between them.

4.3 Model Architecture

Our approach to the construction of the initial argumentative graph in the reading stage is inspired by Zhang et al. (2020). Specifically, we implement a multi-turn question-answering (QA) based top-down method to simulate the human process of discovering the argument structure of the document. A root query is firstly used to obtain the ROOT AC of the graph as an answer. There are two possible types of argument structures, whose ROOT AC node we determine in different ways. In tree structures—i.e., graphs where each AC has at most one outgoing AR, such as Accuosto et al. (2021)—the root node of the tree is set as the ROOT AC. For non-tree structures (e.g. Mayer et al. 2020), the source node with the longest path is defined as the ROOT AC. It should also be noted that for certain documents, their specific argumentation structure may result in the generation of a graph that consists of a number of unconnected subgraphs. Some of these subgraphs may correspond to singleton ACs, i.e, ACs that neither support nor attack other ACs, nor are they supported or attacked by other ACs. An example of the four turns QA is shown in Figure 4.3 . In the first trun (Figure 4.3 (1)) of this example, Q_R is used to represent the root query and Node 9 is the ROOT AC.

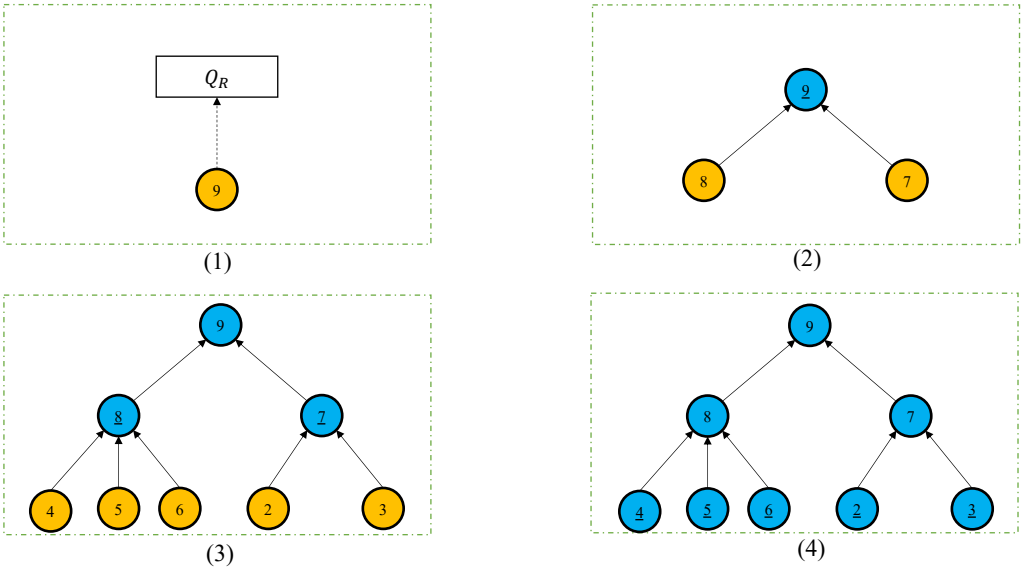


Figure 4.3: An example of the four turns QA of the initial graph generation for the argumentative text in Figure 4.1. Q_R represents the root query. The ACs whose numbers are underlined are the queries in the i-th turn. The ACs highlighted with blue are ACs in the subgraph in the i-th turn. The ACs highlighted with yellow represents the predicted answers in the i-th turn. Since no answer is predicted in the fourth turn, the initial graph is constructed completely.

In the next QA turn (Figure 4.3 (2)), the ROOT AC node (Node 9) is itself treated as a query, whose answers correspond to the set of AC nodes (Node 7 and Node 8) that point to it. All of these new nodes are jointly predicted *at the same time*, by treating the problem as a multi-label classification task, which allows sibling level information to be exploited. Each of the AC nodes (Node 7 and Node 8) that constitutes an answer in this turn is subsequently passed on to the next turn (Figure 4.3 (3)) as a new query; the process continues in the same manner until the initial argumentative graph has been fully constructed. This conforms to a layer-by-layer analysis structure from top to bottom (Nussbaum, 2002; Aharoni et al., 2014; Eger et al., 2017). At each QA turn, subgraph-level information (For example, in the third turn shown in Figure 4.3 (3), the subgraph includes Node 7, Node 8 and Node 9 and the edges among them.) is generated based on queries and answers from all previous turns.

In the post-reading stage, a GCN-based model aims to assign labels to both the ACs and the relations between them, using information from the entire graph obtained from the reading stage.

The framework of the reading stage is shown in Figure 4.4. Detailed information

about the reading stage is provided in Sections 4.3.1-4.3.5, while Section 4.3.6, describes the post-reading stage.

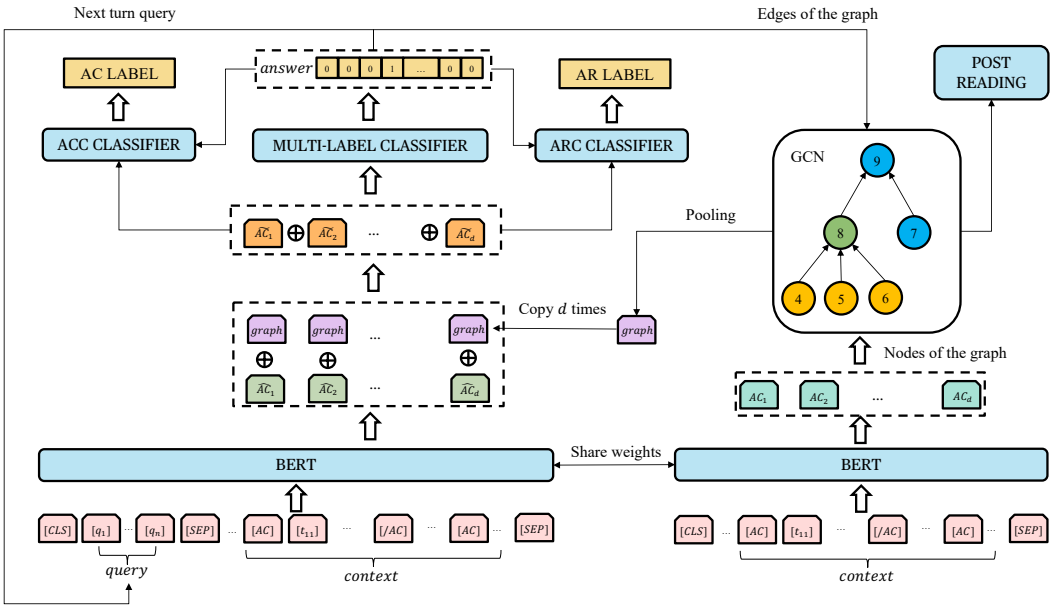


Figure 4.4: Framework of reading stage. It is an example for the third turn in Figure 4.3. This turn has two query nodes (7 and 8) and we take node 8 as an example. Now the subgraph contains three blue nodes(7, 8, 9) and the green node 8 is used as the query. The three yellow nodes (4, 5, 6) are the answers for the query.

4.3.1 Context Representation

Let $DOC = \{t_1, t_2, ..., t_l\}$ denote an argumentative document, where t_i represents the i-th token in DOC. As mentioned in Section 4.2.1, we assume that the ACI has already been performed, such that the model already identifies which ACs exist in the document, along with their span boundaries. The context representation, which remains the same for all QA turns, consists of all tokens in the document, with additional tokens inserted to denote the boundaries of each AC. Specifically, we insert an `[AC]` token before the start token of each AC token sequence in document D and an `[/AC]` token after the final token of the sequence.[1]

[1]An alternative method would be to use the [CLS] and [SEP] tokens to denote the start and end of each AC span. However, by using `[AC]` and `[/AC]` for this purpose, we are able to use the [SEP] token to help the model to distinguish between query and context parts of the input.

Since we treat each QA turn as a multi-label classification task, the number of ACs (which also corresponds to the number of label representations) must be the same for each document. Accordingly, the number of ACs in a document is fixed as the count of ACs in the document with the highest number of ACs. Then, for documents containing fewer ACs than this maximum number, the tokens `[AC][/AC]` are used for AC-level padding.

4.3.2 Query Generation

Let $query_i = \{q_i^1, q_i^2, ..., q_i^l\}$ denote the set of queries in the i-th turn. Since each QA turn may return multiple AC nodes as answers (each of which is used as a query in the next turn), there may be more than one query in each turn. The *root query*, which is used as the first turn of QA to obtain the `ROOT` AC, is handled in a different manner from non-root queries, which are used in subsequent QA turns.

To determine the `ROOT` AC, we use a manually designed query *"What is the main argument of the document?"*, which we call Q_R.[2]

$$query_1 = \{Q_R\} \tag{4.1}$$

Each non-root query q_i^j corresponds to an AC predicted during the previous QA turn. Specifically, in the i-th QA turn, given the $query_i$ and the context, the model will predict the answers $answer_i = \{a_i^1, a_i^2, ..., a_i^l\}$, where a_i^j is an AC. For each answer $a_i^j \in answer_i$, a_i^j is treated as a query for $i+1$-th turn:

$$query_{i+1} = answer_i, i > 0 \tag{4.2}$$

4.3.3 Subgraph Representation

After each QA turn, a directed subgraph G is obtained, which corresponds to the partial representation of the argument structure of the document DOC, which has been constructed based on the answers of the current turn and all previous turns. In G, each node represents an AC, and each edge represents an AR.

At the i-th turn, $G_i = \{V_i, E_i\}$ denotes the subgraph after the i-th turn QA, where

[2]We compare different types of the root query and find that the natural language query performs best, the details are introduced in Section 4.6.1.

V_i is a set of ACs, and E_i is a set of ARs. V_i is updated based on V_{i-1} and the $answer_i$:

$$V_i = V_{i-1} \cup answer_i \tag{4.3}$$

E_i is updated based on E_{i-1}, $answer_i$ and $query_i$:

$$E_i = E_{i-1} \cup \{(a_i^m, q_i^n) | a_i^m \in answer_i, q_i^n \in query_i\} \tag{4.4}$$

To represent the graph, a GCN layer is employed. Specifically, we encode D using a BERT-based pre-trained model to obtain a representation of each `[AC]` token, which is used as the node representation in the subgraph:

$$AC_1, AC_2, ..., AC_d = BERT(D) \tag{4.5}$$

A GCN layer is then employed to conduct message passing, which will ensure that each node has access to information about all other nodes in the graph when the construction of the graph is complete:

$$G_i^0 = [AC_1, AC_2, ..., AC_d] \tag{4.6}$$

$$G_i^{l+1} = \sigma(\tilde{A} G_i^l W^l + b^l) \tag{4.7}$$

where G_i^l is the set of all vectors in the l-th layer of GCN at the i-th turn and $\tilde{A}$ is the normalized adjacency matrix of the graph. W^l, b^l are trainable parameters that represent the weight matrix and bias vector of the l-th layer, respectively. σ represents the ReLU activation function.

We use the output of the final GCN layer as the representation of each AC. During the construction of the graph, nodes are not fully aware of the full graph structure. Therefore, a mean pooling method is used to obtain the representation of the subgraph:

$$Graph_i = Pooling(node_i^1, node_i^2, ..., node_i^d) \tag{4.8}$$

4.3.4 Answer Selection via Multi-turn QA

At the i-th turn, given a set of queries $query_i$, we firstly concatenate each query q_i^j (where q_i^j is the j-th query in $query_i$) and the document D as the input to the QA-based

model:

$$C_{q_i^j \oplus D} = [CLS] q_i^j [SEP] D [SEP] \tag{4.9}$$

where $\oplus$ represents the concatenation operation. We then employ the same BERT-based model introduced in the previous subsection as the encoder. For each AC in the document, its related embedding $\hat{AC}_i^j$ for the j-th `[AC]` token is utilised as the representation. Here, the $\hat{AC}_i^j$ is a query-aware representation.

$$\hat{AC}_i^1, \hat{AC}_i^2, ..., \hat{AC}_i^d = BERT(C_{q_i^j \oplus D}) \tag{4.10}$$

Then, the subgraph representation from the previous turn, $Graph_{i-1}$, and $\hat{AC}_i^j$ are concatenated, so that the model can capture the subgraph information:

$$\widetilde{AC}_i^j = \hat{AC}_i^j \oplus Graph_{i-1} \tag{4.11}$$

Finally, $\{\widetilde{AC}_i^1, \widetilde{AC}_i^2, ..., \widetilde{AC}_i^d\}$ are concatenated and used as input into a multi-label classifier to predict whether an AR exists between the query AC and other ACs.

$$\widetilde{AC}_i = \widetilde{AC}_i^1 \oplus \widetilde{AC}_i^2 \oplus ... \oplus \widetilde{AC}_i^d \tag{4.12}$$

$$label_{ARI} = MultilabelClassifier(\widetilde{AC}_i) \tag{4.13}$$

Based on the findings that a joint learning method for all AM subtasks is able to improve the performance of each single task (Accuosto et al., 2021), we carry out ARC and ACC subtasks as auxiliary tasks of the main ARI subtask. It should be noted, however, that we do not use labels predicted by these auxiliary tasks as the final AC and AR labels in the argumentative graph. Rather, we use the labels predicted by slightly different methods (as described in Section 4.3.6), which are able to take advantage of information about the entire argumentative graph structure to support their predictions.

For the auxiliary ARC classifier, we use the same AC representations as those introduced above for ARI subtask, which are both query and subgraph aware, i.e., $\{\widetilde{AC}_i^1, \widetilde{AC}_i^2, ..., \widetilde{AC}_i^d\}$. These query-aware representations are also relevant for the ARC subtask, since the goal of the classifier is to determine the correct label of the

relation that holds between a query AC and an answer AC:

$$label_{ARC} = Classifier_{ARC}(\widetilde{AC}_i^j) \tag{4.14}$$

In contrast to the above two substasks, the aim of the ACC subtask is to label the ACs themselves. Therefore, relations between ACs are not directly relevant, and thus query information can be excluded from the AC representations for this task. As such, we use the query-free AC representations $\{AC_1, AC_2, ..., AC_d\}$ that are also used as the input of the GCN layer as the AC embedding. However, to allow the model to benefit from potentially important information in the subgraph, we firstly concatenate $AC_1, AC_2, ..., AC_d$ with $Graph_i$ to obtain the subgraph-aware AC representations:

$$AC_i' = AC_i \oplus Graph_i \tag{4.15}$$

Then the subgraph-aware AC representations AC_i' are subsequently passed to a classifier to predict ACC labels:

$$label_{ACC} = Classifier_{ACC}(AC_i') \tag{4.16}$$

4.3.5 Training and Inference

At training time, we train the whole model in a strongly supervised manner. To be specific, we train each QA turn i independently and construct the graph required for the $i-th$ turn based on ground truth annotations. Our training objective is:

$$Loss = Loss_{ARI} + Loss_{ARC} + Loss_{ACC} \tag{4.17}$$

For the ARI subtask, we use the Binary Cross Entropy (BCE) loss function, while for ARC and ACC subtasks, we use cross-entropy loss.

At inference time, the subgraph representation is acquired based on the predicted subgraphs. Since each AC can only appear once in the argument structure, we use a buffer *buffer* to store the ACs that have not yet been added to the argumentative graph, in order to avoid repeated predictions by the model.

There are two conditions that cause the multi-turn QA-based graph generation algorithm to terminate:

Algorithm 1: The multi-turn QA-based argumentative graph generation

Input : argumentative document D, a query Q_R for the ROOT AC, a *buffer* that consists of all ACs
Output : argumentative graph G.

1 $query \leftarrow Q_R$, $flat \leftarrow 0$;
2 $V \leftarrow \emptyset$, $E \leftarrow \emptyset$, $G \leftarrow \{V, E\}$;
3 **while** $query$ *is not none* **do**
4 $answer \leftarrow$ Select_Answer$(query, D, G)$;
5 $query \leftarrow answer$;
6 $V \leftarrow V \cup answer$;
7 **for** q *in* $query$ **do**
8 **for** a *in* $answer$ **do**
9 $E \leftarrow E \cup \{(a, q)\}$;
10 **end**
11 **end**
12 **for** a *in* $answer$ **do**
13 *buffer*.remove(a);
14 **end**
15 **end**
16 **if** buffer *is not none and* $flat = 0$ **then**
17 $query \leftarrow$ *buffer*;
18 $flat \leftarrow 1$;
19 jump to line 3;
20 **end**

No new query and buffer is empty [3]. In this situation, all ACs in the document have already been added to the argumentative graph, which means that the construction of the graph is complete, and there is no need to generate new queries.

No new query but the buffer is not empty. In this situation, none of the ACs remaining in the *buffer* are related to the ACs in the predicted subgraph that is currently under construction. This means that these ACs are either singleton ACs or they belong to different subgraph that is unconnected to the subgraph that has already been generated. In this case, we initiate a *query extension process*, which aims to generate other subgraphs and/or determining which ACs constitute singleton ACs, by using all remaining *buffer* as queries. The algorithm will terminate when either *buffer* is empty or when the next turn returns no answers.

[3] No new query for the i-th turn means that no $answer_{i-1}$ is predicted.

4.3.6 Post-Reading Stage

The output of reading stage is the whole initial graph $G_T = \{V_T, E_T\}$ representing the argument structure of the document, but without AC or AR labels. We handle the ACC and ARC subtasks in the post-reading stage, in which we leverage the whole graph level information.

