A Family of Generative Frameworks for Computational Argument Mining

A dissertation submitted in partial fulfillment of the requirements
for the award of the degree of

Doctor of Philosophy

Submitted by

Nilmadhab Das

Under the supervision of

Prof. Ashish Anand & Prof. V. Vijaya Saradhi

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
Guwahati 781039, Assam, India

May 2026

TH-4046_216101004

TH-4046_216101004

This thesis is dedicated with love and gratitude to my family, whose faith and encouragement have illuminated every step of this journey.

To my late grandfather, whose ideals and guidance continue to shape my path, and to my grandmother, whose warmth and motivation inspire me each day. Your blessings have been my constant strength.

To my parents, whose unconditional love and sacrifices made it possible for me to dream and achieve. Every success of mine is a reflection of your belief in me.

To my wife, my closest companion, whose patience, understanding, and quiet strength carried me through the most challenging moments. Your presence has been my greatest comfort and motivation.

To my uncle, aunt, and my beloved brothers, for their unwavering support and affection that have always reminded me of the strength of family.

To my friends and mentors, who have guided, encouraged, and stood by me throughout this academic pursuit, I extend my heartfelt appreciation. Your influence has left an enduring mark on both my work and my life.

TH-4046_216101004

TH-4046_216101004

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted, in whole or in part, for consideration for any other degree or qualification in this or any other university. Portions of this dissertation are derived from my own previously published or submitted research articles, including preprints available on arXiv. Approximately 15% textual overlap arises from such self-authored works; all such material has been appropriately cited, extended, revised, and integrated to form a coherent and original dissertation, in accordance with institutional guidelines on self-plagiarism. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as explicitly acknowledged in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables, and equations, and includes fewer than 150 figures.

Nilmadhab Das

May 2026

TH-4046_216101004

TH-4046_216101004

Acknowledgements

It gives me immense pleasure to express my deepest gratitude to my supervisors, **Prof. Ashish Anand** and **Prof. V. Vijaya Saradhi**, for their continuous guidance, encouragement, and patience throughout my PhD journey. Their mentorship has been invaluable, not only for the technical insights and critical feedback they provided but also for the academic freedom they allowed me to explore ideas that truly inspired me. Their unwavering support has played a defining role in shaping the direction of this thesis.

I am sincerely thankful to my doctoral committee members, **Prof. Sanasam Ranbir Singh, Prof. Pahi Saikia,** and **Prof. Amit Awekar**, for their constructive comments and thoughtful suggestions that helped refine my research work. My sincere thanks go to all the faculty members of the Department of Computer Science and Engineering, IIT Guwahati, for their direct and indirect contributions toward my academic growth. I would also like to acknowledge the help and cooperation of the technical and administrative staff from various sections of the institute, including the Department of CSE, Computer and Communication Centre, Academic Affairs, and Student Affairs.

I gratefully acknowledge the Ministry of Human Resource and Development, Government of India, for providing financial support through the PhD fellowship. I also thank the IIT Guwahati Technology Innovation and Development Foundation (TI&DF) as a part of the National Mission on Interdisciplinary Cyber-Physical Systems with financial assistance from the Department of Science and Technology, India, through grant number DST/NMICPS/TIH12/IITG/2020 and the Department of CSE for providing the computational facilities essential for carrying out the research presented in this thesis. I wish to extend my gratitude to my friends and colleagues who made this journey pleasant and memorable.

Finally, I owe my deepest gratitude to my family for being my constant pillar of strength. I bow in reverence to the loving memory of my late grandfather, **Mr. Sasanka Sekhar Das**, and to my grandmother, **Mrs. Pratima Das**, whose values, sacrifices, and blessings laid the earliest foundation of my character and aspirations. I express my heartfelt thanks to my father, **Mr. Gobinda Gopal Das**, whose perseverance,

TH-4046_216101004

optimism, and unwavering support have guided me through every stage of life, and to my mother, **Mrs. Tapasi Das**, whose presence continues to inspire and strengthen me each day. I am profoundly thankful to my wife, **Ms. Susoma Sarkar Das**, whose patience, emotional strength, and unshaken faith stood beside me during the most challenging moments of this journey, especially during periods of academic setbacks. I am deeply grateful to my uncle, **Mr. Bijoy Gopal Das**, and my aunt, **Mrs. Ranju Das**, for their constant love, care, and reassurance, and to my brothers, **Nabin** and **Ananta**, whose companionship and support have been a source of joy and motivation. I am equally indebted to my maternal uncles, **Mr. Samaresh Bisai** and **Prof. Alakesh Bisai**, and to my maternal grandparents, **Mr. Parikshit Bisai** and **Mrs. Gitarani Bisai**, for their encouragement, blessings, and faith in me. I thank all my extended family members and those I may have inadvertently missed for their unconditional love and support that have been the foundation of every achievement in my academic journey.

TH-4046_216101004

Abstract

Argument Mining (AM) is a subfield of Natural Language Processing (NLP) that focuses on the automatic extraction of argumentative structures from unstructured text. These structures are composed of Argumentative Components (ACs), such as *claims* and *premises*. They are connected through Argumentative Relations (ARs), including *support* and *attack* relations. By producing structured representations of discourse, AM enables discourse-level reasoning and systematic argumentative analysis. Owing to these capabilities, AM has become increasingly important in applications such as public policy analysis, online deliberation, and the synthesis of scientific and biomedical literature. AM is generally formulated through four core tasks: *(i) Argument Component Identification (**ACI**)*, which detects argumentative text spans; *(ii) Argument Component Classification (**ACC**)*, which assigns roles such as *claim* or *premise* to these spans; *(iii) Argumentative Relation Identification (**ARI**)*, which identifies the existence of relations between components; and *(iv) Argumentative Relation Classification (**ARC**)*, which determines the type of these relations, for example *support* or *attack*.