For the ACC subtask, the representation of each AC consists of two parts, i.e., the AC representation prior to the application of the GCN (the input) AC_i (the query-free AC representation introduced in Section 4.3.4) and its representation after application of the GCN (the output) $node_T^i$ given an argumentative text D. Since there may be some singleton ACs, which are not related to any other ACs in the graph, we use the *add* operation to combine these two representations. If AC_i does not exist in the subgraph, the embedding for $node_T^i$ is a zero vector.

$$\widetilde{node}_T^i = node_T^i + AC_i \tag{4.18}$$

$$label_{ACC} = Classifier_{ACC}(\widetilde{node}_T^i) \tag{4.19}$$

For the ARC subtask, we adopt a method that is similar to the QA task used in the first stage. Although it is not exactly a QA task, we want to keep the format same as that used in the first stage. Specifically, when considering the relation between the i-th AC and other ACs, we consider the i-th AC as a query and concatenate it with the whole document graph. The concatenation representation of the j-th `[AC]` token $\hat{AC}_j$ and the whole graph information $Graph_T$ is utilised as the relation representation $\widetilde{edge}_{i,j}$ between AC_i and AC_j.

$$\widetilde{edge}_{i,j} = concatenate(Graph_T, \hat{AC}_i^j) \tag{4.20}$$

$$label_{ARC} = Classifier_{ARC}(\widetilde{edge}_{i,j}) \tag{4.21}$$

For both the ARC and ACC subtasks, cross-entropy loss is employed as the loss function.

4.4 Experiments

4.4.1 Datasets

We use two publicly available datasets as a means to evaluate our model and compare it with results obtained by previously proposed models. We provide brief descriptions of these datasets are shown as below, and report statistical details regarding their composition in Table 4.1. It is worth mentioning that we only introduced the types of ARs, but not the types of ACs, because this part has been introduced in the previous chapter.

	Documents	All ACs	Avg. AC Num	All ARs	Avg. AR Num
AbstRCT_Neo_train	350	2267	6.5	1427	4.1
AbstRCT_Neo_dev	50	326	6.5	210	4.2
AbstRCT_Neo_test	100	686	6.9	424	4.2
AbstRCT_Gla_test	100	594	6.0	367	3.7
AbstRCT_Mix_test	100	570	5.7	329	3.3
SciARG	285	2787	9.8	2502	8.8

Table 4.1: Statistics of datasets used in our paper. In order to show the difference of different test sets of the AbstRCT dataset, we report the data statistics of three test sets separately. Here, *Neo*, *Gla* and *Mix* represents neoplasm, glaucoma and mixed.

AbstRCT. (Mayer et al., 2020) consists two types of ARs (*support* and *attack*). Argumentative graphs in this dataset have a non-tree structure.

SciARG (Accuosto et al., 2021) employ a fine-grained annotation scheme consisting of six types of argument relations. The details can be obtained from Table 4.2. Argumentative graphs in this dataset have a tree structure.

AR types	Description
support	provides new supporting information/evidence
elaboration	provides additional information relevant to assess/contextualize
by-means	describe methods through which supporting evidence is obtained
info-required	provides information essential to understand/contextualize
sequence	describes a step that comes after a step described in a process
info-optional	provides non-essential information

Table 4.2: The definition of AR types on the SciARG dataset. Adopted from Accuosto et al. (2021).

4.4.2 Evaluation and Implementation

Evaluation For the AbstRCT dataset, we follow previous studies (Mayer et al., 2020; Si et al., 2022; Galassi et al., 2023) by merging *major claim* and *claim* into a single category, given that the number of major claims in the dataset is small. Previously proposed models that use this dataset (Mayer et al., 2020; Si et al., 2022; Galassi et al., 2023) carry out ARI and ARC simultaneously, i.e., they predict the existence of ARs and their types at the same time. Therefore, to facilitate comparison with results obtained by these models, we treat the ARC subtask on the AbstRCT dataset as a three-type (None, Support, Attack) directed relation classification task. Here, a true positive is an outcome where the model correctly predicts both the relation type and direction, given two ACs. We report the macro-averaged F1 scores for ARI, ARC and ACC subtasks. For the SciARG dataset, we report macro-averaged F1 score for the ARI subtask and weighted-averaged F1 scores for the ARC and ACC subtasks, in line with the literature (Accuosto et al., 2021). Galassi et al. (2023) treat the ARC subtask as an AC classification task, because each argumentative graph in this dataset is a tree. This means that given an AC, the task is to predict the AR between the AC and its (single) parent without access to information about its parents. Hence, we also treat the ARC subtask on this dataset as an AC classification task, rather than an AC pair classification task.

Implementation For the AbstRCT dataset, we use the same train-development-test split as used in (Si et al., 2022). Following (Accuosto et al., 2021), five-fold cross-validation is utilised on the SciARG dataset. We use the pre-trained cased version of SciBERT (Beltagy et al., 2019) for the AbstRCT and the SciARG datasets. We fixed the maximum sequence length at 512. A learning rate of 2e-5 is used for both datasets. The training epoch is set to 15 for the AbstRCT and the SciARG dataset. For the AbstRCT dataset, during the reading stage, a GCN with 6 layers is used for the ARI subtask. During the post-reading stage, the number of layers of the GCN is set to 2 for the ACC subtask and 6 for the ARC subtask. The hyperparameters of the GCN layer when applied to SciARG dataset are the same as those used for the AbstRCT dataset. We set the dropout rate to 0.5 for all the GCN layers. Due to GPU memory constraints, we set the batch size to 3 and accumulate gradients over 3 batches for AbstRCT dataset. For the SciARG dataset, the batch size is 2, and the gradient is accumulated over 4 batches. We report the averaged results of five different random seeds. Our model is implemented in PyTorch (Paszke et al., 2019) on a NVIDIA Tesla V100 GPU. The AdamW optimizer (Loshchilov and Hutter, 2019) is adopted for parameter optimisation.

4.4.3 Baselines

In order to evaluate our proposed method, we compare it with the following baselines. The following baseline models are based on the AbstRCT dataset:

RA (Galassi et al., 2018) is a residual network model combined with a long short-term memory (LSTM) network that jointly addresses the ACC, ARI and ARC subtasks.

RAA (Galassi et al., 2023) is an extension of the RA model that includes an attention module and ensemble learning. It is also a multi-task model which is designed to solve the ACC, ARI and ARC subtasks simultaneously. Both RA and RAA have an average and an ensemble version. The average version means that the final scores are the average scores of 10 different networks trained with 10 different seeds, while the ensemble version assigns each element as the class of the majority votes of the same 10 networks.

SeqMT (Si et al., 2022) implements a multi-task learning framework that leverages the sequential dependency between the ACC and relation identification [4] tasks by transferring the representation of the input and output of the ACC subtask to the relation identification task.

BERT-Trans (Bao et al., 2021) is a neural transition-based model designed for ACC and ARI subtasks. This model is also used on the SciARG dataset.

The following baseline models are based on the SciARG dataset:

SciARG_S (Accuosto et al., 2021) applies the standard method of considering the representation of the $[CLS]$ token from a Bert encoder, and feeding it into linear classifiers to obtain the predicted labels. This model solves ACC, ARI and ARC subtasks separately.

SciARG_M (Accuosto et al., 2021) is a multi-task model based on SciARG_S. This model deals with ACC, ARI and ARC subtasks at the same time. Here, the BERT encoder is shared among all the subtasks. For both SciARG_S and SciARG_M, two tricks are used during training. First, for the ARI subtask, in order to train the model with more positive examples, they sample it twice in the training set when a relation exists between two sentences. Moreover, additional features (the sentence positions in the abstracts as well as their relative distance and order) are included by adding special tokens to the standard BERT tokenizer to represent these features.

[4] A joint task that solves ARI and ARC subtasks simultaneously

4.5 Results

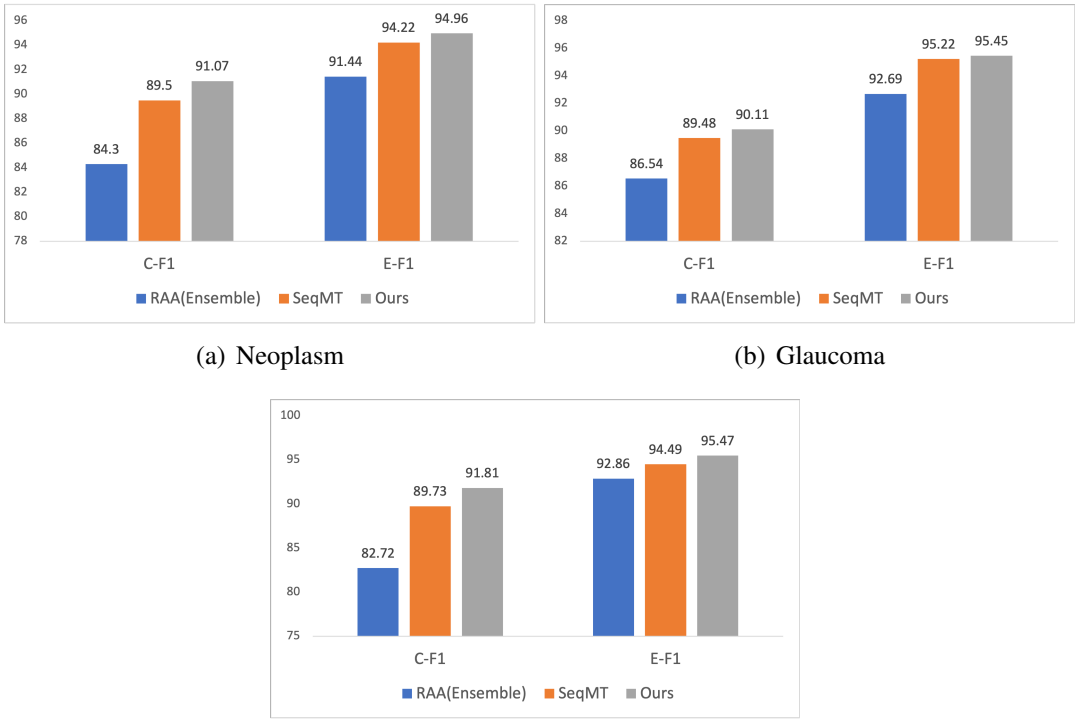
The experimental results are shown in this section. The overall results on the two datasets are given in Section 4.5.1. We evaluate the contribution of different types of global information utilised in our model in Section 4.5.2.

4.5.1 Overall Results

	ACC			ARC			ARI		
	Neo	Gla	Mix	Neo	Gla	Mix	Neo	Gla	Mix
RA(avg)	86.18	85.53	86.74	59.15	57.23	60.31	-	-	-
RA(Ensemble)	86.38	87.13	87.59	63.16	61.86	68.35	-	-	-
RAA(avg)	86.19	86.26	87.51	66.49	62.68	63.47	-	-	-
RAA(Ensemble)	87.87	87.71	89.70	70.92	68.40	67.66	74.49	74.80	72.70
BERT-Trans	91.85	90.76	91.50	-	-	-	76.80	75.06	76.16
SeqMT	91.89	92.35	92.21	71.24	**73.27**	72.71	-	-	-
GIAM (Ours)	**93.02***	**92.78†**	**93.64***	**74.80***	72.33	**73.31***	**78.94***	**78.77***	**77.54***

Table 4.3: Overall results on the AbstRCT dataset. Here, *Neo*, *Gla* and *Mix* correspond to the results achieved for the neoplasm, glaucoma and mixed test sets, respectively. The highest scores are emboldened. * and †indicates statistically significant improvements over the baselines compared to our model, according to a t-test with $p<0.05$ and $p<0.1$.

AbstRCT. Performance comparisons between our model and baselines are shown in Table 4.3. As can be observed, our model achieves state-of-the-art performance on all subtasks on both datasets, with the exception of the ARC subtask on the glaucoma test set. For the ACC subtask, we obtain the best performance on all three test sets. Our model achieves improvements of 1.13, 0.43 and 1.43 points in terms of F1-Score, for the neoplasm, glaucoma and mixed test sets, respectively. In Figure 4.5, we further analyse the ACC results in terms of performance on the the ifferent AC classes, i.e. *claim* and *evidence*. Although our model achieves improvements in F1 scores for both claim (C-F1) and evidence (E-F1), the improvement over other models is more substantial for the C-F1 scores. Specifically, in terms of C-F1, our model obtains improvements ranging from 0.67 for the glaucoma test set to 2.07 points for the mixed test set. In contrast, the average improvement for the E-F1 score for all three sets is only 0.65 points. This suggests that compared to premises, the identification and classification of claims is more dependent on global information, which provides a general understanding of the argument structure. For the ARC subtask, our model outperforms the current



(a) Neoplasm

(b) Glaucoma

(c) Mixed

Figure 4.5: F1-scores of different models for the ACC subtask, broken down according to class. C-F1 refers to F1-scores for claims and E-F1 refers to F1-scores for evidence.

state-of-the-art model by 3.56 points on the neoplasm test set and 0.6 points on the mixed test set. Meanwhile, on the glaucoma test set, the performance of our model is 0.94 points lower than the score achieved by SeqMT. According to the results of an ablation study of SeqMT (Si et al., 2022), it was found that removing information about predicted ACC labels resulted in an 8.43 points decrease on the glaucoma test set. In contrast, the influence of these labels on the neoplasm (-0.04) and mixed (-1.81) test sets was found to be negligible. This suggests that ARC performance is highly dependent on the explicit utilisation of ACC label information, which is not used for ARC classification in our framework. For the ARI subtask, we can only compare our model with RAA(Ensemble) because results (for this task) are available only for this model. Our results show that our proposed model outperforms the RAA(Ensemble) significantly.

SciARG. As shown in Table 4.4, our model achieves the best performance for all subtasks. Specifically, the improvements are 0.68, 7.46 and 3.51 points for the ARI,

ARC and ACC subtasks, respectively. Noteably, BERT-Trans gets a worse performance on both test sets for the ACC subtask. We suspect the reason to be twofold: Firstly, we re-use the best hyperparameters for another dataset reported in Bao et al. (2021), which may not be the best choice on this dataset. Secondly, unlike the dataset they used, the SciArg ACC subtask features eleven argument classes classification task with an imbalanced frequency distribution. Our results suggst that the BERT-Trans approach struggles in this setting.

	ARI	ARC	ACC
BERT-Trans	68.87	-	45.54
SciARG_S	69.93	72.21	67.97
SciARG_M	70.46	74.25	69.51
GIAM (Ours)	**71.14†**	**81.71***	**73.02***

Table 4.4: Overall results on the SciARG dataset. The highest scores are in emboldened. * and †indicates statistically significant improvements over the baselines compared to our model, according to a t-test with $p<0.05$ and $p<0.1$.

4.5.2 Ablation Study

		AbstRCT			SciARG
		Neo	Gla	Mix	
ARI	Ours(-multi-label)	75.82*	76.88*	74.93*	70.02*
	Ours(-subgraph)	78.12*	77.79*	76.63*	70.98
	Ours	**78.94**	**78.77**	**77.54**	**71.14**
ACC	Ours(-whole graph)	92.68	92.42	93.07†	69.89*
	Ours	**93.02**	**92.78**	**93.64**	**73.02**
ARC	Ours(-whole graph)	72.04*	70.07*	71.91*	79.73*
	Ours	**74.80**	**72.33**	**73.31**	**81.71**

Table 4.5: Results of ablation experiments on the AbstRCT dataset and the SciARG dataset. * and †indicates statistically significant improvements over the ablation experiments compared to our model, according to a t-test with $p<0.05$ and $p<0.1$.

To evaluate the contribution of different types of global information utilised in our model, we conducted ablation experiments as follows: **Ours(-multi-label)** refers to the use of a multi-class classifier to predict the ARI label of each AC in isolation during each QA turn, thus excluding sibling level information. **Ours(-subgraph)** excludes the use of subgraph-level information during QA turns. **Ours(-whole graph)** excludes whole graph level information.

The results are shown in Table 4.5. Without the multi-label setting to provide sibling level information, the performance of the model drops significantly on all five test sets, with decreases ranging from -1.06 to -3.12 points. In terms of subgraph information, it can be observed that its degree of positive influence depends the overall level of performance of our model on the ARI subtask. The greatest improvement when subgraph information is used is observed for the AbstRCT test set, while the smallest improvement occurs with the SciARG test set, for which the model achieves the lowest overall performance. By comparing the complete model with Ours(-whole graph), we find that the whole graph level information has a positive effect for both ARC and ACC subtasks on both datasets, especially for the ARC subtask, where the model improves by at least 1.41 points on the test set.

4.6 Discussion

We discuss the experimental results in this section. Further analyses regarding hyper-parameters as well as a case study and common errors are introduced in Section 4.6.2-Section 4.6.5.

4.6.1 Impact of Different Representations of the ROOT Query

As mentioned in Section 4.3.2, we use a natural language query "What is the main argument of the document?" as the ROOT query Q_R. However, this choice of representation is not obligatory. In this section, we explore the impact when using a pseudo query or a dedicated token as Q_R. Here, a pseudo question is not a complete question but only a phrase. It can be seen as a short representation of the natural language query. To be specific, we use "Main argument" as the pseudo question. Besides leveraging the pre-trained semantics, we also use a special token `<query>` with its embedding trained from scratch as the representation of Q_R. Although the representation of Q_R will affect the performance of all the ARI, ARC and ACC subtasks, the impact of the latter two is based on the performance of the ARI subtask. Thus, we only report the performance on the ARI subtask. The results are shown in Table 4.6.

Concretely, the natural language query performs best. Compared with the pseudo query and specific token query, the natural language query makes it easier for our model to capture the meaning of Q_R, because BERT is pre-trained on natural languages. The pseudo query can be regarded as a concise expression of the natural language query.

	Neo	Gla	Mix
special token	78.13	78.54	77.12
pseudo query	78.03	78.75	76.98
natural language query	**78.94**	**78.77**	**77.54**

Table 4.6: The impact of different types of the root query on the AbstRCT dataset.

Hence our model can still capture the meaning of it. The specific token query needs to be learned from scratch, due to the limitation of the training size, the model might not learn the rich semantic information as the natural language query, but it can still obtain similar results with the pseudo query which also features only part of the semantics.

4.6.2 Effects of Wrong Prediction of ROOT ACs

As mentioned above, we design a rule to define a ROOT AC for each argumentative graph as the starting point of the reading stage. However, the performance of the prediction for ROOT ACs is lower than the overall performance and the accuracy scores on the neoplasm, glaucoma and mixed test sets are 0.65, 0.62 and 0.64, respectively.

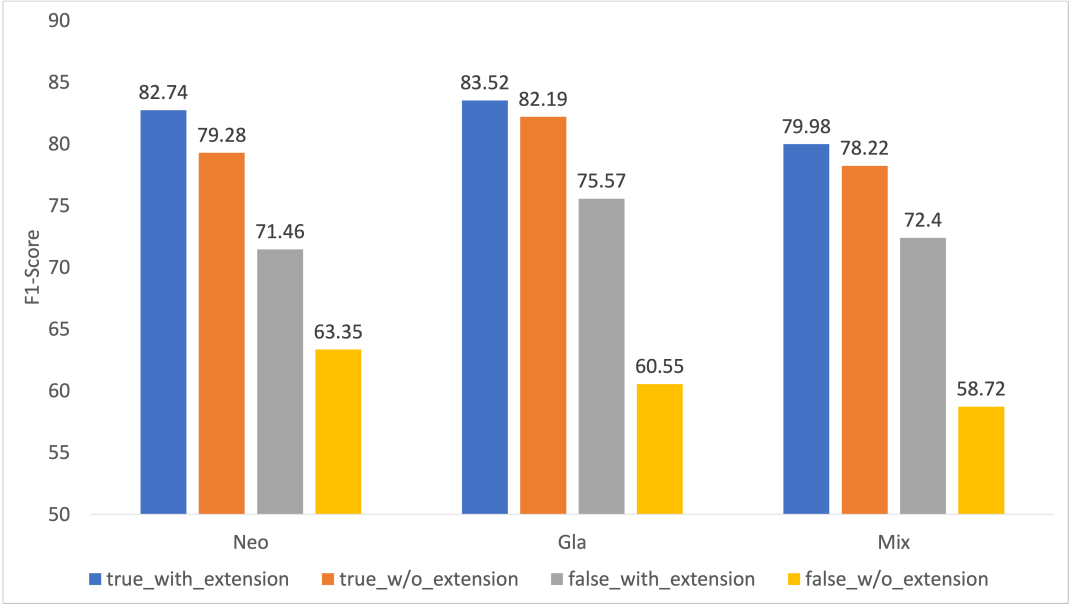


Figure 4.6: Error analysis results of ARI subtask. Here, *true* and *false* represent cases in which the ROOT AC is predicted correctly or incorrectly, respectively, while *with_extension* and *w/o_extension* denote whether the query extension setting is used or disabled

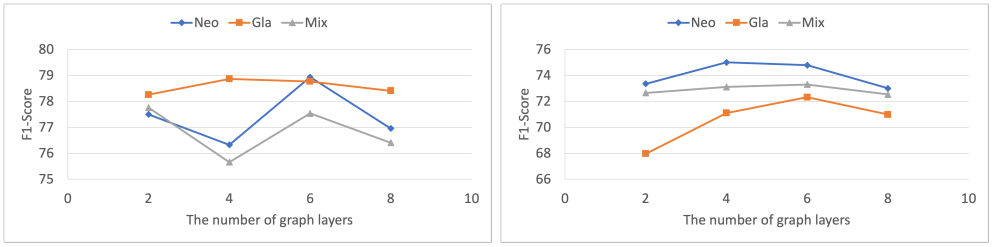
To further investigate the influence of the incorrectly predicted ROOT AC nodes on overall performance of ARI, Figure 4.6 separates the F1-Scores of the model in cases

where the ROOT AC node is predicted correctly (shown using the blue *true_with_extension* bars in the figure), and in cases where it is incorrectly predicted (shown using the grey *false_with_extension* bars in the figure). As expected, the model performs better when the ROOT AC is predicted correctly. In the opposite case the performance drops significantly (up to 11.28 points). Both results mentioned above make use of of the *query extension* setting, which was introduced in Section 4.3.5, and is aimed at alleviating the negative effects of incorrectly predicting the ROOT The query extension setting uses all ACs that remain in the *buffer* as queries when no new query is generated. This means that even if the ROOT AC was incorrectly predicted at the start of the reading stage, it is still possible that the correct ROOT AC will eventually be used as a query (provided that it remains in the buffer), and that at least some of its related AC nodes will be identified. To investigate whether the query extension setting has the intended effects, Figure 4.6 additionally shows the results obtained when this setting is disabled, both for cases in which the ROOT AC node is initially incorrectly predicted (shown using the yellow *false_w/o_extension* bars in the figure) and for cases in which the ROOT AC node is correctly predicted (shown using the orange *true_w/o_extension* bars in the figure). When the model predicts a wrong ROOT AC and the query extension is disabled, the average drop in performance is 12.27 points compared to when the query extension is used. This result serves to demonstrate the effectiveness of the query extension setting in reducing the negative effects of incorrect ROOT AC prediction. Furthermore, it is worthwhile noting that even when the ROOT AC node is correctly predicted, the use of the the query extension setting is advantageous (with improvements ranging from 1.33 points on the neoplasm test set to 3.46 points on the glaucoma test set, compared to when this setting is disabled). We likely observe the improvement because the setting is needed to allow the identification of additional unconnected subgraphs, which are sometimes a feature of complex argumentation structures.

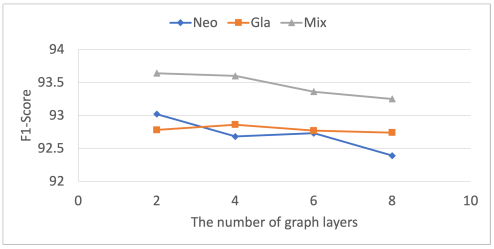
4.6.3 Hyper-parameter Analysis

4.6.3.1 Effect of Different Number of Graph Layers

In Figure 4.7, we illustrate the effects of using different numbers of graph layers (i.e., 2, 4, 6 and 8 layers) when our model is applied to the AbstRCT dataset. For the ACC subtask, the general trend is for performance to decrease as the numer of graph layers increases. In contrast, for the ARI and ARC subtasks, the performance of the model tends to improve asthe number of layers is increased. One possible reason is



(a) ARI (b) ARC



(c) ACC

Figure 4.7: The performance of our model with different number of graph layers.

that the relation-related tasks (ARI and ARC) need more global information than the component-related task (ACC).

4.6.3.2 Effect of Multi-label Thresholds

Additionally, we investigate whether varying the decision thresholds affects the performance of the ARI subtask during the reading stage. Figure 4.8 shows the impact of setting the threshold at different levels between 0.1 and 0.9. It is clear that the performance on the neoplasm test set is better when the threshold is low. In contrast, the model achieves better performance with a much higher threshold when applied to the glaucoma and mixed tests. Specifically, the best performance of the glaucoma and mixed test sets is achieved when threshold is set to 0.8. One possible reason is that the ARs are harder to identify in the neoplasm test set than in the other two test sets. According to the varying trends in the three different test sets, we set a common threshold of 0.5 for all datasets. As can be seen in Figure 4.8, using this threshold permits a reasonable level of performance in all of the test sets, alleviating the need for fine tuning for each individual dataset.

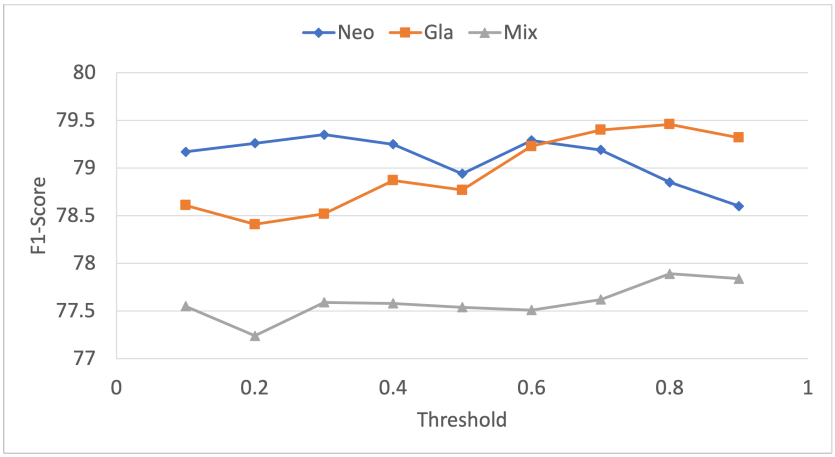


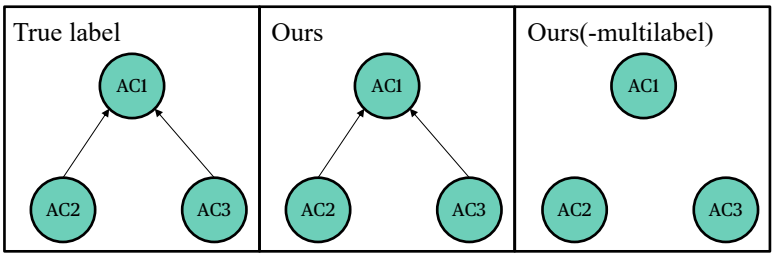
Figure 4.8: The performance of our model on the ARI subtask using different thresholds.

4.6.4 Case Study

To better analyse the influence of using sibling information to support the ARI subtask, Figure 4.9 illustrates an example from the AbstRCT dataset. It can be seen that AC2 concerns systolic blood pressure (SBP) while AC3 discusses diastolic blood pressure (DBP). Since AC1 contains references to both DBP and SBP, the argument only makes sense when considering all three ACs together. If sibling level information is ignored, and relations are learned only between individual AC pairs during training, then only part of the semantics is learned, and the relations among these ACs are missed, as shown in the depiction of the results from *Ours(-multilabel)*.

4.6.5 Error Analysis

We randomly selected 30 wrong predictions of the ARC subtask based on the results of the AbstRCT dataset. We find that there are two main types of errors. The first type of error might due to the lack of knowledge of some biomedical terms. For example, our model wrongly predicted a relation between the two ACs *"All of the patients on the DepoCyt arm but only 53% of those on the ara-C arm were able to complete the planned 1-month induction therapy regimen."* and *"DepoCyt injected once every 2 weeks produced a high response rate and a better quality of life as measured by Karnofsky score relative to that produced by free ara-C injected twice a week."* as support while there is no relation between these two ACs. The confusion might stem from a lack of clarity regarding the definitions of "induction therapy regimen", "high response rate" and "quality of life". In this case, we think a possible solution is to



AC1: Neither the SBP of the NISH patients nor the DBP of either group were similarly decreased, which indicated that ISH patients were more sensitive to salt restriction.

AC2: The mean SBP only decreased by 5.10 mm Hg (95% CI: -2.02 to 12.2, P=.158) in the NISH LSSalt group compared with that of the NISH NSalt group.

AC3: The mean diastolic blood pressure (DBP) had no significant differences in the ISH and NISH groups.

Figure 4.9: Case Study involving three ACs. From left to right are the gold standard, prediction by Ours and prediction by Ours(-multilabel).

leverage external knowledge as part of model architecture or training procedure. The second one is the lack of numerical reasoning ability. For example, our model missed the relations between the two ACs *"The rate of progressive disease was 47%, 21%, and 13% at the same dose levels, respectively."* and *"Bexarotene (Targretin capsules) (the first retinoid X receptor-selective rexinoid) was well tolerated and effective as an oral treatment for 15 (54%) of 28 patients with refractory or persistent early-stage cutaneous T-cell lymphoma at doses of 300 mg/m(2) per day."* because the model cannot capture the correct semantics of comparison.

4.7 Related Work

4.7.1 Argument Mining

Recently, the task of AM has received increasing attention from the research community, and a range of models has been proposed (Eger et al., 2017; Gemechu and Reed, 2019; Hewett et al., 2019; Lytos et al., 2019; Dutta et al., 2020; Liu et al., 2022; Cheng et al., 2022). The performance of these models may be influenced by various factors. For example, some approaches only utilise single ACs or a pair of ACs as input, without using wider contextual information from elsewhere in the document (Mayer et al., 2020; Accuosto et al., 2021). Although the model of Galassi et al. (2023) also uses such input,

they acknowledge that single propositions will not always contain sufficient information to predict argument components and relations, and thus suggest that additional document context may be required.

In contrast, other studies are able to make use of wider document context by supplying the whole text to their models. They approach the AM task in various ways, e.g., as a sequence tagging task (Eger et al., 2017); as a dependency parsing task (Ye and Teufel, 2021; Eger et al., 2017); using pointer networks (Potash et al., 2017; Wang et al., 2020) or using a prompt-based model (Dutta et al., 2022). More recently, Bao et al. (2022a) propose a generative framework with a constrained pointer mechanism for the end-to-end AM task.

Instead of involving more context information, some methods prefer to leverage external knowledge to enrich the semantic information between AC pairs. For example, Morio et al. (2022) transfer knowledge from different argument mining datasets. Rodrigues and Branco (2022) transfer knowledge from other natural language processing tasks, i.e., reasoning and comprehension. They find that the knowledge transfer enabled by the transfer learning from language processing tasks that are confluent to argument mining is an effective approach to improve neural argument mining. Saadat-Yazdi et al. (2023) directly leverage knowledge from the existing knowledge graphs to classify the type of argument relation between each AC pair. Even though external knowledge is used, they still only use the information between each AC pair but no global information is included.

A further type of variation among models concerns the use of different task-specific features. These features aim to encode various types of information about how ACs and/or ARs are typically expressed in text as a means to try to improve performance. They include the distance between two ACs (Galassi et al., 2023; Eger et al., 2017); the position of the AC in the document or paragraph (Potash et al., 2017; Bao et al., 2021); discourse parser features (Hewett et al., 2019); and zoning information (Liu et al., 2022). Hewett et al. (2019) examine the internal structure of ACs and find that they can typically be broken down into more fine-grained elements, whose similarities and differences can be used to help to determine argument structure.

Despite the previous employment of global information in the form of the whole document text, and/or different features relating to the discourse/argument structure, there are few approaches that try to explicitly exploit the global *argument structure* of the document, i.e., using information about how the about wider argument structure of a document could impact upon the most appropriate predictions for different AM

subtasks. To our knowledge, the only work that uses such information is that of Si et al. (2022), where they aggregate information about related AC pairs to support its predictions. In their study, *related* ACs are those that support or attack a common AC (i.e., ACs that are siblings of each other in the argumentative graph). However, performance depends on the manually defined continuous window size. We extend upon this approach, in that we do not use a fixed window size, and we use all available information from the constructed argumentative graph, rather than only sibling-level information.

4.7.2 Machine Reading Comprehension

Our proposed framework also incorporates ideas from recent research into the task of machine reading comprehension (MRC), whose aim is to find an answer for a query based on a given context. While a large number of approaches to this task have been proposed (Liu et al., 2023c; Malhas and Elsayed, 2022; Zhang et al., 2022), a recent trend that has achieved great success has been to cast various other NLP tasks as MRC tasks. For example, Levy et al. (2017) demonstrate that by formulating relation extraction (RE) as an MRC task, it is possible to achieve zero-shot generalisation on unseen relation types. Meanwhile, by treating named entity recognition (NER) as an MRC task, Li et al. (2019) are able to obtain superior performance in identifying nested named entities. Going a step further, Li et al. (2019) propose the use of a multi-turn QA-based model to carry out NER and RE simultaneously. MRC techniques have also been applied to the more complex task of event extraction by Liu et al. (2020). Their proposed model generates a set of natural questions from an event schema in an unsupervised manner; the answers to these questions are then retrieved as the event extraction results. The employment of MRC in addressing coreference resolution has also been investigated (Wu et al., 2020).

Considering that treating other NLP tasks as an MRC task could improve the performance, more recently, Bao et al. (2022b) proposed an MRC-based model for the task of argument pair extraction, which involves extracting related pairs of sentences from two inter-related documents that discuss the same issues. We adopt a similar approach, but we make use of MRC techniques to identify relations among ACs within the scope of a single document.

4.8 Summary

In this chapter, we proposed a two-stage model that leverages graph-level argument-specific structural features to perform subtasks of AM since argumentative texts in biomedical abstracts usually have argument-specific structural features and graphs are direct to represent such features. In the first stage, the (sub-)graph level argument-specific structural features are used as to enhance performance of the ARI subtask, using a multi-turn QA-based model to predict an initial argumentative graph. In the second stage, the structure of the whole initial graph is then utilised as (whole) graph level argument-specific structural features for other subtasks.

We have demonstrated the effectiveness of our model and the importance of these three types of information through their evaluation and comparison to other approaches using publicly available datasets, showing that our approach achieves state-of-the art performance in most cases.

Based on further analysis in Section 4.6.2, we found that one limitation of our model is that the ROOT AC is not always accurate, which can lead to error propagation and have negative impact upon the performance of our model. Although we have attempted to alleviate this issue through the use of a customised query extension setting, the performance is still lower when the ROOT AC is predicted wrongly compare to that when the ROOT AC is predicted correctly. We intend to avoid this pipeline way by designing another method to leverage the argument-specific structural features in the next chapter.

Chapter 5

Argument Mining with Path-level Argument-specific Structural Features

In the previous chapter, we introduced our model GIAM, which allows the use of graph-level argument-specific structural features to help the model pay attention to global argument structure information. Our experimental results showed that such graph-level argument-specific structural features are helpful for AM. However, the GIAM model suffers from the error propagation problem, which impacts performance when a wrong (sub)graph is predicted. To address this issue, we model argument-specific structural features from another perspective in this chapter.

Specifically, an argumentative graph that consists of argument components and argument relations contains complete information about an argument and reveals the logic of the argument. Therefore, the argument structure of an argumentative text can be considered as an answer to a "why" question , i.e., why the main conclusion of a biomedical abstract is correct. In this way, the entire argument structure is similar to the concept of a "chain of thought," i.e., the sequence of ideas that lead to a specific conclusion for a given argument (Wei et al., 2022). As these "chains" are represented as paths, we refer to this type of argument-specific structural feature as a path-level argument-specific structural feature, which is related to our last research question (**RQ3**).