Most existing models treat these tasks independently or in a pipeline manner. This fragmented design suffers from error propagation. Recent efforts have explored end-to-end paradigms, utilising dependency parsing or graph-based inference, to jointly predict components and relations. While promising, these approaches involve complex post-processing to construct final argumentative structures. Moreover, relation modelling frequently involves Cartesian product assumptions, evaluating all possible AC pairs. This leads to high class imbalance and inefficiency, as the majority of AC pairs are unrelated. Another gap lies in the treatment of semantic signals that guide human understanding of argumentation. For example, prior work often overlooks important linguistic indicators, such as discourse and argumentative markers, or conceptual key phrases that serve as semantic bridges between components. These cues carry significant utility in improving span detection and relation modelling. Their absence from most AM formulations results in shallow text representations that fail to capture deeper discourse-level structure. To address these limitations, this thesis explores several generative modelling approaches to AM that provide an efficient and unified solution. Four distinct frameworks are introduced to advance AM across multiple dimensions

TH-4046_216101004

and domains: *(i) end-to-end generation with marker integration*, *(ii) semantic key phrase enrichment joint modeling*, *(iii) entity-enhanced end-to-end biomedical AM* and *(iv) autoregressive argument structure prediction*.

The first chapter introduces a generative framework, *arg*TANL. It reformulates AM as a text-to-text generation task using Augmented Natural Language (ANL). This formulation enables joint modelling of ACs and ARs by generating label-augmented sequences from raw text. To improve performance, discourse and argumentative markers are incorporated. These markers often precede or co-occur with argumentative spans. Two variants are proposed. Marker-Based Fine-Tuning exposes the model to marker distributions through auxiliary pretraining. Marker-Enhanced *arg*TANL (ME-*arg*TANL) embeds marker information directly in the target sequence. These variants improve AC-type classification. However, it performs poorly on AR-type classification. This indicates that marker-based cues are insufficient to capture deeper semantic dependencies between related ACs.

Motivated by this limitation, the second chapter focuses on enhancing relation modelling by introducing explicit semantic linkages between connected ACs. An ANL-based generative framework is developed that integrates *Related Key Phrases*, concise spans that capture conceptual connections between ACs. By utilising these semantic cues, the framework jointly performs ACC and ARI tasks, achieving SoTA results across multiple AM benchmarks.

The third chapter builds on the second contribution, which improved relational modelling by introducing related key phrases as semantic bridges in ANL-based generation. Since these cues are largely domain-agnostic, this chapter shifts to domain-specific semantic grounding for biomedical texts. Accordingly, the third chapter introduces *Ent-BAM*, an entity-aware generative framework for Biomedical Argument Mining (BAM). Ent-BAM incorporates biomedical entity (co-)occurrence information, particularly PICO-aligned entities inside the target ANL. This formulation enables end-to-end joint modelling of all four BAM tasks: ACI, ACC, ARI, and ARC. Experimental results demonstrate that Ent-BAM achieves SoTA performance across all benchmark BAM datasets.

The final chapter addresses the structural limitations of ANL-based generation in AM. Although such approaches simplify the modelling space, the repetition of text spans introduces redundancies that make the target output lengthy and less efficient. To overcome this, we introduce Autoregressive Argumentative Structure Prediction (AASP), which reconceptualises AM as a sequence of constrained decoding actions using a conditional language model. AASP constructs argument structures through three atomic actions: span identification, boundary pairing, and type labelling. This stepwise formulation mirrors human argumentative reasoning and enables the model to

TH-4046_216101004

more effectively capture long-range dependencies and reasoning chains. Experimental evaluations show that AASP achieves SoTA performance on two AM benchmarks and remains competitive on a third.

This thesis presents generative frameworks that enhance AM tasks' performance by addressing existing challenges in the literature. The proposed approaches improve performance over existing methods and extend the applicability of AM in the era of generative AI.

TH-4046_216101004

TH-4046_216101004

Table of Contents

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

List of Figures

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

List of Tables

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

TH-4046_216101004

List of Abbreviations

Terms	Abbreviations
AASP	Autoregressive Argumentative Structure Prediction
AAE	Argument Annotated Essay
AAE-FG	Fine-Grained Argument Annotated Essay
AAEC	Argument Annotated Essay Corpus
AbstRCT	Abstracts of Randomized Controlled Trials
AC	Argumentative Component
ACC	Argument Component Classification
ACE	Argument Component Extraction
ACI	Argument Component Identification
ACRE	Argument Component and Relation Extraction
ADU	Argumentative Discourse Unit
AF	Argumentation Framework
AM	Argument Mining
A-MKT	Augmented Marker Knowledge Transfer
ANL	Augmented Natural Language
AR	Argumentative Relation
ARC	Argumentative Relation Classification
ARI	Argumentative Relation Identification
ASP	Autoregressive Structured Prediction
BAM	Biomedical Argument Mining
BART	Bidirectional and Auto-Regressive Transformer
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
BIO	Begin–Inside–Outside (tagging scheme)

TH-4046_216101004

CDCP	Consumer Debt Collection Practices
CFPB	Consumer Financial Protection Bureau
CNN	Convolutional Neural Network
COMET	Commonsense Transformers
CPM	Constrained Pointer Mechanism
CRF	Conditional Random Field
DT	Decision Tree
D-MKT	Discourse Marker Knowledge Transfer
DM	Discourse Marker
E-MKT	Marker Knowledge Transfer through Encoding
Ent-BAM	Entity-enriched Biomedical Argument Mining
FFN	Feed-Forward Network
GloVe	Global Vectors for Word Representation
GPU	Graphics Processing Unit
ILP	Integer Linear Programming
KP	Key Phrases
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LR	Logistic Regression
LSTM	Long Short-Term Memory
ME	Maximum Entropy
ME-argTANL	Marker-Enhanced argTANL
MRC	Machine Reading Comprehension
NB	Naive-Bayes
NLP	Natural Language Processing
NER	Named Entity Recognition
NEO	Neoplasm (test split)
GLAU	Glaucoma (test split)
MIX	Mixed diseases (test split)
PLM	Pre-trained Language Model
POS	Parts-of-Speech
Ptr-Net	Pointer Network

TH-4046_216101004

PICO	Patients/Population, Intervention, Control/Comparison, Outcome
QLoRA	Quantized Low-Rank Adaptation
RCT	Randomized Controlled Trial
REGEX	Regular Expression
RF	Random Forest
RNN	Recurrent Neural Network
RPE	Reconstructed Positional Encoding
RoBERTa	Robustly Optimized BERT Approach
SM-MKT	Span-Masked Marker Knowledge Transfer
SoTA	State-of-the-Art
ST	Single-Task
SVM	Support Vector Machine
TANL	Translation between Augmented Natural Language
T5	Text-to-Text Transfer Transformer

TH-4046_216101004

TH-4046_216101004

List of Symbols

W	Input paragraph / argumentative text
n	Number of tokens in W
w_i	i-th token in W
$w_{i:j}$	Contiguous text span from token i to j in W
C	Set/list of Argument Components (ACs)
R	Set/list of Argument Relations (ARs)
t_i	Type of the i-th AC (e.g., Claim, Premise)
s_i, e_i	Start and end token indices of the i-th AC in W
h_j, t_j	Head (source) and tail (target) ACs in the j-th relation
r_j	Type/label of the j-th relation (e.g., Support, Attack)
C_{ACI}	Set of AC spans for Argument Component Identification (ACI)
C_{ACC}	Set of typed ACs for Argument Component Classification (ACC)
R_{ARI}	Set of AC pairs (relations) for Argument Relation Identification (ARI)
R_{ARC}	Set of typed relations for Argument Relation Classification (ARC)
T^c	Set of possible AC types; n_c is the number of AC types
T^r	Set of possible AR types; n_r is the number of AR types
c_p	p-th AC/span
t_p	Type of AC c_p
c_p^s, c_p^e	Start and end token indices of AC c_p in W
r_{pq}	Relation type between head AC c_p and tail AC c_q
S	Set of special tokens used in ANL formatting
$[$, $]$	Start and end of an augmented/component label in ANL
$(()$, $))$	Parentheses used for grouping in augmented labels
$=$	Relation assignment operator in ANL

TH-4046_216101004

$\mid$	Label separator in ANL
KP	Set of LLM-extracted related key-phrase pairs from ARs
e_{i_h}, e_{i_t}	Head and tail key phrases of the i-th pair in KP
y_i	Action tuple variable at step i
Y_i	Action space at step i
A	Set of span-identifying symbols $\{\langle m\rangle, \rightarrow, \langle /m\rangle\}$
B_n	Set defining valid boundary-pairing actions up to step n
Z_n	Set defining valid type-labeling actions up to step n
L	Combined set of AC and AR labels used for type-labeling
a_i, b_i, z_i	Span-identifying, boundary-pairing, and type-labeling actions at step i
h_n	Decoder hidden state vector at step n
p_n	AC span representation at step n (e.g., $[h_n^\top; h_{b_n}^\top]^\top$)
$\langle m\rangle$, $\langle /m\rangle$	Start and end markers for AC spans
$\rightarrow$	Copy operator that copies input tokens left-to-right

TH-4046_216101004

Chapter 1

Introduction

1.1 Overview

Computational Argument Mining (AM) [7] is an interdisciplinary research area within Natural Language Processing (NLP) that seeks to automatically identify and extract arguments from unstructured text. AM aims to uncover the reasoning that underpins human communication. At its core, AM operates with ACs, such as *claims* and *premises*, and ARs, such as *support* or *attack*. Together, these form the building blocks of reasoning-oriented discourse. Figure 1.1 illustrates the ACs and ARs through examples. Traditionally, AM is decomposed into four subtasks: *Argument Component*

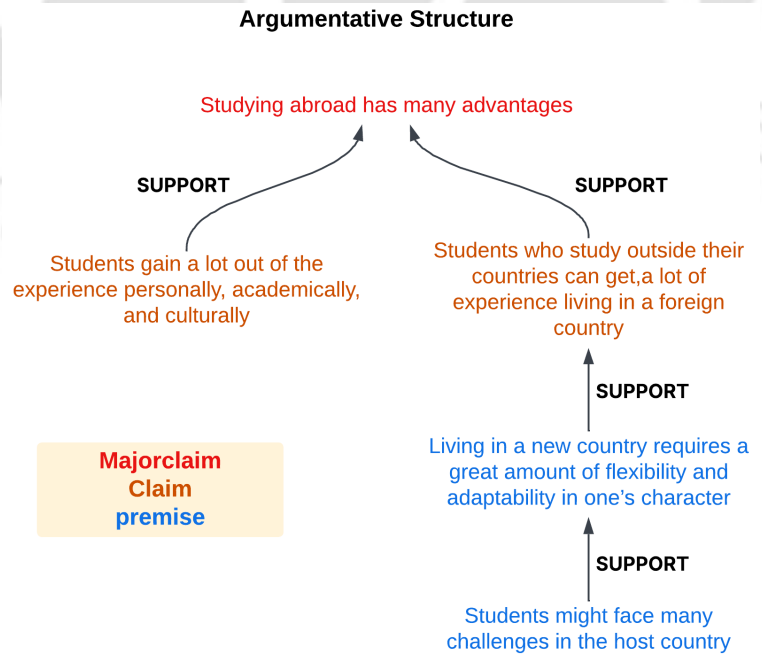

Fig. 1.1 Argumentative structure showing the relation hierarchy of Claims and Premises for the MajorClaim *"Studying abroad has many advantages."*