To enable the model to learn path-level argument-specific structural features, we propose a new perspective that transforms the argument mining task into a multi-hop reading comprehension task. This approach enables the model to identify these structures by not only predicting the answer to a given query but also analysing the logical sequence that leads to the answer.

We conduct a comprehensive evaluation of our approach on two argument mining benchmarks and observe that we surpass state-of-the-art results. Finally, we present a detailed analysis to illustrate the significance of the ”chain of thought” information, showcasing its helpfulness for the argument mining task. This chapter is drawn from our publication Liu et al. (2023a).

5.1 Motivation

The argument structure of an argumentative text can be regarded as an answer to a “why” question; therefore, the whole argument structure is similar to the “chain of thought” concept—spelling out a sequence of reasoning steps (forming a reasoning path) that lead to a specific conclusion for a given argument (Wei et al., 2022). For argumentative texts of the same genre, the structure of such “chains of thought” tends to be similar: In a student essay, there is usually a major claim supported by several other claims, and then a number of premises which are related to the claims (Eger et al., 2017). Similarly, for scientific abstracts such as shown in Figure 5.1, premises about experimental results are used to support aspect-based claims (i.e., AC5 in Figure 5.1 mentions about three aspects, postoperative IOP, bleb morphology, and complications), which are further used to support a high-level claim (AC6 in Figure 5.1) as the conclusion of the abstract. Such “chains of thought” is also called path-level argument-specific structural features in this chapter. However, most of the previous works (Mayer et al., 2020; Rodrigues and Branco, 2022; Saadat-Yazdi et al., 2023) ignore such structural similarity and mainly pay attention to single ACs for the AC-related subtasks and AC pairs for the AR-related subtasks.

To enable the model to learn such chain and extract argument structure simultaneously, we propose to convert AM into a generative multi-hop machine reading comprehension (MRC) task (Yavuz et al., 2022). It is a sequence to sequence task where the input sequence is a combination of a query as well as a context and the output sequence is the answer with the reasoning path about how the model gets the answer. Concretely, given an AC as a *query* and the whole text as the *context*, our approach predicts both the reasoning *path* and the type of the AC or which are the other ACs related to such AC according to the different subtasks of argument mining as *answer*. Here, the reasoning path is a path from the `ROOT` AC to the query AC, where the `ROOT` AC is defined as the source node of the longest path in an argumentative graph, which usually contains the core opinion of an argument, such as the main conclusion

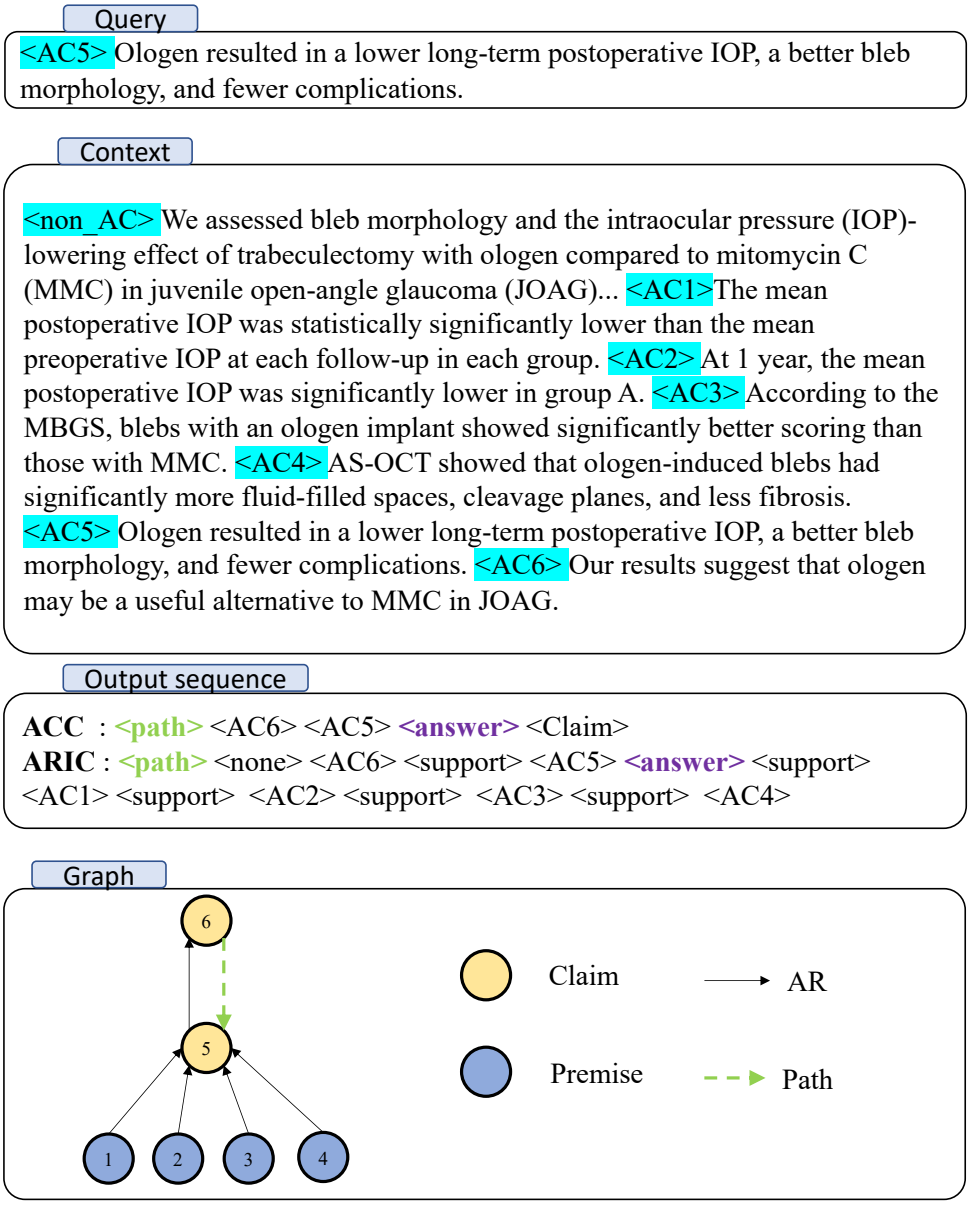


Figure 5.1: An example of how our model works on the AbstRCT dataset. Given AC5 as a query and the whole abstract as the context, the output sequence is the combination of the predicted path (the tokens between the special tokens $< path >$ and $< answer >$) and the answer (the tokens followed by $< answer >$). In this example, the path contains two ACs, the ROOT node AC6 and the query node AC5. The answer differs from the subtask, i.e., for the ACC subtask, the answer is $< Claim >$ which means the type of AC5 is claim; for the ARIC subtask, the answer means that AC1, AC2, AC3 and AC4 are ACs that support AC5. The whole argumentative graph of the context in the right part can be obtained after all ACs are used as queries. All relation types in this graph are "support".

of the abstract. An example is shown in Figure 5.1. It is worth mentioning that the direction of the path is reversed from the AR so that the model can learn the high-level semantics first. The reason that we choose the generative paradigm is that only the output is changed and no extra parameters are included to let the model leverage the path information. Therefore, we can directly test the impact of the path information.

Furthermore, to alleviate the bias of answers' order learned by the generative model, we propose a bi-direction-based method that allows the model to learn the output sequence from both directions.

To summarise, our contributions are as follows:

- to our best knowledge, we are the first to transfer the argument mining task into the multi-hop MRC task to let the model learn the path-level structural features;
- to alleviate the bias of order learned by the generative model, we propose a bi-direction-based method;
- extensive experimentation on two different benchmark datasets shows that our framework outperforms related models on most subtasks

This chapter is organised as follows: Section 5.2 introduces how to learn path-level structural features by modeling the AM task as a multi-hop MRC task. Section 5.3 discusses experimental details. In Section 5.4, we evaluate our model and compare it with other models. Analysis of the results is presented in Section 5.5, with the following discussion of related work in Section 5.6. Some conclusions are provided in Section 5.7.

5.2 Method

Following previous works (Si et al., 2022; Galassi et al., 2023), we assume that the ACI subtask is solved and we only focus on the other subtasks. Thus, there are two ways to combine the remaining subtasks, i.e., ACC + ARIC and ACC + ARI + ARC. Considering there is no established preference in existing literature—the former is used on the AbstRCT dataset (Mayer et al., 2020; Si et al., 2022; Galassi et al., 2023) and the latter is used on the SciARG dataset (Accuosto et al., 2021; Liu et al., 2023b)—we focus on all four subtasks to enable a fair comparison.

In this section, we introduce how to transfer argument mining into a multi-hop generative MRC task. The multi-hop MRC consists of four parts $< query, context, answer, path >$, where the input sequence is the query and the context, and the output sequence is the

answer and the path. We introduce the details of these two sequences in the following sections .

5.2.1 Input Sequence

The input of the model is a word sequence that contains the *context* and the *query*.

Context Representation Let $DOC = \{t_1, t_2, ..., t_l\}$ denote an argumentative document, where t_i represents the i-th token in DOC. As mentioned in Section 5.1, we assume that the position of all ACs in the document is known. The context representation consists of all tokens in the document, with additional tokens inserted to denote the boundaries of each AC. Specifically, we insert an $< AC\_i >$ token before the start token of the i-th AC token sequence in document D and a $< non\_AC >$ token before the first token of all non-argumentative sequences. The context sequence C is shown below:

$$C =< AC\_1 > t_1, t_2... < non\_AC > ...t_i... \tag{5.1}$$

Query Representation We use the ACs tokens as queries. There are two types of query representation Q according to the subtasks: a unary query Q_u that consists of one AC for the ACC, ARI and ARIC subtasks and a binary query Q_b which is composed of two ACs separated by the special tokens for the ARC subtask:

$$\begin{aligned} Q_u &= t_{AC_{i_1}}, ..., t_{AC_{i_n}} \\ Q_b &=< AC_i > t_{AC_{i_1}}, ..., t_{AC_{i_n}} \\ &< AC_j > t_{AC_{j_1}}, ..., t_{AC_{j_n}} \end{aligned} \tag{5.2}$$

Finally, the input sequence is the concatenation of the query Q and the context C differentiated by two specific tokens $< query >$ and $< context >$.

$$Input =< query > Q < context > C \tag{5.3}$$

5.2.2 Output Sequence

The output sequence of a given query consists of two parts, the *answer* sequence and the *path* sequence.

Answer Representation

The answer representation differs for different subtasks. The details are shown in Table 5.1. For the ACC subtask, given an AC as query, the model needs to predict the

	Path Representation	Answer Representation
ACC	$< AC_{p_1} > .. < AC_{p_n} >$	$< ACT_i >$
ARI	$< AC_{p_1} > .. < AC_{p_n} >$	$< AC_{a_1} > ... < AC_{a_n} >$
ARC	$< AC_{p_1} > .. < AC_{p_n} >$	$< ART_i >$
ARIC	$< ART_1 >< AC_{p_1} > ... < ART_n >< AC_{p_n} >$	$< ART_1 >< AC_{a_1} > ... < ART_n >< AC_{a_n} >$

Table 5.1: The path and answer representations for different subtasks. Here, $< AC_{p_1} >$ is the ROOT AC and $< AC_{p_n} >$ denotes the query AC; $< AC_{a_1} > ... < AC_{a_n} >$ are ACs that point to the query AC; $< ACT_i >$ and $< ART_i >$ represent the AC type and the AR type.

type of the AC. Thus, the answer for the ACC subtask is a specific token $< ACT_i >$ representing the AC type, where $< ACT_i >$ belongs to $< AC\_TYPE >$. The exact tokens included in $< AC\_TYPE >$ dependent on the annotation scheme of different datasets.

Similar to the ACC subtask, given a pair of ACs with a relation between them, the answer for the ARC subtask is a specific token $< ART_i >$ representing the AR type, where $< ART_i >$ belongs to $< AR\_TYPE >$.

For the ARI subtask, the answer of a given query contains all ACs that are related to it. In this case, the answers for the query q is a sequence of ACs:

$$A =< AC_{a_1} > ... < AC_{a_n} > \tag{5.4}$$

where $< AC_{a_1} > ... < AC_{a_n} >$ are ACs that point to the query AC.

As for the ARIC subtask, given an AC as a query, the model needs to predict all the ACs related to it and the relation types at the same time. Therefore, the ARIC subtask on the AbstRCT dataset is defined as a three-type (None, Support, Attack) directed relation classification task. Here, a true positive is an outcome where the model correctly predicts both the relation type and direction, given two ACs. Therefore, the answer contains both the type tokens $< ART_i >$ and the AC tokens $< AC_{a_i} >$.

$$A =< ART_1 >< AC_{a_1} > ... < ART_n >< AC_{a_n} > \tag{5.5}$$

Path Representation A reasoning path starts from the ROOT node and ends at the query AC. We define $< AC_i >$ as ROOT when it satisfies the following: first, $< AC_i >$ does not point to any other $< AC_j >$; second, there is a path that starts with $< AC_i >$ and the length of the path is the longest among all the paths. The path representations are shown in Table 5.1. To be specific, we use the path representation without relation types

for the ACC, ARI and ARC subtasks.

$$P =< AC_{p_1} > ... < AC_{p_n} > \tag{5.6}$$

where $< AC_{p_1} >$ is the ROOT AC and $< AC_{p_n} >$ denotes the query AC. For the ARIC subtask, we also include type information in the path representation to align it with the answer representations.

$$P =< ART_1 >< AC_{p_1} > ... < ART_n >< AC_{p_n} > \tag{5.7}$$

The final output sequence is the combination of the path sequence and the answer sequence denoted by two specific tokens $< path >$ and $< answer >$.[1]

$$Output =< path > P < answer > A \tag{5.8}$$

5.2.3 Output Order Debias

One thing we need to consider is the order of the output sequence since the output part is composed of special tokens which are used to represent AC rather than natural language in input text which already follows a fixed order. Since the order of the path is fixed, we directly follow the order of the path for the tokens in the path sequence. As for the answer part of the ARI and ARIC tasks where the order of the answer is not fixed, one method is to keep the serial numbers of ACs are based on the order of the appearance of ACs in the documents. However, if we keep the same ascending order ($< AC_1 >, < AC_2 >, \ldots$) of the answers, the model might learn an order bias, which may hinder the generation of $< AC_j >$ when $< AC_{i>j} >$ is generated, resulting in "unrecoverable" errors.

To alleviate this issue, we propose a simple but effective *two-direction* augmentation method. Concretely, for each ARI and ARIC query, we create two training samples with different answers, one sorted by the ordinal number of the AC tokens appearing in the sequence and the other one sorted in reverse.

[1] Incorporating the whole graph rather than the path information might incorporate additional global information, but it performed worse in initial experiments (see Appendix 5.5.4).

5.2.4 Training and Inference

Training The output sequence contains two parts, path and answer, this is a form of multi-task paradigm that includes the learning of both sequences jointly. Thus, during training, we calculate the loss of each part separately and then sum the two as our final loss function.

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{path} + \lambda\mathcal{L}_{answer} \tag{5.9}$$

where $\mathcal{L}$ is the cross entropy loss function, λ denotes the weights of the $\mathcal{L}_{answer}$ and λ takes values in increments of 0.1, starting from 0.1 up to 0.9.

To let the model leverage the information from pre-trained language models, we employ a warm start strategy inspired by Guo et al. (2022). To be specific, we use the embedding of number i as the initial representation for the specific token $< AC_i >$ instead of training the embedding from scratch. The model will cost more time and the training will be unstable if the model learns the embeddings from scratch during fine-tuning. The representation of the context is kept the same for all four subtasks.

During the training phase, we optimise the negative loglikelihood using teacher forcing.

Inference During the inference, we use beam search decoding to get the output sequence $Output$ in an autoregressive manner. We then post-process the decoded sequence using the answer indicator ($< answer >$) to obtain the answer and convert the output sequence into labels according to their meanings described in Section 5.2.2.

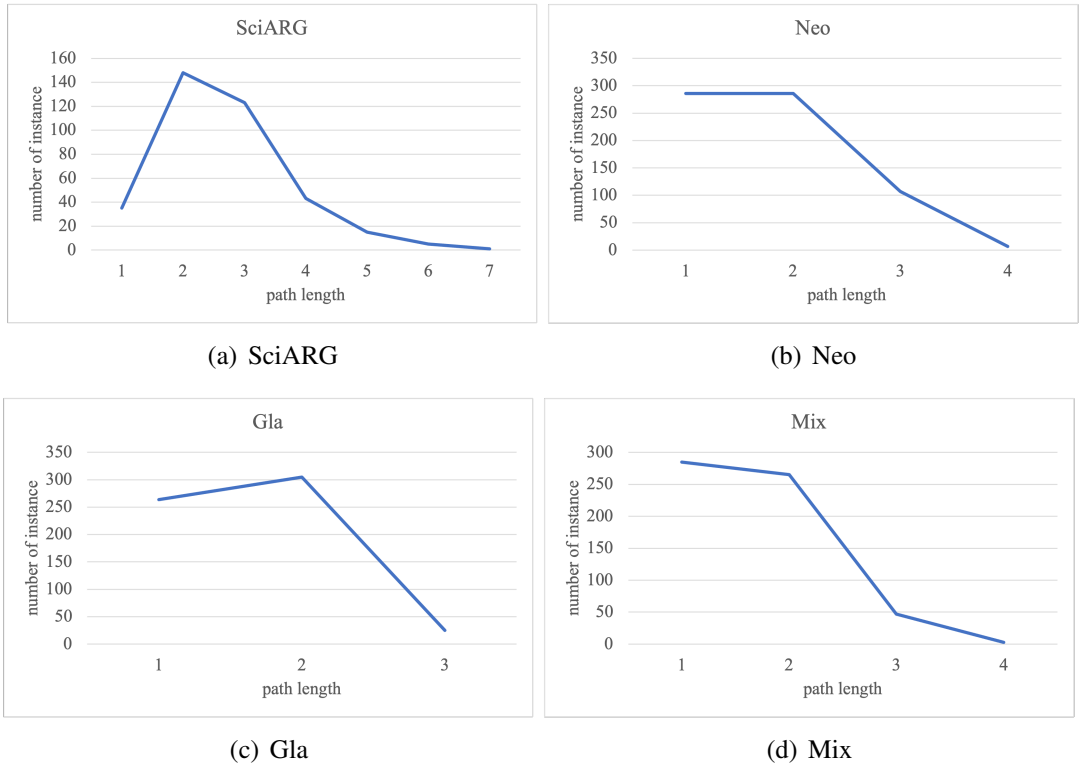
5.3 Experiments

5.3.1 Dataset

We still use two publicly available datasets AbstRCT and SciARG to evaluate our model and compare it with results obtained by previously proposed models. The path length distribution information of these two datasets are shown in Figure 5.2.