TH-4046_216101004

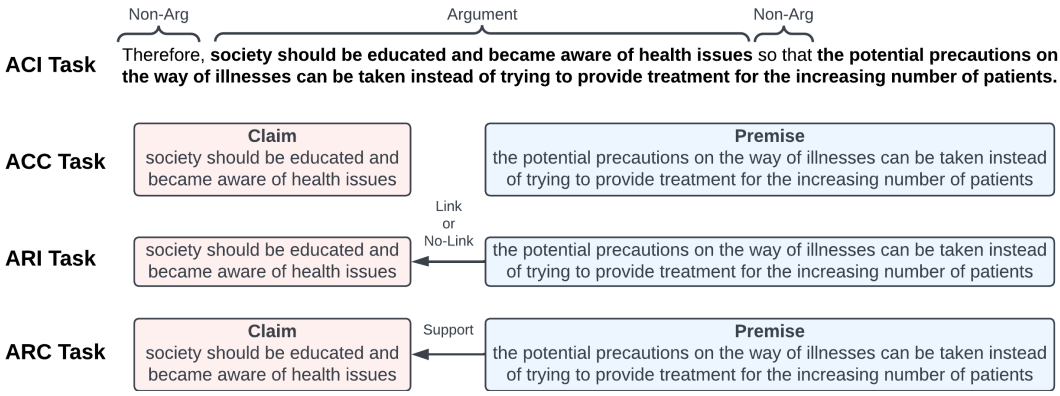

Fig. 1.2 Illustration of different Argument Mining tasks: Argument Component Identification (ACI), Argument Component Classification (ACC), Argument Relation Identification (ARI), and Argument Relation Classification (ARC).

Identification (ACI), which detects argumentative spans within text; *Argument Component Classification (ACC)*, which distinguishes claims from premises; *Argumentative Relation Identification (ARI)*, which determines whether a link exists between two components; and *Argumentative Relation Classification (ARC)*, which assigns a relation type, such as supportive or opposing. The same is illustrated in Figure 1.2. These subtasks are conceptually interdependent. However, much of the early research treated them independently or in sequential pipelines [8]. This results in fragmented systems with limited coherence. The rise of large-scale textual data in domains such as public debates, parliamentary proceedings, online discussions, academic essays, and news media has made AM an increasingly important tool for both academic inquiry and practical applications. By rendering implicit reasoning structures explicit, AM allows the analysis of discourse at a semantic and pragmatic level. Thus, it offers insights into how people argue, persuade, and justify decisions. This not only enables advanced computational reasoning but also creates opportunities for applications in automated essay scoring [9], legal decision support systems [10], healthcare argument analysis [8] etc.

1.2 Significance

The significance of AM lies in its transformative potential to bridge human reasoning and computational processing. ***Firstly***, in public policy and governance, AM enables large-scale analysis of deliberative texts such as parliamentary debates or opinion editorials, assisting policymakers in synthesizing diverse viewpoints [11]. In education, AM-driven tools can analyze student essays [12] or debate transcripts, providing structured feedback on argument quality, coherence, and logical soundness. Similarly, in online platforms and media [13], AM can help detect persuasive techniques, misinfor-

TH-4046_216101004

mation, or patterns of reasoning in large-scale discussions, offering valuable insights for both researchers and practitioners. ***Secondly***, from a research perspective, AM represents one of the most challenging problems in NLP. Unlike lexical-level tasks such as sentiment classification or named entity recognition, AM requires the integration of syntax, semantics, pragmatics, and discourse-level understanding. Argumentation is inherently complex: it relies on contextual interpretation, background knowledge, and subtle linguistic cues such as discourse markers ("however," "therefore") or rhetorical strategies [14]. As such, AM provides a rich testing ground for developing advanced machine learning models capable of reasoning beyond shallow text features. ***Third***, AM plays a critical role in fostering transparency and explainability in AI systems [15]. While neural models have achieved remarkable progress in text classification and generation, their reasoning processes often remain opaque. By explicitly modeling argumentative structures, AM can serve as a foundation for interpretable AI, offering users insights into why a system reaches a particular conclusion. This is particularly significant in high-stakes domains such as law, journalism, and public deliberation, where accountability and justification are as important as predictive accuracy.

1.3 Motivation and Research Gaps

Despite substantial progress, AM research continues to face significant challenges that limit its practical deployment. One of the primary issues is the fragmented treatment of subtasks in the earlier studies [16]. Pipeline-based models often propagate errors from one stage to the next: a misclassified component at the ACC stage, for example, directly impacts relation identification and classification downstream. Independent modeling of tasks also prevents information sharing between components and relations, leading to shallow representations that fail to capture the holistic nature of argumentative discourse. A ***second*** limitation lies in the inefficiency of relation modeling. Many existing approaches rely on Cartesian product assumptions [17], evaluating every possible pair of components in a document. Since the majority of component pairs are unrelated, this approach not only creates extreme class imbalance but also leads to unnecessary computational overhead. A ***third*** challenge is the underutilization of semantic and linguistic cues. Humans often rely on subtle indicators such as discourse markers ("because," "although"), argumentative markers ("the point is," "in fact"), and conceptual key phrases that signal relationships between claims and premises [14]. However, many computational models ignore these weak yet informative signals, resulting in representations that miss deeper discourse-level structures. ***Finally***, most existing generative approaches represent arguments as flattened sequences output [18], which can capture local relationships but struggle with long-range dependencies and the stepwise nature of the reasoning. Human argumentation is inherently procedural:

TH-4046_216101004

it unfolds through a sequence of steps where components are introduced, connected, and expanded upon. Capturing this process computationally requires models that can mirror the autoregressive and constrained nature of human reasoning.

These gaps highlight the need for efficient modeling approaches that unify subtasks, reduce inefficiencies, leverage semantic signals, and introduce richer structural representations. Generative formulations, particularly when combined with customizations, offer a powerful paradigm for overcoming the limitations of fragmented and shallow approaches. This motivation underpins the present research, which seeks to design a new class of AM frameworks that are robust, efficient, and capable of capturing the depth of human argumentative structure.

1.4 Problem Statement

We represent an argumentative paragraph as $W = w_1, w_2, w_3, \ldots, w_n$, where n denotes the total number of tokens in the sequence. A contiguous segment spanning from token w_i to token w_j is denoted as $w_{i:j}$. Among the four core tasks in Argument Mining (AM), the first task, *Argument Component Identification (ACI)*, is concerned with locating all argumentative spans in the text. Given W as input, ACI predicts a set of span boundaries $C_{ACI} = \{(s_i, e_i)\}_{i=1}^{n}$, where each pair (s_i, e_i) specifies the start and end indices of a candidate Argument Component (AC) corresponding to the span $w_{s_i:e_i}$. The second task, *Argument Component Classification (ACC)*, assigns a semantic label t_i, such as Premise or Claim, to each span in C_{ACI}, yielding the set $C_{ACC} = \{(t_i, s_i, e_i)\}_{i=1}^{n}$. The third task, *Argument Relation Identification (ARI)*, aims to identify directed links between components and produces a relation set $R_{ARI} = \{(h_j, t_j)\}_{j=1}^{m}$, where each pair denotes a connection from a head component h_j to a tail component t_j, with $h_j, t_j \in C_{ACC}$. The final task, *Argument Relation Classification (ARC)*, assigns a relation label r_j, such as Support or Attack, to each identified pair (h_j, t_j), resulting in the set $R_{ARC} = \{(h_j, t_j, r_j)\}_{j=1}^{m}$.

Broadly, the literature can be categorised into three paradigms: **Standalone**, **Joint**, and **End-to-End** formulations. In **Standalone formulations**, each task is solved independently, often using outputs from preceding ones (e.g., solving ACC or ARI after ACI). **Joint formulations** address a subset of tasks together, such as ACC+ARI or ACC+ARI+ARC, to reduce error propagation and enhance inter-task learning. **End-to-end formulations** aim to predict all four tasks jointly from plain text, offering full structural modelling but with higher complexity.

In this thesis, we consider two distinct problem settings. **The first setting** assumes that the argumentative spans (i.e., the output of ACI) are provided as input. Under this setting, the objective is to solve the ACC, ARI, and ARC tasks jointly. **The second**

TH-4046_216101004

setting, termed the *End-to-End* framework, considers the plain text paragraph W as the only input. The goal is to jointly infer all four AM tasks. This unified formulation aims to simulate a realistic AM scenario where no prior structural information is available. Thereby, enabling the model to learn the entire argumentative structure directly from unannotated text.

1.5 Contributions

This thesis proposes four generative approaches to enhance AM systems. The first contribution presents a simple text-to-text generation framework. Given plain text as input, the model generates an ANL output containing ACs and ARs. Argumentative structures are then reconstructed through postprocessing. The second contribution extends this idea by adding key phrases that link related ACs. Given AC spans as input, the model jointly solves ACC, ARI, and ARC by appending these key phrases to the ANL output. The third contribution adapts this framework to the biomedical domain. Here, key phrases are replaced with biomedical entities extracted from ACs. To form an entity-enhanced representation, we place these entities at the beginning of the ANL output, enabling end-to-end modeling of all four AM tasks from plain text. The final contribution models AM as a sequence of constrained decoding actions. Using three structured actions, the model builds argumentative structures step by step, resembling human reasoning. Together, these contributions move from simple generation to structured and semantically rich modelling, resulting in a family of AM frameworks.

The key contributions of this thesis are summarised as follows:

1. **Exploration of Marker-Based Approaches in Argument Mining through Augmented Natural Language:** Reformulates AM as a generative task using ANL, integrating discourse markers as weak semantic cues to enhance component and relation modeling.

2. **On the Role of Key Phrases in Argument Mining:** Introduces a key phrase-aware generative framework that embeds related key phrases to the ANL output, enriching relational understanding between ACs.

3. **Ent-BAM: A Generative Framework for Biomedical Argument Mining Enriched with Entity Information:** Extends generative AM to the biomedical domain by embedding biomedical entities into the ANL output for entity-enhanced modeling of all four tasks.

4. **End-to-End Argument Mining through Autoregressive Argumentative Structure Prediction:** Proposes AASP, a constrained decoding framework that models AM as sequential actions, enabling structured and procedural generation of argumentative structures.

TH-4046_216101004

Together, these contributions establish a comprehensive foundation for next-generation AM systems adaptable across both general and specialised argumentative domains. Extensive experiments are conducted on four standard AM benchmarks: *Argument Annotated Essay (AAE)* [12], *Consumer Debt Collection Practices (CDCP)* [11], and *Fine-Grained Argument Annotated Essay (AAE-FG)* [19], along with the biomedical benchmark *Abstracts of Randomised Clinical Trials (AbstRCT)* [20]. Across all datasets, the proposed frameworks achieve consistent improvements over competitive baselines, yielding SoTA or near SoTA results.

Thesis Outline

This thesis is organised as follows (See Fig. 1.3):

Chapter 1: Introduction introduces the field of AM, outlines its key tasks, highlights existing limitations, and presents the research motivation, objectives, and overall contributions.

Chapter 2: Literature Review surveys prior works on AM, including pipeline, joint, and end-to-end formulations, with emphasis on semantic cues, discourse signals, and structural modelling.

Chapter 3: Exploration of Marker-Based Approaches in Argument Mining through Augmented Natural Language presents a generative ANL-based formulation that integrates different markers as weak semantic cues to improve AM modeling.

Chapter 4: On the Role of Key Phrases in Argument Mining introduces a key phrase-aware generative framework that incorporates related key phrases into the ANL output to enrich relational reasoning between ACs.

Chapter 5: Ent-BAM: A Generative Framework for Biomedical Argument Mining Enriched with Entity Information extends the generative approach to the biomed-

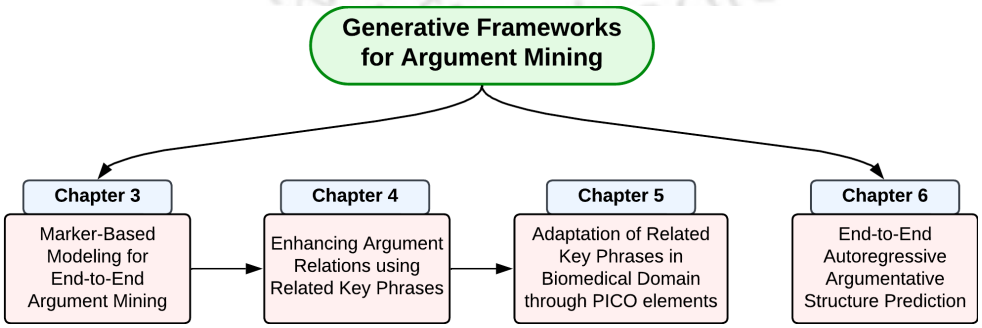

Fig. 1.3 Overview of the Thesis Contributions

TH-4046_216101004

ical domain by embedding biomedical entities into the ANL output, enabling entity-aware modeling of all four AM tasks.