5.3.2 Evaluation and Implementation

For the AbstRCT dataset, we follow previous studies (Mayer et al., 2020; Si et al., 2022; Galassi et al., 2023) by merging *major claim* and *claim* into a single category. All results on both datasets are averaged scores of three different random seeds and



(a) SciARG (b) Neo

(c) Gla (d) Mix

Figure 5.2: The path length distribution information on different test sets.

are reported as macro-averaged F1 scores. For the AbstRCT dataset, we use the same train-development-test split as Si et al. (2022). For the SciARG dataset, we keep the same train-test split and randomly select 10% of the training set for validation, like Bao et al. (2021) do. We fine-tune BioBART-Base models (Lewis et al., 2020) for both datasets. Regarding the learning rate, we set it to 3e-5 for the ACC and ARC subtasks, 2e-5 for the ARI subtask, and 8e-5 for the ARIC subtask. The max sequence length is 768 for both datasets. The batch size is 16, and we assign a value of 0.7 to the hyperparameter λ. During inference, we employ beam search with a beam size of 4 for decoding purposes. To optimize our model, we employ AdamW (Loshchilov and Hutter, 2019). We train our model 15 epochs except for the ARC subtask which is 20 epochs and select the best checkpoint on the development set. We also fine-tune Llama-3.1-8B-Instruct[2] to explore whether our method is suitble for large language models. The models are trained based on the AdamW optimizer (Loshchilov and Hutter, 2019) for three epochs, using DeepSpeed (Rasley et al., 2020) to reduce memory usage. We set the batch size to 32. The initial learning rate is set to 1e-6 with a warm-up ratio

[2]https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

Subtask	Prompt Template
ACC	Task: Given an argumentative text that the argument components are labeled using special tokens (¡ACi¿) and an argument component, classify the type of the argument component and generate the reasoning path to it.
ARI	Task: Given an argumentative text that the argument components are labeled using special tokens (¡ACi¿) and an argument component, determine all argument components that are related to it and generate the reasoning path to it.
ARC	Task: Given an argumentative text that the argument components are labeled using special tokens (¡ACi¿) and a pair of argument components, label the type of this argument relation and generate the reasoning path to the parent argument component.
ARIC	Task: Given an argumentative text that the argument components are labeled using special tokens (¡ACi¿) and an argument component, determine all argument components that are related to it, label the type of each argument relation and generate the reasoning path to it.

Table 5.2: Prompts used for each subtask.

of 5%. All models are trained on one Nvidia Tesla A100 GPU with 80GB of memory. The prompts used for each subtask is shown in Table 5.2.

5.3.3 Baselines

We compare our method it with the following baselines on AbstRCT:

ResArg (Galassi et al., 2018) is a residual network model combined with a long short-term memory (LSTM) network that jointly addresses the ACC, ARI and ARIC subtasks.

ResAttArg (Galassi et al., 2023) is an extension of the ResArg model that includes an attention module and ensemble learning. Both ResArg and ResAttArg have an average and an ensemble version.

SeqMT (Si et al., 2022) implements a multi-task learning framework that leverages the sequential dependency between the ACC and ARIC subtasks by transferring the representation of the input and output of the ACC subtask to the ARIC subtask.

BERT-Trans (Bao et al., 2021) is a neural transition-based model designed for ACC and ARI tasks. This model is also used on the SciARG dataset.

For SciARG, we compare with the following baseline approaches:

SciARG_S (Accuosto et al., 2021) applies the standard method of considering the representation of the $[CLS]$ token from a Bert encoder, and feeding it into linear

classifiers to obtain the predicted labels. This model solves ACC, ARI and ARC subtasks separately.

SciARG_M (Accuosto et al., 2021) is a multi-task model based on SciARG_S. This model deals with ACC, ARI and ARC subtasks at the same time. Here, the BERT encoder is shared among all the subtasks.

5.4 Results

5.4.1 Main Results

	ACC			ARIC		
	Neo	Gla	Mix	Neo	Gla	Mix
ResArg(avg)	86.18	85.53	86.74	59.15	57.23	60.31
ResArg(Ensemble)	86.38	87.13	87.59	63.16	61.86	68.35
ResAttArg(avg)	86.19	86.26	87.51	66.49	62.68	63.47
ResAttArg(Ensemble)	87.87	87.71	89.70	70.92	68.40	67.66
SeqMT	91.89	92.35	92.21	71.24	73.27	72.71
GIAM	93.02	92.78	93.64	74.80	72.33	73.31
MRC_GEN(Bio-BART)	92.76	92.62	93.97	74.97	74.28*	73.87*
MRC_GEN(Llama3.1-8b)	**94.30***	**94.11***	**95.02***	**77.35***	**76.89***	**76.45***

Table 5.3: Overall results on the AbstRCT dataset. Here, *Neo*, *Gla* and *Mix* correspond to the results achieved for the neoplasm, glaucoma and mixed test sets, respectively. The highest scores are emboldened. * indicates statistically significant improvements over the baselines compared to our model, according to a t-test with $p<0.05$.

	ARI	ARC	ACC
BERT-Trans	68.87	-	45.54
SciARG_S	69.93	72.21	67.97
SciARG_M	70.46	74.25	69.51
GIAM	71.14	81.71	73.02
MRC_GEN(Bio-BART)	71.96*	82.35*	72.24
MRC_GEN(Llama3.1-8B)	**75.32***	**86.33***	**76.91***

Table 5.4: Overall results on the SciARG dataset. The highest scores are in emboldened. * indicates statistically significant improvements over the baselines compared to our model, according to a t-test with $p<0.05$.

Performance comparisons between our model and the baselines are shown in Table 5.3 and Table 5.4. Our model achieves SOTA results on most of the tasks for

both datasets, even though our model is a fine-tuned BART-Based model, while other baselines are considerably more complex than ours. It is evident that with the assistance of large models, our method can achieve better performance.

However, we observe that for both datasets, our improvement is smaller compared with GIAM which is proposed in Chapter 4. This aligns with our expectations, as both models fundamentally leverage argument-specific structural features in different ways, each with its own strengths and weaknesses. GIAM utilises graph-level argument-specific structural features to model argument structure, providing a broader perspective, especially in addressing type classification subtasks. However, it encounters issues of error propagation when constructing these graph-level argument-specific structural features. In contrast, MRC_GEN does not face error propagation issues, but it relies solely on path-level argument-specific structural features, lacking some information compared to the graph-level ones.

5.4.2 Ablation Study

	ACC			ARIC		
	Neo	Gla	Mix	Neo	Gla	Mix
MRC_GEN	92.76	92.62	93.97	**74.97**	**74.28**	**73.87**
MRC_GEN(-path)	92.29	92.27	93.72	72.47*	70.10*	71.20*
MRC_GEN(-td)	-	-	-	74.29	71.90*	73.50
MRC_GEN(-ws)	**92.83**	**92.72**	**94.11**	74.23	70.12*	73.09*

Table 5.5: Results of ablation experiments on the AbstRCT dataset. MRC_GEN(-path) denotes that the model only needs to predict the answer without the path information; MRC_GEN(-td) means that the two-direction method is excluded; MRC_GEN(-ws) uses a cold start method and the specific tokens are trained from scratch. The highest scores are in emboldened. * indicates statistically significant improvements over other ablation models compared to our model, according to a t-test with $p<0.05$.

We perform ablation experiments to investigate the effect of our method design on the overall performance on both benchmarks. There are mainly three models for the ablation study, MRC_GEN(-path) is designed to test whether the path information can improve the performance; MRC_GEN(-td) aims to test the effect of the two-direction method; and MRC_GEN(-ws) is for exploring the impact of warm start. Since the two-direction method is only suitable for the ARI and ARIC subtasks where the answer is a sequence of special tokens, MRC_GEN(-td) can not be applied to the ACC subtask. The results are shown in Table 5.5 and Table 5.6.

	ARI	ARC	ACC
MRC_GEN	**71.96**	**82.35**	**72.24**
MRC_GEN(-path)	70.64*	79.76*	71.28*
MRC_GEN(-td)	71.46	-	-
MRC_GEN(-ws)	70.73*	81.29*	71.39*

Table 5.6: Overall results on the SciARG dataset. MRC_GEN(-path) denotes that the model only needs to predict the answer without the path information; MRC_GEN(-td) means that the two-direction method is excluded; MRC_GEN(-ws) uses a cold start method and the specific tokens are trained from scratch. The highest scores are in emboldened. * indicates statistically significant improvements over other ablation models compared to our model, according to a t-test with $p<0.05$.

Two-Direction Method. Comparing the results of MRC_GEN(-td) and MRC_GEN, reveals that our two-direction method works for the ARI and ARIC subtasks. To explain this observation, we calculate the percentage of examples where the order of answer AC tokens is strictly ascending. For the ARIC subtask on the AbstRCT dataset, the percentage is 27.09%. The proportion on the SciARG dataset for the ARI subtask is 19.53%. This shows that forcing the model to learn only from examples in ascending order may inhibit its ability to generate the correct answer.

Warm Start. The warm start method improves the performance in most cases, in line with literature (Guo et al., 2022). Leveraging pre-trained embeddings as starting points for newly-added tokens is better than training their embeddings from scratch, since the size of the dataset for fine-tuning is much smaller than that of the pre-training dataset. Therefore, it is difficult for the model to fully learn the semantics of the new tokens only from the fine-tuning data. However, from Table 5.5, we also find that warm start is clearly hurting the ACC task on AbstRCT. From an intuitive point of view, each AC token needs to include two types of information: the location of the AC and the content of the AC. The warm start method only includes location information since we use the embedding of the numbers as a starting point for each AC token, while the model also needs to learn that this token is about the content of the AC. However, overlearning the representation from the warm start method may also decrease the performance. Therefore, good hyper-parameters are also important to let the model learn a balanced representation of these two types of information. Thus, we believe that the hyper-parameters for the ACC subtask on the SciARG dataset are good enough to learn a balanced representation.

Path Information. In general, without path information, the scores drop for all tasks on both datasets. The decrease is more obvious on the relation-based tasks such as

ARI, ARC and ARIC as opposed to ACC. Specifically, for the ACC task, the average improvement from the path information on all three test sets on the AbstRCT and SciARG datasets is 0.36 points and 0.96 points, respectively. One possbible reason is that the path information is more related to the AR-related subtasks compared with the AC-related ones as the path can be regarded as a chain of ARs. As for the ARI and ARC subtasks, generating path information results in improvements of 1.32 and 2.59 points on the SciARG dataset, which shows the positive impact of the path information and indicates that the "chain of thought" method is useful for the argument mining task. The path information is most important for the ARIC subtasks. Here, without the path information, the performance drops 3.1 points on average, possibly because the ARIC subtask is a combination of the ARI and ARC subtasks which both benefit from path information.

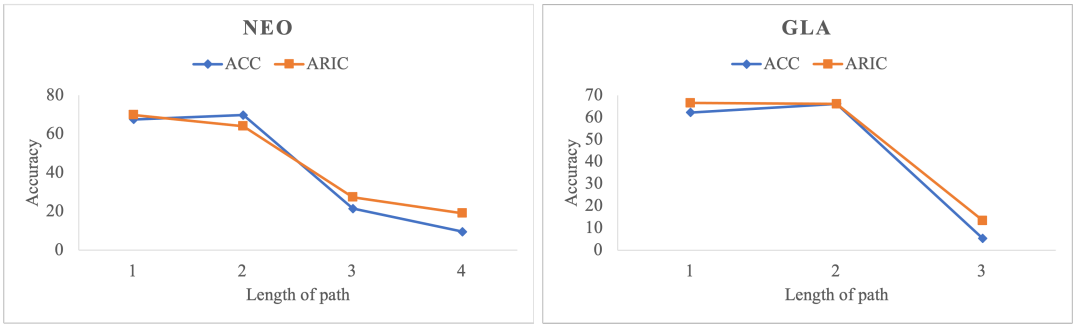
5.5 Analysis

5.5.1 Impact of the Path

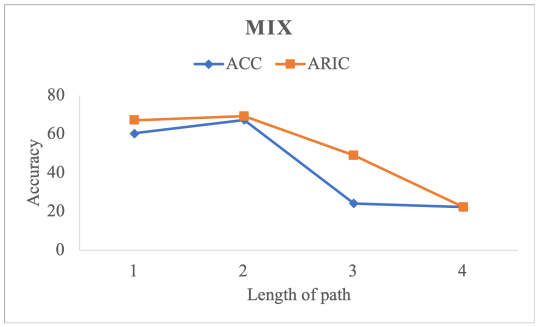
In Section 5.4.2, we only show the general impact of the path for the argument mining task by comparing model performance with and without path information. Here, we examine the effect of path information.

First, we calculate the accuracy of the predicted path. Here, only exactly matching true and predicted paths are treated as a correct instance. We report the accuracy according to the length of the path. For the AbstRCT dataset, the path length ranges from 1 to 4. As for the SciARG dataset, apart from path lengths 1 to 3, we introduce a new class called "4+" to represent instances with a path length greater than or equal to 4, given the limited number of instances with longer path lengths. It is clear in Figure 5.3 and Figure 5.4 that an increasing path length leads to significant drops in prediction accuracy. This is in line with our expect that longer paths are harder to learn for the model. Another interesting phenomenon is that the overall accuracy on the ARI and ARIC subtasks is higher than that on the ACC subtask. One possible reason is that the path prediction subtask requires the comprehension of ARs, and thus might benefit from the ARI and ARIC subtasks.

Furthermore, we explore whether correctly predicting the path indeed improves the overall task performance. Therefore, we compare the results of MRC_GEN with



(a) NEO (b) GLA



(c) MIX

Figure 5.3: Accuracy of the predicted path on the AbstRCT dataset. Here, *1*, *2*, *3* and *4* refer to the length of the path.

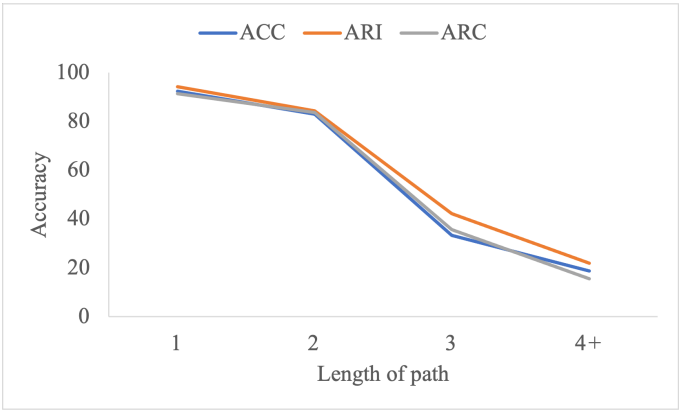


Figure 5.4: Accuracy of the predicted path on the SciARG dataset. Here, *1*, *2*, *3* and *4+* refer to the length of the path and *4+* means a path length greater than or equal to 4.

	ACC	ARI	ARC
Wrong_path	-2.97	-1.93	-2.34
Correct_path	+5.38	+3.47	+4.14

Table 5.7: The difference between MRC_GEN and MRC_GEN(-path) on the SciARG dataset when the path is predicted correctly or wrongly. Here, (Wrong/Correct)_path means that the path is predicted wrongly/correctly. Positive values mean that the path information improves the performance.

		ACC	ARIC
NEO	Wrong_path	+1.07	+2.86
	Correct_path	+0.30	+2.45
GLA	Wrong_path	-1.49	+11.52
	Correct_path	+0.66	+4.75
MIX	Wrong_path	-1.49	+3.67
	Correct_path	+0.81	+3.31

Table 5.8: The difference between MRC_GEN and MRC_GEN(-path) on the AbstRCT dataset when the path is predicted correctly or wrongly. Here, (Wrong/Correct)_path means that the path is predicted wrongly/correctly. Positive values mean that the path information improves the performance.