Chapter 6: End-to-End Argument Mining through Autoregressive Argumentative Structure Prediction proposes an autoregressive framework that models AM as a sequence of constrained decoding actions, incrementally constructing argument structures in a human-like manner.

Chapter 7: Conclusions and Future Work summarises the findings, emphasising advances in unified modelling, semantic enrichment, and structural representation, and outlines directions for future research in AM.

TH-4046_216101004

TH-4046_216101004

Chapter 2

Literature Survey

2.1 Early Stages of Argument Mining

Before moving towards the recent advancements of argument mining, we will briefly touch upon the origins of it. In the domain of *logic and critical thinking*, there is a process called *"Argumentation"* [21]. In this process, deeper analysis and understanding of arguments are being studied with a framework called *"Argumentation Framework (AF)"* [21]. Substantial efforts have been made to construct the best possible AF over the years [22]. In general, the commonality shared by all AFs comprises a finite set of arguments (X) and a set of relations $D \in X \times X$. Rather than presenting the mathematical formulation of AF, we provide a conceptual overview of the underlying notion of *argumentation*. In simple terms, *"Argumentation"* refers to the *"process of constructing, presenting, and evaluating arguments with a compatible argumentation framework (AF)"* [22]. It encompasses the study of how arguments are formulated, supported, and presented in various contexts, as well as the analysis of their logical, rhetorical, and persuasive aspects. It also focuses on understanding the principles, norms, and processes involved in constructing and evaluating arguments. This includes examining the structure of arguments, the patterns or schemes used to formulate them, the fallacies or errors in reasoning that can undermine their strength, and the contextual factors that influence their effectiveness. The emergence of computational argument mining can be attributed to several factors, despite the existence of argumentation theory and frameworks. While argumentation theory has long provided a conceptual framework for analyzing arguments, argument mining utilizes computational methods along with machine learning techniques to automatically extract and analyze arguments from large volumes of textual data. This enables the analysis of large datasets and facilitates the identification of arguments on a scale that would be challenging or time-consuming for humans to accomplish manually.

TH-4046_216101004

2.2 Overview of the Existing Literature

In the previous studies of argument mining, the distinction between various sub-tasks was not clearly defined. However, several attempts are made to categorize them into four distinct sub-tasks. Over time, the methodologies for all four sub-tasks in argument mining have evolved from using statistical classifier-based models [23] to the current era of large language models [18]. Handcrafted features [24] have been replaced by representation learning [25], marking a significant advancement. Additionally, the granularity of these sub-tasks also has evolved from a sentence level [26] to a more detailed segment level [27]. Several survey works have contributed significantly to the development of argument mining research. Lippi and Torroni [28] presented a comprehensive survey covering the foundational concepts, major computational approaches, and challenges in argumentation mining. Stede and Schneider [29] provided a detailed overview of the linguistic, theoretical, and computational aspects of argumentation mining in their book. Furthermore, Lawrence and Reed [7] summarized the progress of the field by discussing key datasets, methodologies, evaluation strategies, and emerging research directions in computational argumentation. One notable challenge in sentence-level argument mining was accurately determining the relations (e.g., support or attack) for sentences that contained conflicting multiple premises. Furthermore, a single sentence could encompass multiple claims and premises, complicating the differentiation of various argument components when considering the sentence as a whole. To address these difficulties, researchers gradually shifted their focus from sentence-level sub-tasks to segment level sub-tasks. Further discussions on all four sub-tasks will be provided in the subsequent sections. This section will be divided into six subsections: (2.2.1) Review of identifying argumentative or non-argumentative text (ACI Task), (2.2.2) Review of clausal properties classification (ACC Task), i.e., determining if a given argumentative text is a premise or a claim, (2.2.3) Review of identifying and classifying relational properties between ACs (ARI & ARC Tasks), (2.2.4) Joint formulations of different AM tasks, (2.2.5) End-to-End approaches, which solves all four tasks of AM, and (2.2.6) Biomedical Argument Mining, which is a biomedical domain adaptation of AM.

2.2.1 Argumentative or Non-argumentative Text Identification (ACI Task)

In computational argument mining, this sub-task deals with the identification of a piece of text, whether it is argumentative or not. The very first approach to tackle this task was developed in [24], where authors considered argument as a sentence-level component. The experiment was performed in legal texts by constructing manually-

Features
1. Syntactic / Positional Information
2. Lexicons
3. Semantic Similarity
4. Sentiment
5. Embeddings
6. Patterns (REGEX)
7. Discourse Markers
8. Bag-of-words
9. Wikipedia-based
10. PMI
11. Emoticons

Table 2.1 Most used features for several argument mining tasks [5]

engineered features (See Table 2.1) such as Adverbs, Verbs, Modal auxiliary, Word couples, N-grams, Text statistics, Punctuation, Keywords, and Parse-tree features. They used Araucaria corpus [30], which consists of data from various domains (newspaper editorials, advertising, judicial judgments, parliamentary debates, etc.). They performed the classification with a multinomial Naive-Bayes classifier and a maximum entropy model. The highest accuracy 0.74 was obtained with the multinomial Naive-Bayes classifier.

Similarly, in another experiment [26], the authors proposed a two-step approach for argument detection from social media, which resembles the method employed in a previous work[24]. Along with the handcrafted features, they used a logistic regression classifier for the classification of a sentence if it is argumentative or not. They used a corpus from social media that was on renewable energy sources in Greek language. By employing this method, they achieved a significant performance improvement, increasing the F-score from base-case 0.21 to 0.77, enabling the identification of argument-containing sentences. This approach was further enhanced in [31], where authors used the Continuous Random Field (CRF) classifier for detecting arguments at the sentence level. They used the “word2vec" [38] representation with features such as POS tags and a small lexicon of cue words of the Greek language articles and trained a CRF-based classifier to detect argumentative texts.

In a study by Madnani et al. [32], the authors aimed to categorize argumentative discussion text into two components: *argumentative content* and the linguistic material used to organize or present such content, referred to as the *shell*. For example, in the sentence: “*So I strongly agree on a point that,* we should never think twice to use our military force…to keep the safety of European people.", the italicized segment constitutes a *shell*. To identify such shell segments, the authors proposed

Paper	Features	Model	Granularity	Type	Dataset	Result
[24]	General Features	NB, ME	Sentence	Classification	Araucaria	F1 upto 0.73
[26]	General Features	LR	Sentence	Classification	Greek Social Media	F1 upto 0.77
[31]	Word2Vec	CRF	Sentence	Classification	Greek Social Media	F1 upto 0.75
[32]	Lexical	Rule-Based + CRF	Segment	Sequence Labeling	Own Corpus	F1 upto 0.61
[1]	Semantic, Syntactic, Structural, Pragamtic	Bi-LSTM	Segment	Sequence Labeling	AAE[12], Editorials[33], Web Discourse[34]	In-domain: F1 upto 0.88, Cross-domain: F1 upto 0.69
[35]	GloVe, BERT, Flair embeddings	Bi-LSTM	Segment	Sequence Labeling	AAE	F1 upto 0.87
[36]	ELMo, BERT embeddings	Bi-LSTM-CRF	Segment	Sequence Labeling	AAE	F1 upto 0.90
[37]	BERT embedding	BERT + CRF	Segment	Sequence Labeling	Own Corpus	In-domain: F1 upto 0.75, Croess-domain: F1 upto 0.68

Table 2.2 A summary of different approaches used for ***Argument Component Identification***.

three methods: (i) a *rule-based system*, (ii) a *supervised probabilistic sequence model* based on *conditional random fields (CRF)*, and (iii) a *hybrid approach* that combined the rule-based and CRF-based predictions using a voting-based mechanism introduced in the paper. The rule-based system consisted of 25 manually crafted regular expression (REGEX) patterns designed to capture common discourse markers, sentential openers, hedges, stance indicators, and other organizational cues that typically signify shell language. This system achieved an F-score of 0.44. The supervised sequence model is a conditional random field (CRF), trained using lexical and contextual features. These include token frequency statistics such as thresholded frequency indicators, prompt-level occurrence proportions, orthographic cues, positional features, transitions between labels, and two features incorporating predictions from the rule-based system. The hybrid system, which integrated the outputs of both the rule-based patterns and the CRF model through a weighted combination strategy, achieved an improved F-score of 0.61. When compared to a baseline system and human annotators, the hybrid method outperformed both the rule-based system and the baseline. However, it still remained below the human annotator performance, which achieved an F-score of 0.74. It is important to highlight that although the detected *shell* material does not itself identify argumentative content, its presence signals an increased likelihood that an argument component will follow.

One of the very first approaches to segment-level ADU identification was done in [39]. In this work, the authors proposed a supervised learning approach with a specific focus on identifying the boundaries of ADUs. They used two Naive-Bayes classifiers for "*Proposition Boundary Learning*"; the first one is to find the first word of a proposition and another to get the last word. Training data consists of manually extracted propositions. The segmentation process starts by splitting the text into words. Subsequently, a comprehensive set of features is generated for each word such as its length, Part-of-Speech tags, the preceding and following words, punctuation marks, etc. The trained classifiers labeled each word as either a "start" or an "end." In cases where start-end ordering is not maintained, indicating ambiguity in the length of the proposition, the probabilities calculated by the classifiers are utilized to resolve such situations.

While these results are promising, it is important to note that most argumentative text identification methods relied solely on sentence-intrinsic features. As a consequence, these approaches lacked robustness when handling texts that may be argumentative in one context but not in another. This indicates that contextual information was not adequately captured, as these methods largely treated context only at the sentence level. The significance of contextual domain knowledge was initially investigated in [40]. Through the analysis of diverse corpora, the study delved into the essential types of knowledge necessary for the development of an effective argument

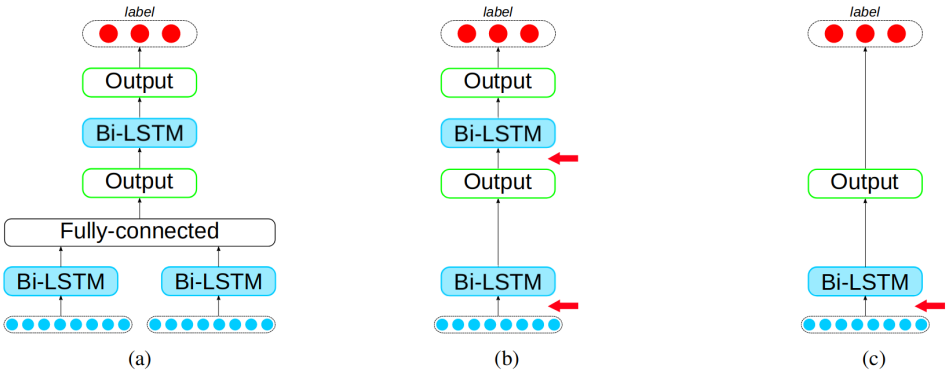

Fig. 2.1 (a) Original model of [1]. (b) and (c) Modified models with attention heads inserted at red-arrow positions

mining system. This exploration revealed that approximately 75% of cases demand contextual knowledge to precisely identify arguments concerning a contentious matter.