MRC_GEN(-path) when the path is predicted correctly or wrongly, respectively. Specifically, we first run the MRC_GEN model and then split the test set into parts where the predicted path is correct and incorrect. Then, we report the performance difference between the two models on these two subsets. The results are shown in Table 5.7 and Table 5.8. On the SciARG dataset, it is clear that with the path information, the performance of our model drops when the predicted path is wrong, while it increases when the predicted path is correct. This means that the performance truly improves because of correctly predicted paths. More interestingly, however, for the AbstRCT dataset, the performance increases on most of the subtasks on all three test sets regardless of the correctness of the predicted path. To investigate this issue, we manually analyse the cases where MRC_GEN(-path) predicts the wrong answer while MRC_GEN predicts the correct answer even though the path is predicted wrongly. We find two main behaviours. We call the first one *path extension*, where the predicted path is an extension of the ground truth path. It usually occurs on subgraphs with a smaller number of nodes, when an argumentative graph consists of two unconnected subgraphs. Because in most cases, the smaller subgraphs do not contain the full "chain of thought", the model may learn it from the largest subgraph to have a more comprehensive analysis of an argument. One example can be seen in Example 1 of Table 5.9. There are two

	Example 1	Example 2
Text	<AC3> Kaplan-Meier estimates showed a trend in overall survival favoring epoetin alfa (P =.13, log-rank test), <non_AC> ... < AC6> Epoetin alfa safely and effectively ameliorates anemia and significantly improves QOL in cancer patients receiving nonplatinum chemotherapy.< AC7 > Encouraging results regarding increased survival warrant another trial designed to confirm these findings.	...<AC6> Hepatic glucose production decreased after rapamycin pre-treatment (- 1.1 ± 1.1 mg/kg/min, p = 0.04) and after ITx (- 1.6 ± 0.6 mg/kg/min, p = 0.015), <non_AC>... <AC8> Rapamycin pre-treatment before ITx succeeds in reducing insulin requirement, enhancing hepatic insulin sensitivity. <AC9> This treatment may improve short-term ITx outcomes, possibly in selected patients with T1DM complicated by insulin resistance.
Rel	(<AC1> sup <AC6>), (<AC2> sup <AC6>), (<AC4> sup <AC6>), (<AC5> sup <AC6>), (<AC3> sup <AC7>)	(<AC2> sup <AC8>), (<AC3> sup <AC8>), (<AC6> sup <AC8>)
TP	'<none> <AC7>'	'<none> <AC8>' → '<support> <AC6>'
PP	'<none> <AC6>' → '<support> <AC7>'	'<none> <AC9>' → '<support> <AC6>'

Table 5.9: Two examples where MRC_GEN(-path) predicts the wrong answer while MRC_GEN gets the correct answer even though the path is predicted wrongly. Rel denotes the relations in the given argumentative text. TP denotes "true path" and PP represents "predicted path".

unconnected subgraphs, the smaller one containing AC3 and AC7. To capture the full "chain of thought", AC6 is wrongly predicted as the predecessor of AC7, while the model correctly predicts the answer. The second behaviour, which we refer to as *claim replacement*, occurs when one claim is exchanged with another one with similar high-level semantics to conclude the whole paper. As shown in Table 5.9, the model wrongly predicted the path as $< none >< AC9 >\rightarrow< support >< AC6 >$ given Example 2 due to the similarity of AC8 and AC9. Seemingly, in these two situations, a slightly wrong path is also beneficial to the model. We sampled 30 examples where the path was predicted wrongly and manually analysed them. We found that 16 were instances of path extension whereas 6 were instances of claim replacement.

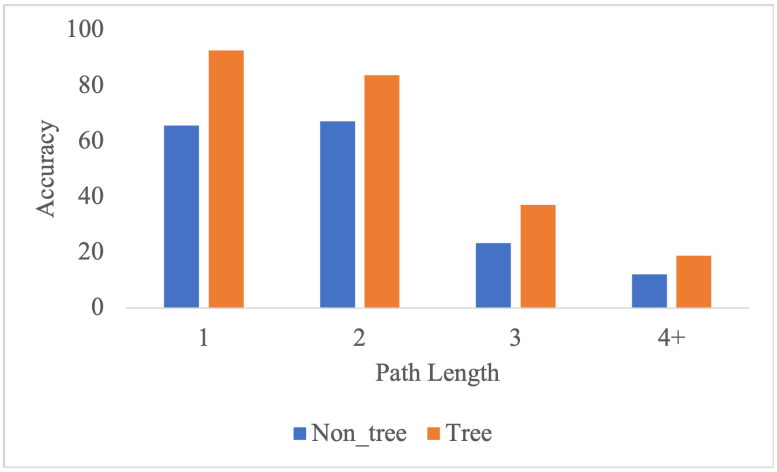


Figure 5.5: The accuracy of path prediction for tree and non tree argument structure. Here, *1*, *2*, *3* and *4+* refer to the length of the path.

5.5.2 Tree vs. Non-tree Argument Structure

As we mentioned in Section 5.3.1, the argumentative graphs in the two datasets exhibit different structures (non-tree structure for the AbstRCT dataset vs. tree structure for the SciARG dataset). In this section, we will discuss the impact of the difference in structure.

We first explore the difficulty of predicting paths in both types of structure. Specifically, as the accuracy of path prediction drops significantly with the growth of the path length, we compare the accuracy score of identical path lengths on the two datasets. The results are shown in Figure 5.5, where the score for each path is an average of all subtasks. It is clear that the accuracy on tree argument structure dataset is higher than that on the Non-tree argument structure dataset. One possible reason is that the graph structure is more random compared with the tree structure. Therefore, it is more challenging to predict the path in a graph structure.

Another conclusion is that the path information might be more useful on the graph-based dataset. From Table 5.7 and Table 5.8 we can see that even when the path information is predicted wrongly, the model could still improve the performance. See Example 1 in Table 7. The argumentative graph consists of two unconnected subgraphs. The first one includes five ACs with AC1, AC2, AC4 and AC5 supporting AC6. The second one contains only two ACs (AC3 supports AC7). When AC7 is used as a query, the true path is $< none >< AC7 >$. However, the model predicts a wrong path $< none >< AC6 > \rightarrow < support >< AC7 >$ that includes the information of AC6 and correctly predicts the relation between AC3 and AC7. It is clear that AC7 is a

more general claim which can be used in many papers while AC6 and AC3 share some important information that is specific in this paper, such as *epoetin alfa*. From this point of view, AC6 can be regarded as a piece of implicit information for the relation between AC3 and AC7. Therefore, the path extension is helpful for the non-tree structures because some subgraphs may not contain enough information. Meanwhile, for the tree structures, all the relations are connected explicitly. Thus, the model will surf from the wrongly predicted paths significantly.

5.5.3 Hyperparameter Analysis

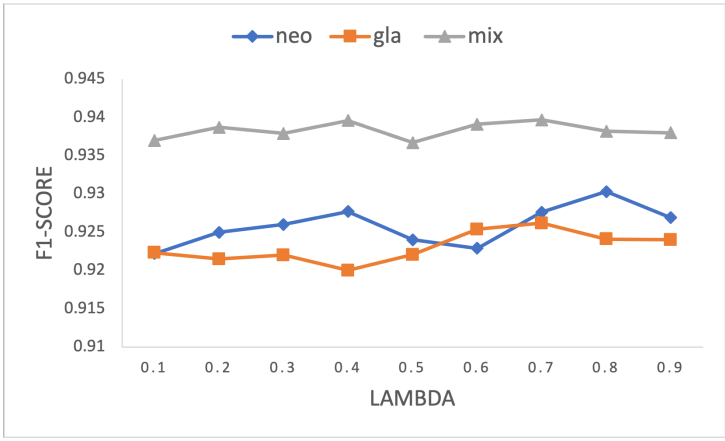


Figure 5.6: The performance of the ACC subtask on the AbstRCT dataset using different value of λ.

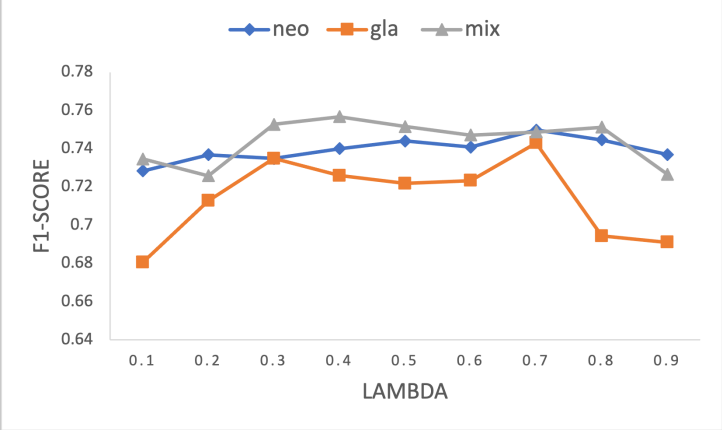


Figure 5.7: The performance of the ARIC subtask on the AbstRCT dataset using different value of λ.

In this section, we investigate the effect of the loss weights of λ, wchich is used to adjust the model's attention to AM subtasks and path information. λ takes values in [0.1, 0.2, 0.3, ...,0.9]. As mentioned in Section 5.2.4, the higher λ is, the less attention is paid to the path information. Figure 5.6 and Figure 5.7 show the impact of λ on the AbstRCT dataset.

It can be observed that the impact of the λ on model performance trends differently on ACC and ARIC subtasks. Overall, when the value of λ is relatively large, the performance on the ACC subtask tends to be slightly better. This may be because the accuracy of the ACC task is already high. It is possible that even with less focus on ACC content, the model can still learn effectively, and it is more important to pay attention to the path information. As for the ARIC subtask, it is clear that when λ is too small, the model cannot learn the answer part of the output sequence well, which causes a lower performance. However, if the λ is too large, the model mainly concentrates on the answer part and the path information is not fully captured by the model.

5.5.4 Graph as Reasoning Path

		no_path	graph_adj	graph_topo
ACC	NEO	92.29	90.14	**92.70**
	GLA	92.27	91.92	**93.02**
	MIX	93.72	91.20	**94.14**
ARIC	NEO	**72.47**	67.79	64.03
	GLA	**70.10**	64.99	62.74
	MIX	**71.20**	67.45	64.75

Table 5.10: Results of leveraging the whole graph information on the AbstRCT dataset.

We also conducted some initial experiments to leverage the whole graph information instead of the path information since the path information can only include part of the argument structure information. Since our model is a BART-Based model and the output is only a sequence but not a graph, we need to transfer the graph representation into a sequence. We propose two ways to do this. Due to the nature of argumentation, argumentative graphs are all directed acyclic graphs. Therefore, we use a graph traversal algorithm, namely topological sorting, to represent a graph as a sequence. Given an argumentative graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the algorithm returns a sequence of nodes:

$$< AC_1 >< AC_2 > ... < AC_n > \tag{5.10}$$

In order to show the hierarchical structure of the graph, we use "|" as a separator between different layers.

$$G =< AC_{l_1^1} >< AC_{l_1^2} > | < AC_{l_2^1} > ... \tag{5.11}$$

In addition, we also leverage the adjacency matrix to represent the argumentative graph inspired by Guo et al. (2022); Bao et al. (2022a). Each edge is represented by the start and end pairs of the edge, and "|" is used to distinguish different edges:

$$\begin{aligned} G =& < AC_1 >< AC_2 > |... \\ & | < AC_i >< AC_j > |... \\ & | < AC_{n-1} >< AC_n > \end{aligned} \tag{5.12}$$

From Table 5.10 we find that the models using the graph information instead of the path information work worse even compared with the there is no further information included on most cases. It seems that the topological sorting representation is helpful on the ACC subtask, however, it drops the performance on the ARIC subtask significantly.

In general, we believe that the reason why including the whole argumentative graph instead of the reasoning path harms performance is that the sequence-to-sequence model is not strong enough to learn the whole graph information. The accuracy (based on exact match) of the predicted graph is too low (5.86% on the AbstRCT dataset). We think that there are mainly two reasons. First, the number of tokens for the representation of a whole graph (from 5 tokens to 26 tokens) is much longer than that of a path (from 1 token to 4 tokens). As shown in Figure 5.3 and Figure 5.4, the accuracy of the path prediction drops significantly with the increase of the path length. When the path length is 4, the accuracy can be only 9.52%. Another reason is that the semantics of a path is in a sequential order which is consistent with the learning method of pre-trained language models, while the graph structure is more complex and is not easy to learn in a sequential order.

5.6 Related Work

A work similar to our approach is Chain of Thought Reasoning. As an essential cognitive process underlying human intelligence, the chain of thought reasoning has attracted significant attention in artificial intelligence and natural language processing. Recently, many scholars have found that chain of thought prompting can help improve the

reasoning ability of language models, especially large language models. The first work to introduce the chain of thought prompting in large language models is Wei et al. (2022). They found that manually designed exemplars with chains of thought can improve the reasoning ability of models. To enhance the certainty of the reasoning process and reduce the inconsistency between reasoning paths and answers, programming languages are used as annotated rationales to transform the problem-solving into an executable Python program (Gao et al., 2023; Chen et al., 2022a; Zhang et al., 2023a). However, such manually constructed chain of thought methods bring significant costs and have shortcomings such as difficulty in task generalization. Furthermore, some automated methods have been proposed (Zhang et al., 2023b; Wan et al., 2023). For example, Kojima et al. (2022) found that simply adding a magic phrase "let's think step by step" to the prompt enabled large language models to perform zero-shot thought chain inference without any human annotation. Further, Wang et al. (2023c) proposed Reprompting to find effective chain of thought prompts by iteratively using Gibbs sampling. However, automatic methods face the problem of the generation of low-quality reasoning paths. Different from previous work, we leverage the path-level argument-specific structural features to model an argument process that is similar to a chain of thought. Since we use golden labels to describe argument-specific structural features, we avoid the disadvantages of manual methods that are time-consuming and labour-intensive and automatic methods of poor path quality.

5.7 Summary

In this chapter, we casted the argument mining task as a multi-hop generative MRC task, which provides us with a means to leverage the path-level argument-specific structural features that are similar to the concept named "chain of thought". Specifically, given an AC as a query, our model not only predicts the type of the AC or all other ACs that are related to it based on the different subtasks, but also predicts how it reach the answers, which shows the path-level argument-specific structural features of an argument. In addition, we also introduce a two-direction method to alleviate the order bias of the output sequences of our model.

The extensive experimental results on two biomedical argument mining datasets demonstrated that our model outperforms other approaches without argument-specific structural features. We also found that the MRC_GEN model obtained comparative results compared with the GIAM model proposed in last chapter. We conclude that it is

because that both of them have their advantages and disadvantages. MRC_GEN avoids the problem of error propagation which is faced by GIAM, while GIAM can used more broad argument-specific structural features as a path is only a part of an argumentative graph. Consider the results shown in Section 4.6.2, we believe that GIAM performs better if the performance of the prediction for ROOT ACs is high. Otherwise, MRC_GEN might be a good choice.

We conducted a thorough analysis of the model by examining three aspects: i) the accuracy of the predicted paths that are used for path-level argument-specific structural features. ii) The impacts of path-level argument-specific structural features when the path is predicted correctly or wrongly iii) the impact of path-level argument-specific structural features on different type of argument structures (tree vs. graph). From our analysis, we found that as the length of the path increases, there is a notable decrease in the accuracy of the path prediction. Furthermore, we noticed that incorrect predictions of paths resulted in a decline in results on the tree argument structures. Conversely, as for the non-tree argument structures, the model performance improves across most subtasks regardless of the accuracy of predicted paths. This suggests that our model may have greater potential in non-tree argument structures, especially when the argumentative graph consists of multiple disconnected subgraphs.

Chapter 6

Conclusion

This thesis studied the task of argument mining, which involves the extraction of argument structure from a given argumentative text. The main content was structured across six chapters, comprising an introductory chapter, a background chapter, three method chapters, and a conclusion chapter summarising the findings of this research.

In Chapter 2, we firstly provide the technical foundation essential to our methodology. We started with the fundamental Feed-forward Neural Network, and then we progressed with several neural network components, including recurrent neural networks, graph neural networks, attention mechanisms and Transformers. These serve as the core tools of our methodology. Then, we discussed pre-trained language models such as BERT and BART. Secondly, we provide an in-depth introduction to the argument mining task and an extensive overview of various methodologies. After detailing the task and methods, we introduce the biomedical datasets used in argument mining. Finally, we end this chapter with the limitations of previous models, which give insights into this research.

6.1 Validation of Research Hypotheses

The main purpose of this chapter is to validate the research hypotheses postulated in this thesis. In this chapter we explore the role of structural features on AM tasks. The primary objective of this thesis was to investigate the impact and utility of structural features on AM tasks, and to accomplish this, various approaches were designed to model different structural features. We will examine each of the methods developed for AM in the following parts of this chapter.

In Chapter 3, we addressed our initial research question and hypotheses regarding

the connection between the genre-specific structural features of biomedical literature and subtasks of argument components. Specifically, we explored:

RQ_1 Does the identification of genre-specific structural features of biomedical literature facilitate the mining of argument components in biomedical literature abstracts?

H_1 The genre-specific structural features of biomedical literature are helpful for mining argument components in biomedical literature abstracts as it can be used to locate the argumentative parts.

In order to examine our first hypothesis, we started from modelling the genre-specific structural features of arguments in biomedical literature. As a result, we selected text zoning. It is a task that aims at segmenting a text into zones, where each zone, differing from others, consists of text parts with specific functions. Among several zoning schemes for biomedical abstracts, we chose the one (*Background*, *Objective*, *Method*, *Result* and *Conclusion*) designed for the PubMedRCT (Dernoncourt and Lee, 2017) dataset because it is the biggest dataset and an off-the-shelf tool named HSLN (Jin and Szolovits, 2018) achieves very high performance on this dataset. We predicted the zoning labels for each biomedical abstract from the AM dataset named AbstRCT (Mayer et al., 2020) and analysed the distribution of argument components and zoning information within the training subset of this dataset. We found that most of the argument components exist in the *Result* and *Conclusion* zones, with a very few ACs in the *Background* zone. As for the type of ACs, we found that the premises are highly related to the *Result* zones and there is a strong correlation between the claims and the *Conclusion* zones. It confirms our hypothesis **??** that there should be a strong correlation between the genre-specific structural features of biomedical literature and argument components in biomedical literature abstracts.

To address our second hypothesis, we introduced two transformer-based models that integrate zoning information as genre-specific structural features into argument component identification and classification sub-tasks. The first model addresses sentence-level argument mining, while the second focuses on the token-level task. Notably, we enhanced each sentence through adding the zoning labels predicted by a pre-existing model to the beginning of it, drawing inspiration from the common practice in biomedical abstracts. Additionally, we leveraged multi-head attention to propagate sentence-level zoning information to individual tokens within a sentence.

In the experiment part, we first designed a heuristic method that predicts the boundary and type of each AC only based on the zoning labels and we found that the heuristic

method exhibited competitive performance even without any semantic information. Further, evaluation on our proposed token-level and sentence-level approaches on two AM datasets revealed that even though the predicted zoning labels are helpful for argument component identification and classification sub-tasks. By conducting analysis, we observed that even when only one type of zoning label is used, the model still obtained improvement from zoning information, especially when the selected label is *Result* or *Conclusion*. This is because such two types of labels are highly related to premises and claims. In addition, we also found that using the *Method* label alone achieves quite a good performance especially in the glaucoma test set. We argue that it is due to the high proportion of *Method* labels (30%) in the dataset and the almost perfect performance on such label.

Based on the findings shown above, we conclude the following:

- Text zoning can be used to describe the genre-specific structural features of biomedical abstracts.
- Zoning labels are highly related to the ACs in biomedical abstracts.
- The model can gain improvement through the zoning labels even only silver labels are provided.

In Chapter 4, we showed the positive impact of the graph-level argument-specific structural features, which addressed our second research question and hypothesis:

RQ_2 Can the predicted argument (sub-)graph be used as argument-specific structural features to improve the performance?

- $H_{2.1}$ AM can be modeled as a graph generation process by transferring it into a task such as muti-turn machine reading comprehension.
- $H_{2.2}$ The graph generation process allows the utilisation of graph-level argument-specific structural features through graph neural network.

First we introduced how to represent argument-specific structural features as (sub)graphs. Specifically, such features were represented using initial (sub)graphs, which only has argument relation connection information but no AR and AC types. To exploit this graph-level argument-specific structural features, we proposed a two-stage approach. First, the model generated an initial (sub)graph by modeling the ARI task as a multi-turn machine reading comprehension task, and in this process, the subgraph generated in the previous turn was used as argument-specific structural features. Considering that there

will be connections between multiple ACs associated with the same AC, we modeled the ARI task as a multi-label classification task in each QA turn to strengthen the model's learning of this connection. When the complete initial graph is predicted, the model used it as graph-level argument-specific structural features to predict the AC type labels and AR type labels.

Evaluation on two biomedical datasets illustrated that our model can achieve SOTA results. Furthermore, through ablation study, we demonstrated that this kind of graph-level argument-specific structural features helps to improve the model performance and confirmed our second hypothesis. Since the graph-level argument-specific structural features utilised in our model were constructed based on the initial graph that was predicted by the model, we also analysed the impact of error propagation on model performance. We found that the effect of the model suffer from error propagation severely. Specifically, when the model predicts `ROOT` AC incorrectly, the model's performance drops by nearly twenty points compared to the case when `ROOT` AC is correctly predicted. We proposed *query extension* to alleviate this problem, which improves the model's performance by more than ten points when `ROOT` AC is incorrectly predicted, and also improves the model's performance to a certain extent when `ROOT` AC is correctly predicted. We also compared different methods of building ROOT query, and found that although the natural language query contains richer information, the improvement compared to the case where special token is used as the ROOT query is minimal.

We can thus conclude that:

- AM can be modeled as a graph generation process by transferring it into a muti-turn machine reading comprehension task.
- The graph-level argument-specific structural features are helpful for the argument mining task.
- When the ROOT query is correctly predicted, the model is better able to take advantage of graph-level argument-specific structural features.

In chapter 5, the final research question and hypothesis are addressed in this chapter:

RQ_3 Can each argument be viewed as a chain of reasoning so that the reason path can be used as argument-specific structural features?

$H_{3.1}$ The argumentative process can be viewed as a chain of reasoning path.

$H_{3.2}$ The AM task can be transfered as a multi-hop MRC task to explicitly learn this reasoning path.

$H_{3.3}$ This reasoning path can improve the performance of AM models.

Evaluation on two biomedical datasets shows that our proposed model is able to outperform models that do not exploit argument-specific structural features. Through an ablation study, we found that although path-level argument-specific structural features only contain partial information compared with argumentative graph, they are still significantly helpful in improving the model performance, which confirms our third hypothesis.

However, further analysis shows that correctly predicting this path is not easy for our model. Although the model can perform relatively well when the path length is short, especially when the argument structure resembles a tree, the accuracy of the model's path prediction declines significantly as the path length increases. Moreover, on the SciARG dataset, we observed that accurate path prediction enhances the model's performance, whereas incorrect predictions lead to a decline in results. This illustrates that the model relies heavily on path-level argument-specific structural features when solving AM tasks.

In contract, it is interesting to note that for the AbstRCT dataset, performance improves on most subtasks on all three test sets, regardless of the correctness of the predicted paths. In order to gain a deeper understanding of this phenomenon, we conducted a case study. We found that there are two main situations. The first one is *path expansion*, where the predicted path is an expansion of the ground-truth path. This usually happens on subgraphs with fewer nodes, when an argumentative graph consists of two unconnected subgraphs. Because in most cases the smaller subg-raphs do not contain a complete "thought chain", the model may learn from the largest subgraph to gain a more comprehensive analysis of an argument.' The second is *claim replacement*, which occurs when one claim is replaced with another claim that has similar high-level semantics to arrive at the conclusion of the entire paper. This also proves that this method are more helpful in predicting the argument structure of graph structure than the argument structure of tree structure.

The following conclusions can be obtained based on the above observations:

- The argument-specific structural features can be viewed as a chain of reasoning.
- The path-level argument-specific structural features are helpful for the argument mining task.
- The path-level structural features are more helpful in predicting the argument structure of graph structure (especially when it contains several subgraphs)than the argument structure of tree structure.

Overall, the work presented in this thesis demonstrates the importance of structural features for argument mining in biomedical abstracts. First, we proved that the genre-specific structural features using zoning labels as a representation method is very helpful for ACI and ACC subtasks. These features can effectively solve AM tasks even without using semantic information. Secondly, we also found that both graph-level and path-level argument-specific structural features help enhance the model's processing of argument-specific structural features. Further, we also showed that when there is a demonstration relationship between one AC and multiple ACs, it is helpful to consider the correlation between multiple ACs to improve the effect of the model.

6.2 Limitations and Future Work

In this section, we discuss the shortcomings of the models proposed in this thesis. We also analyse possible approaches used to address these shortcomings and our future work.

6.2.1 Mitigation of Error Propagation

In our analysis of Chapter 4, our proposed GIAM model suffers from error propagation issues. During the training process, we are based on the ground-truth graph-level argument structure features, but during the inference process, we can only use the graph-level argument structure features predicted by the model. Therefore, an obvious research direction is how to alleviate this problem. One possible research direction is to use reinforcement learning (Kaelbling et al., 1996) to solve this problem. In reinforcement learning, an agent learns through interaction with the environment and continuously adjusts its behavior to maximize the expected cumulative reward. Unlike

traditional supervised learning, reward signals in reinforcement learning are often sparse, delayed, and potentially noisy. Therefore, agents need to learn to make appropriate decisions under such uncertainty and noise, which may help mitigate the propagation of errors. In fact, reinforcement learning has been widely explored to solve the error propagation problem (Lê and Fokkens, 2017). Another possible approach is to utilise scheduled sampling methods (Mihaylova and Martins, 2019; Bengio et al., 2015). Scheduled sampling is a technique used to train sequence generation models, often used in sequence-to-sequence models. During training, scheduled sampling gradually reduces the probability that the model receives true previous outputs, and gradually increases the probability that the model receives its own generated outputs. Doing so gradually frees the model from the constraints of the teacher and better adapts to the real generative environment, thereby reducing errors in the inference phase.

6.2.2 Extending Argument-specific Structural Features to Token-level AM

Among the three models proposed in this thesis, only the zoning information based model proposed in Chapter 3 can be applied at the token-level. This is because both GIAM proposed in Chapter 4 and MRC_GEN proposed in Chapter 5 assume that the ACI task has been solved perfectly, and then they work at the AC level. We use such a setup based on three reasons. First, as we mentioned before, previous models are also evaluated based on such settings (Si et al., 2022), and adopting the same settings facilitates comparison with previous state-of-the-art models. Secondly, the reasonability of this setting also depends on the choice of annotation scheme. On the SciARG dataset, AC annotation is performed at the sentence level, and each sentence is assigned an AC category. Third, this setting does not affect our investigation of **RQ2** and **RQ3**, which focuses on the impact of argument-specific structural features on the AM task. However, this setting may not be suitable for all types of AM tasks, like real-world situations where argument texts may contain non-argument parts and the golden ACs cannot be directly provided. There are two different situations. First, for sentence-level AM tasks that include non-argumentation spans, our model can be easily extended by adding a non-argumentation category in the ACC task to represent sentences as non-argumentation. For AR-related subtasks, the answer is none when a non-argumentative sentence is used as a query. However, the token-level AM task poses a challenge to our model regardless of whether the argument text contains non-argumentative text.

Therefore, a future research direction is how to utilise argument-specific structural features in the token-level AM task.

6.2.3 Extension to Other Domains and Tasks

The structural features used in this thesis are based on biomedical abstracts. Therefore, a possible limitation is whether this model is effective on the AM task in other domains. We argue that for argumentative texts in the same specific genre, such as student essays and legal texts that are written following some implicit rules, the argument-specific structural features of such texts are usually similar. For example, in a student essay, there is usually a major claim supported by several claims, and then a number of premises which are related to the claims are included (Eger et al., 2017). Therefore, our model can also be useful for such genres. However, structural features may be not useful in user-generated arguments drawn from other domains such as social media, i.e., online forums, where the arguments can be more random compared with scientific abstracts.

Next, we discuss how our models can help other non-AM tasks. In this thesis, we propose different methods of utilising structural features. Using such features is not only helpful for AM tasks, but may also be useful for other document-level NLP tasks, such as discourse parsing and event extraction. Therefore, a possible research direction is to test the effect of our model on a wider range of tasks. In addition, our method can also help downstream tasks, such as the misinformation detection task (Sarrouti et al., 2021; Mohr et al., 2022). Especially during the COVID-19 epidemic, a lot of misinformation spread rapidly on the Internet. In order to alleviate this problem, many researchers try to use evidence extracted from biomedical literature to automatically determine whether the information is real or fake (Vladika et al., 2023; Wang et al., 2023b). However, these methods are mainly based on golden evidence annotated by experts. Therefore, our future work includes exploring whether the evidence extracted by our proposed model could be helpful for the misinformation detection task.

Bibliography

Pablo Accuosto. 2021. *Mining arguments in scientific abstracts: Application to argumentative quality assessment*. Ph.D. thesis, Pompeu Fabra University, Spain.

Pablo Accuosto, Mariana Neves, and Horacio Saggion. 2021. Argumentation mining in scientific literature: from computational linguistics to biomedicine. In *Frommholz I, Mayr P, Cabanac G, Verberne S, editors. BIR 2021: 11th International Workshop on Bibliometric-enhanced Information Retrieval; 2021 Apr 1; Lucca, Italy. Aachen: CEUR; 2021. p. 20-36*. CEUR Workshop Proceedings.

Pablo Accuosto and Horacio Saggion. 2019a. Discourse-driven argument mining in scientific abstracts. In *International Conference on Applications of Natural Language to Information Systems*, pages 182–194. Springer.

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

Pablo Accuosto and Horacio Saggion. 2020. Mining arguments in scientific abstracts with discourse-level embeddings. *Data & Knowledge Engineering*, 129:101840.

Titipat Achakulvisut, Chandra Bhagavatula, Daniel Acuna, and Konrad Kording. 2019. Claim extraction in biomedical publications using deep discourse model and transfer learning. *arXiv preprint arXiv:1907.00962*.

Irina Achmadeeva, Irina Kononenko, Natalia Salomatina, and Elena Sidorova. 2019. Indicator patterns as features for argument mining. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 0886–0891. IEEE.

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68, Baltimore, Maryland. Association for Computational Linguistics.

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

Ahmet Aker, Alfred Sliwa, Yuan Ma, Ruishen Lui, Niravkumar Borad, Seyedeh Ziyaei, and Mina Ghobadi. 2017. What works and what does not: Classifier and feature analysis for argument mining. In *Proceedings of the 4th Workshop on Argument Mining*, pages 91–96, Copenhagen, Denmark. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, and Benno Stein. 2017. Patterns of argumentation strategies across topics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1351–1357, Copenhagen, Denmark. Association for Computational Linguistics.

Abdulaziz Alamri and Mark Stevenson. 2016. A corpus of potentially contradictory research claims from cardiovascular research abstracts. *Journal of biomedical semantics*, 7(1):1–9.

Alaa Alhamzeh, Mohamed Bouhaouel, Előd Egyed-Zsigmond, Jelena Mitrović, Lionel Brunie, and Harald Kosch. 2021. A stacking approach for cross-domain argument identification. In *Database and Expert Systems Applications: 32nd International Conference, DEXA 2021, Virtual Event, September 27–30, 2021, Proceedings, Part I 32*, pages 361–373. Springer.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Patricia G Avery and Michael F Graves. 1997. Scaffolding young learners' reading of social studies texts. *Social studies and the young learner*, 9(4):10–14.

Moshe Azar. 1999. Argumentative text as rhetorical structure: An application of rhetorical structure theory. *Argumentation*, 13:97–114.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA. OpenReview.net.

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

Jianzhu Bao, Yuhang He, Yang Sun, Bin Liang, Jiachen Du, Bing Qin, Min Yang, and Ruifeng Xu. 2022a. A generative model for end-to-end argument mining with reconstructed positional encoding and constrained pointer mechanism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10437–10449, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianzhu Bao, Jingyi Sun, Qinglin Zhu, and Ruifeng Xu. 2022b. Have my arguments been replied to? argument pair extraction as machine reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 29–35, Dublin, Ireland. Association for Computational Linguistics.

Xiaoyi Bao, Xiaotong Jiang, Zhongqing Wang, Yue Zhang, and Guodong Zhou. 2023. Opinion tree parsing for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7971–7984, Toronto, Canada. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings*

of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15), pages 1171–1179, Cambridge, MA, USA, 1171–1179. MIT Press.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Catherine Blake. 2010. Beyond genes, proteins, and abstracts: Identifying scientific claims from full-text biomedical articles. *Journal of biomedical informatics*, 43(2):173–189.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. AMPERSAND: Argument mining for PERSuAsive oNline discussions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022a. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Zaiqian Chen, Daniel Verdi do Amarante, Jenna Donaldson, Yohan Jo, and Joonsuk Park. 2022b. Argument mining for review helpfulness prediction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8922, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liying Cheng, Lidong Bing, Ruidan He, Qian Yu, Yan Zhang, and Luo Si. 2022. IAM: A comprehensive and large-scale dataset for integrated argument mining tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2277–2287, Dublin, Ireland. Association for Computational Linguistics.

Liying Cheng, Tianyu Wu, Lidong Bing, and Luo Si. 2021. Argument pair extraction via attention-guided multi-layer multi-cross encoding. 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 6341–6353, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Carmen Dayrell, Arnaldo Candido Jr., Gabriel Lima, Danilo Machado Jr., Ann Copestake, Valéria Feltrim, Stella Tagnin, and Sandra Aluisio. 2012. Rhetorical move detection in English abstracts: Multi-label sentence classifiers and their annotated corpora. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1604–1609, Istanbul, Turkey. European Language Resources Association (ELRA).

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023. Bidirectional generative framework for cross-domain aspect-based sentiment analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12272–12285, Toronto, Canada. Association for Computational Linguistics.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

Syed Burhan ud Din Tahir. 2017. Comparative analysis of supervised learning approaches for argument identification. In *2017 International Multi-topic Conference (INMIC)*, pages 1–5, Lahore, Pakistan. IEEE.

Yang Du, Minglan Li, and Mengxue Li. 2017. Joint extraction of argument components and relations. In *2017 International Conference on Asian Language Processing (IALP)*, pages 1–4, Singapore. IEEE.

Subhabrata Dutta, Dipankar Das, and Tanmoy Chakraborty. 2020. Changing views: Persuasion modeling and argument extraction from online discussions. *Information Processing & Management*, 57(2):102085.

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

Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.

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

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.

Aris Fergadis, Dimitris Pappas, Antonia Karamolegkou, and Haris Papageorgiou. 2021. Argumentation mining in scientific literature for sustainable development. In *Proceedings of the 8th Workshop on Argument Mining*, pages 100–111, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Michael Fromm, Evgeniy Faerman, Max Berrendorf, Siddharth Bhargava, Ruoxia Qi, Yao Zhang, Lukas Dennert, Sophia Selle, Yang Mao, and Thomas Seidl. 2021. Argument mining driven analysis of peer-reviews. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4758–4766, Virtual Event. AAAI Press.

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

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2018. Argumentative link prediction using residual networks and multi-objective learning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2021a. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2021b. Investigating logic tensor networks for neural-symbolic argument mining. In *Proc. 1st Int. Joint Conf. Learn., Reasoning*, pages 1–7, Online.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2023. Multi-task attentive residual networks for argument mining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1877–1892.

Esther Galbrun and Pauli Miettinen. 2012. From black and white to full color: extending redescription mining outside the boolean world. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):284–303.

Boris Galitsky, Dmitry Ilvovsky, and Sergey O Kuznetsov. 2018. Detecting logical argumentation in text via communicative discourse tree. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(5):637–663.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799, Honolulu, Hawaii, USA. PMLR.

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

Ann-Sophie Gnehm. 2018. Text zoning for job advertisements with bidirectional lstms. In *Proceedings of the 3rd Swiss Text Analytics Conference*, pages 66–74, Winterthur, Switzerland. CEUR-WS.org.

Ann-Sophie Gnehm and Simon Clematide. 2020. Text zoning and classification for job advertisements in German, French and English. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 83–93, Online. Association for Computational Linguistics.

Nancy Green. 2015a. Identifying argumentation schemes in genetics research articles. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 12–21, Denver, CO. Association for Computational Linguistics.

Nancy L Green. 2010. Representation of argumentation in text with rhetorical structure theory. *Argumentation*, 24:181–196.

Nancy L Green. 2015b. Annotating evidence-based argumentation in biomedical text. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 922–929, Washington, DC, USA. IEEE.

Qipeng Guo, Yuqing Yang, Hang Yan, Xipeng Qiu, and Zheng Zhang. 2022. DORE: Document ordered relation extraction based on generative framework. In *Findings*

of the Association for Computational Linguistics: EMNLP 2022, pages 3463–3474, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ivan Habernal, Daniel Faber, Nicola Recchia, Sebastian Bretthauer, Iryna Gurevych, Indra Spiecker genannt Döhmann, and Christoph Burchard. 2023. Mining legal arguments in court decisions. *Artificial Intelligence and Law*, pages 1–38.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, Las Vegas, NV, USA.

Freya Hewett, Roshan Prakash Rane, Nina Harlacher, and Manfred Stede. 2019. The utility of discourse parsing features for predicting argumentation structure. In *Proceedings of the 6th Workshop on Argument Mining*, pages 98–103, Florence, Italy. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sally Hopewell, Mike Clarke, David Moher, Elizabeth Wager, Philippa Middleton, Douglas G Altman, Kenneth F Schulz, and Consort Group. 2008. Consort for reporting randomized controlled trials in journal and conference abstracts: explanation and elaboration. *PLoS medicine*, 5(1):e20.

Laurine Huber, Justine Reynaud, Mathilde Dargnat, and Yannick Toussaint. 2020. Aoc-poset on discourse and argumentation subgraphs: what can we learn on their dependencies? In *Concept Lattice and their Applications*, Tallinn, Estonia. CEUR-WS.org.

Laurine Huber, Yannick Toussaint, Charlotte Roze, Mathilde Dargnat, and Chloé Braud. 2019. Aligning discourse and argumentation structures using subtrees and redescription mining. In *Proceedings of the 6th Workshop on Argument Mining*, pages 35–40, Florence, Italy. Association for Computational Linguistics.

Hirokazu Igari, Akira Shimazu, and Koichiro Ochimizu. 2012. Document structure analysis with syntactic model and parsers: Application to legal judgments. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2011 Workshops, LENLS, JURISIN,*

ALSIP, MiMI, Takamatsu, Japan, December 1-2, 2011. Revised Selected Papers 3, pages 126–140. Springer.

Khudhair Jasim, Ahmed T Sadiq, and Hasanen S Abdullah. 2019. A framework for detection and identification the components of arguments in arabic legal texts. In *2019 First International Conference of Computer and Applied Sciences (CAS)*, pages 67–72, Baghdad, Iraq. IEEE.

Weidong Jia, Zhiqiang Zhu, Tengyue Zhang, Gaofei Fan, Pingsheng Fan, Yabei Liu, and Qiaohong Duan. 2013. Treatment of malignant ascites with a combination of chemotherapy drugs and intraperitoneal perfusion of verapamil. *Cancer chemotherapy and pharmacology*, 71:1585–1590.

Yichen Jiang, Nitish Joshi, Yen-Chun Chen, and Mohit Bansal. 2019. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2714–2725, Florence, Italy. Association for Computational Linguistics.

Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3100–3109, Brussels, Belgium. Association for Computational Linguistics.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Zixuan Ke, Winston Carlile, Nishant Gurrapadi, and Vincent Ng. 2018. Learning to give feedback: Modeling attributes affecting argument persuasiveness in student essays. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4130–4136, Stockholm, Sweden. ijcai.org.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. *BMC bioinformatics*, 12(2):1–10.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA. OpenReview.net.

Thomas N. Kipf and Max Welling. 2017. Semisupervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France. OpenReview.net.

Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2015. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11, Denver, CO. Association for Computational Linguistics.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, New Orleans, LA, USA. Curran Associates, Inc.

Irina Kononenko, Elena Sidorova, and Irina Akhmadeeva. 2020. The study of argumentative relations in popular science discourse. In *Russian Conference on Artificial Intelligence*, pages 309–324, Moscow, Russia. Springer.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4691–4698, Florence, Italy. Association for Computational Linguistics.

Nicole Landi, Stephen J Frost, W Einar Mencl, Rebecca Sandak, and Kenneth R Pugh. 2013. Neurobiological bases of reading comprehension: Insights from neuroimaging studies of word-level and text-level processing in skilled and impaired readers. *Reading & Writing Quarterly*, 29(2):145–167.

Anne Lauscher, Goran Glavaš, and Kai Eckert. 2018. ArguminSci: A tool for analyzing argumentation and rhetorical aspects in scientific writing. In *Proceedings of the 5th Workshop on Argument Mining*, pages 22–28, Brussels, Belgium. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Minh Lê and Antske Fokkens. 2017. Tackling error propagation through reinforcement learning: A case of greedy dependency parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 677–687, Valencia, Spain. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Minglan Li, Yang Gao, Hui Wen, Yang Du, Haijing Liu, and Hao Wang. 2017. Joint rnn model for argument component boundary detection. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 57–62, Banff, AB, Canada. IEEE.

Weichen Li, Patrick Abels, Zahra Ahmadi, Sophie Burkhardt, Benjamin Schiller, Iryna Gurevych, and Stefan Kramer. 2021. Topic-guided knowledge graph construction for argument mining. In *2021 IEEE International Conference on Big Knowledge (ICBK)*, pages 315–322, Auckland, New Zealand. IEEE.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2010. Corpora for the conceptualisation and zoning of scientific papers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Maria Liakata, Paul Thompson, Anita de Waard, Raheel Nawaz, Henk Pander Maat, and Sophia Ananiadou. 2012. A three-way perspective on scientific discourse annotation for knowledge extraction. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 37–46, Jeju Island, Korea. Association for Computational Linguistics.

Marco Lippi and Paolo Torroni. 2015. Argument mining: A machine learning perspective. In *Theory and Applications of Formal Argumentation: Third International Workshop, TAFA 2015, Buenos Aires, Argentina, July 25-26, 2015, Revised Selected Papers 3*, pages 163–176. Springer.

Boyang Liu, Viktor Schlegel, Riza Batista-Navarro, and Sophia Ananiadou. 2022. Incorporating zoning information into argument mining from biomedical literature. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6162–6169, Marseille, France. European Language Resources Association.

Boyang Liu, Viktor Schlegel, Riza Batista-Navarro, and Sophia Ananiadou. 2023a. Argument mining as a multi-hop generative machine reading comprehension task. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10846–10858, Singapore. Association for Computational Linguistics.

Boyang Liu, Viktor Schlegel, Paul Thompson, Riza Theresa Batista-Navarro, and Sophia Ananiadou. 2023b. Global information-aware argument mining based on a top-down multi-turn qa model. *Information Processing & Management*, 60(5):103445.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, Lisbon, Portugal. Association for Computational Linguistics.

Qian Liu, Rui Mao, Xiubo Geng, and Erik Cambria. 2023c. Semantic matching in machine reading comprehension: An empirical study. *Information Processing & Management*, 60(2):103145.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA. OpenReview.net.

Anastasios Lytos, Thomas Lagkas, Panagiotis Sarigiannidis, and Kalina Bontcheva. 2019. The evolution of argumentation mining: From models to social media and emerging tools. *Information Processing & Management*, 56(6):102055.

Rana Malhas and Tamer Elsayed. 2022. Arabic machine reading comprehension on the holy qur'an using cl-arabert. *Information Processing & Management*, 59(6):103068.

Tobias Mayer, Elena Cabrio, Marco Lippi, Paolo Torroni, and Serena Villata. 2018. Argument mining on clinical trials. In *Computational Models of Argument - Proceedings of COMMA 2018*, pages 137–148, Warsaw, Poland. IOS Press.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, pages 2108–2115, Santiago de Compostela, Spain. IOS Press.

Tobias Mayer, Santiago Marro, Elena Cabrio, and Serena Villata. 2021. Enhancing evidence-based medicine with natural language argumentative analysis of clinical trials. *Artificial Intelligence in Medicine*, 118:102098.

Mikru Lake Melie, Debela Tesfaye, Alemu Kumilachew Tegegnie, and Derejaw Lake Melie. 2023. Argument mining from amharic argumentative texts using machine learning approach. *African Journal of Science, Technology, Innovation and Development*, 15(7):895–901.

Raphaël Micheli. 2012. Arguing without trying to persuade? elements for a non-persuasive definition of argumentation. *Argumentation*, 26(1):115–126.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19:1–22.

Isabelle Mohr, Amelie Wührl, and Roman Klinger. 2022. Covert: A corpus of fact-checked biomedical COVID-19 tweets. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference,LREC 2022*, pages 244–257, Marseille, France. European Language Resources Association.

Gaku Morio and Katsuhide Fujita. 2019. Syntactic graph convolution in multi-task learning for identifying and classifying the argument component. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 271–278, Newport Beach, CA, USA. IEEE.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai. 2022. End-to-end argument mining with cross-corpora multi-task learning. *Transactions of the Association for Computational Linguistics*, 10:639–658.

Umer Mushtaq and Jérémie Cabessa. 2022. Argument classification with bert plus contextual, structural and syntactic features as text. In *Neural Information Processing - 29th International Conference, ICONIP*, pages 622–633, Virtual Event. Springer.

Heinz. Neudecker, Shuangzhe. Liu, and Wolfgang. Polasek. 1995. The hadamard product and some of its applications in statistics. *Statistics*, 26(4):365–373.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.

E Michael Nussbaum. 2002. Scaffolding argumentation in the social studies classroom. *The Social Studies*, 93(2):79–83.

Juri Opitz and Anette Frank. 2019. Dissecting content and context in argumentative relation analysis. In *Proceedings of the 6th Workshop on Argument Mining*, pages 25–34, Florence, Italy. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035, Vancouver, BC, Canada. Curran Associates, Inc.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative relation classification with background knowledge. In *Computational Models of Argument - Proceedings of COMMA*, pages 319–330. IOS Press, Perugia, Italy.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Andreas Peldszus and Manfred Stede. 2015. Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948, Lisbon, Portugal. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2016. Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 103–112, Berlin, Germany. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2016. 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, San Diego, California. Association for Computational Linguistics.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1364–1373, Copenhagen, Denmark. Association for Computational Linguistics.

Jan Wira Gotama Putra, Simone Teufel, and Takenobu Tokunaga. 2021. Multi-task and multi-corpora training strategies to enhance argumentative sentence linking

performance. In *Proceedings of the 8th Workshop on Argument Mining*, pages 12–23, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.

Tim Repke and Ralf Krestel. 2018. Bringing back structure to free text email conversations with recurrent neural networks. In *European Conference on Information Retrieval*, pages 114–126, Grenoble, France. Springer.

João António Rodrigues and António Branco. 2022. Transferring confluent knowledge to argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6859–6874, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Federico Ruggeri, Marco Lippi, and Paolo Torroni. 2021. Tree-constrained graph neural networks for argument mining. *arXiv preprint arXiv:2110.00124*.

Ameer Saadat-Yazdi, Jeff Z. Pan, and Nadin Kokciyan. 2023. Uncovering implicit inferences for improved relational argument mining. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2484–2495, Dubrovnik, Croatia. Association for Computational Linguistics.

David L. Sackett. 1997. Evidence-based medicine. *Seminars in Perinatology*, 21(1):3–5.

Patrick Saint-Dizier. 2017. Knowledge-driven argument mining based on the qualia structure. *Argument & Computation*, 8(2):193–210.

Arif Saricoban. 2002. Reading strategies of successful readers through the three phase approach. *The Reading Matrix*, 2(3).

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Matthew Shardlow, Riza Batista-Navarro, Paul Thompson, Raheel Nawaz, John McNaught, and Sophia Ananiadou. 2018. Identification of research hypotheses and new knowledge from scientific literature. *BMC medical informatics and decision making*, 18(1):1–13.

Jiasheng Si, Liu Sun, Deyu Zhou, Jie Ren, and Lin Li. 2022. Biomedical argument mining based on sequential multi-task learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(2):864–874.

Luciana B Sollaci and Mauricio G Pereira. 2004. The introduction, methods, results, and discussion (imrad) structure: a fifty-year survey. *Journal of the medical library association*, 92(3):364.

Wei Song, Ziyao Song, Ruiji Fu, Lizhen Liu, Miaomiao Cheng, and Ting Liu. 2020. Discourse self-attention for discourse element identification in argumentative student essays. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2820–2830, Online. Association for Computational Linguistics.

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

Christian Stab, Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *ArgNLP*, pages 21–25, Forlì-Cesena, Italy. CEUR-WS.org.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Manfred Stede, Stergos Afantenos, Andreas Peldszus, Nicholas Asher, and Jérémy Perret. 2016. Parallel discourse annotations on a corpus of short texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1051–1058, Portorož, Slovenia. European Language Resources Association (ELRA).

Nikolaos Stylianou and Ioannis Vlahavas. 2021. Transformed: End-to-end transformers for evidence-based medicine and argument mining in medical literature. *Journal of Biomedical Informatics*, 117:103767.

Yang Sun, Bin Liang, Jianzhu Bao, Yice Zhang, Geng Tu, Min Yang, and Ruifeng Xu. 2023. Probing graph decomposition for argument pair extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13075–13088, Toronto, Canada. Association for Computational Linguistics.

Simone Teufel et al. 1999. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

Paul Thompson, Raheel Nawaz, John McNaught, and Sophia Ananiadou. 2011. Enriching a biomedical event corpus with meta-knowledge annotation. *BMC bioinformatics*, 12(1):1–18.

Elif Toprak and Gamze Almacioğlu. 2009. Three reading phases and their applications in the teaching of english as a foreign language in reading classes with young learners. *Journal of language and Linguistic Studies*, 5(1):20–36.

Stephen E Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Dietrich Trautmann. 2020. Aspect-based argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 41–52, Online. Association for Computational Linguistics.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop.*, New Orleans, LA, USA. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on*

Neural Information Processing Systems 2017, pages 5998–6008, Long Beach, CA, USA. Curran Associates, Inc.

Maria Paz Garcia Villalba and Patrick Saint-Dizier. 2012. Some facets of argument mining for opinion analysis. volume 245, pages 23–34, Vienna, Austria. IOS Press.

Juraj Vladika, Phillip Schneider, and Florian Matthes. 2023. Healthfc: A dataset of health claims for evidence-based medical fact-checking. *arXiv preprint arXiv:2309.08503*.

Vern R. Walker, Dina Foerster, Julia Monica Ponce, and Matthew Rosen. 2018. Evidence types, credibility factors, and patterns or soft rules for weighing conflicting evidence: Argument mining in the context of legal rules governing evidence assessment. In *Proceedings of the 5th Workshop on Argument Mining*, pages 68–78, Brussels, Belgium. Association for Computational Linguistics.

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

Qian Wan, Scott Crossley, and Yu Tian. 2022. Automated classification of argumentative components in students' essays. In *International Conference on Intelligent Tutoring Systems*, pages 171–182, Bucharest, Romania. Springer.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Gengyu Wang, Kate Harwood, Lawrence Chillrud, Amith Ananthram, Melanie Subbiah, and Kathleen McKeown. 2023b. Check-COVID: Fact-checking COVID-19 news claims with scientific evidence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14114–14127, Toronto, Canada. Association for Computational Linguistics.

Hao Wang, Zhen Huang, Yong Dou, and Yu Hong. 2020. Argumentation mining on essays at multi scales. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5480–5493, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Bonnie Webber, Markus Egg, and Valia Kordoni. 2012. Discourse structure and language technology. *Natural Language Engineering*, 18(4):437–490.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, New Orleans, LA, USA,. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Amelie Wührl and Roman Klinger. 2022. Entity-based claim representation improves fact-checking of medical content in tweets. In *Proceedings of the 9th Workshop on Argument Mining*, pages 187–198, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Qingrong Xia, Bo Zhang, Rui Wang, Zhenghua Li, Yue Zhang, Fei Huang, Luo Si, and Min Zhang. 2021. A unified span-based approach for opinion mining with syntactic constituents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1795–1804, Online. Association for Computational Linguistics.

Genki Yagawa and Atsuya Oishi. 2021. *Feedforward Neural Networks*, pages 11–23. Springer International Publishing, Cham.

An Yang and Sujian Li. 2018. SciDTB: Discourse dependency TreeBank for scientific abstracts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 444–449, Melbourne, Australia. Association for Computational Linguistics.

Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nitish Shirish Keskar, and Caiming Xiong. 2022. Modeling multi-hop question answering as single sequence prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–990, Dublin, Ireland. Association for Computational Linguistics.

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 97–109, Dublin, Ireland. Association for Computational Linguistics.

Longyin Zhang, Yuqing Xing, Fang Kong, Peifeng Li, and Guodong Zhou. 2020. A top-down neural architecture towards text-level parsing of discourse rhetorical structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6386–6395, Online. Association for Computational Linguistics.

Qixuan Zhang, Xinyi Weng, Guangyou Zhou, Yi Zhang, and Jimmy Xiangji Huang. 2022. Arl: An adaptive reinforcement learning framework for complex question answering over knowledge base. *Information Processing & Management*, 59(3):102933.

Tianhua Zhang, Jiaxin Ge, Hongyin Luo, Yung-Sung Chuang, Mingye Gao, Yuan Gong, Xixin Wu, Yoon Kim, Helen Meng, and James Glass. 2023a. Natural language embedded programs for hybrid language symbolic reasoning. *arXiv preprint arXiv:2309.10814*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Changzhi Zhou, Dandan Song, Jing Xu, and Zhijing Wu. 2022. A multi-turn machine reading comprehension framework with rethink mechanism for emotion-cause pair extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6726–6735, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.