Now, one of the earliest neural network implementations [1] for argument mining was formulated as a task of argument unit segmentation using sequence labeling, which simply means, the detection of ADUs at the token level. They performed a cross-domain experiment with three datasets from different domains such as student essay [12], news editorials [33] and web-discourse [34]. All three corpora were annotated at the segment level. For feature selection, they chose a combination of semantic, syntactic, structural, and pragmatic features. The semantic feature was represented using a Bag-of-Words approach, while syntactic features were captured through Parts-of-Speech analysis. Structural features encompassed sentence, clause, and phrase-level attributes, and pragmatics features included token-before-marker, beginning-of-marker, etc. The implementation employed two Bidirectional Long Short-Term Memory (Bi-LSTM) units. One unit processed the semantic feature input, while the other handled the concatenated input of syntactic, structural, and pragmatic features. The task formulation aimed at sequence labeling at the token level, utilizing the standard BIO-tagging scheme. The experimental results demonstrated that the model performed well when applied to in-domain data. However, its performance on cross-domain data was not as promising.

Another experiment was performed in [35], who made three different variants of the Bi-LSTM-based neural model (see Figure 2.1) of [1] by inserting attention-heads [41] at different positions to check the efficiency of attention-heads in the argument unit segmentation task. Experiments on the AAE dataset [12] showed that although the modified models did not exhibit any performance degradation, the added attention heads did not yield any substantial improvement either. In a separate experiment conducted in [36], a novel approach was adopted by leveraging BERT, ELMo, Flair, and GloVe embedding as inputs to a BiLSTM-CRF model for token-level

sequence labeling in argument unit segmentation. Notably, no manually-engineered features were utilized in this approach. The experiment yielded a macro F1 score of up to 0.90 on the AAE dataset. This performance surpassed the achievements of all previous tasks on argument unit segmentation conducted on the same dataset. Another similar experiment was performed in [37] for the same task. They proposed a crowd-annotated dataset from the web-discourse domain and performed several experiments with $BERT_{BASE}$, $BERT_{LARGE}$ and $BERT_{BASE} + CRF$ for in-domain and cross-domain experiments. In all the variants, the input was fed as the concatenated topic sentence and input sentence and expected output of unit segmented ADU at the token level. Results again indicate that in-domain performance was better than cross-domain performance.

2.2.2 Clausal Properties Classification (ACC Task)

This sub-task determines if the previously identified argumentative text is a premise or a claim i.e., knowing their clausal properties. In the early days of argument mining, researchers used different terminologies to differentiate the clausal property types like the conclusion, value, policy, testimony, fact, reference, etc. The standardized differentiation comes after the introduction of several standard corpora like [42], [43], where authors clearly mentioned two types of major clausal properties “claim" and “premise", where a premise will give the justifications of the claim. One of the early tasks of clausal property identification was done in [44], where the authors performed sentence-level premise and conclusion classification of ECHR (own) corpus with the help of rich features handcrafted for this task only. They used Support Vector Machine (SVM) and Context Free Grammar (CFG) for the classification. Another experiment on the same dataset was done in [45], where the authors experimented with the “*Scheme-Wise*" classification instead of the clausal properties identification. They defined five types of schemes namely “example", “cause to effect", “practical reasoning", “consequences" and “verbal classification". They used statistical classifiers to identify these schemes.

In a study [46] of online user comments, the authors analyzed the clauses at the segment level to classify them into three types: “unverifiable", “verifiable non-experiential", and “verifiable experiential". Different types of supports, such as “reasons", “evidence", and “optional evidence", were associated with these clauses. SVM classifiers trained on various features showed significant improvements. However, further research is needed to establish relations between clause types and determine the argument structure. In a subsequent work [49], the authors revised the clause categories as “fact", “testimony", “value", “policy", and “reference" and created the Consumer Debt Collection Practices (CDCP) corpus, comprising 731 user comments.

Paper	Features	Model	Granularity	Type	Dataset	Result
[44]	General Features	SVM, CFG	Sentence	Classification	ECHR (Own)	F1 upto 0.74
[46]	General Features	SVM	Segment	Classification	CDCP	F1 upto 0.69
[23]	Structural, Lexical, Syntactic, Indicators, Contextual	SVM, NB, DT, RF	Segment	Classification	AAE	F1 upto 0.726
[47]	General Features, Discourse Markers, Topical Similarity	NB, REGEX-rule based	Segment	Boundary Identification	AIFdb	F1 upto 0.83
[48]	GloVe Embedding	Bi-LSTM-CRF based Multi-Task Learning (Low Resource)	Segment	Sequence Labeling	Six different Datasets	F1 upto 0.58
[2]	BERT embeddings	BERT, Pointer Network, CRF	Segment	Span Identification and Sequence Labeling	AAE	F1 upto 0.64

Table 2.3 A summary of different early approaches used for ***Clausal Properties Classification***.

<*claim*> because <*premise*>.

Since <*premise*> I strongly believe that <*claim*>.

As the fact is saying <*premise*> hence <*claim*>.

<*premise*>. Therefore, <*claim*>.

Fig. 2.2 Examples of discourse patterns connecting premises and claims.

Apart from this, another dataset was proposed in [50], it was clearly differentiating between “claim", “premise" and “Major Claim", which was based upon the persuasive essay, known as AAE corpus. In a later work [23], the authors used two classifiers to identify those clausal properties. The first one focuses on identifying argument components, and the second one performs a multi-class classification, categorizing each clause using various feature types such as structural features (such as the position and punctuation of the argument component), lexical features (n-grams, verbs, adverbs, and modals), syntactic features, discourse indicators features, and contextual features.

Whereas, the previous work used discourse markers as features, another work by [47] used them as a method for identifying arguments. *“Reasoning/discourse markers"* are words or phrases that indicate the type of reasoning used in an argument.

In Figure 2.2, each underlined text segments are some instances of reasoning/discourse markers because they are connecting the different argument components (<*claims*> and <*premises*>). So, by detecting reasoning/discourse markers, it is possible to detect the different premises, claims and their corresponding relations. Because, some markers are supportive in nature, for example, “because”, “it follows that”, “therefore”, “I strongly believe that” etc., by which we can directly say that, the claim is *“supported”* by the given premise; whereas some are attacking in nature, for example, “but”, “however”, “nevertheless”, “on the other hand” etc., which *“attacks”* the claim with the given premise. In addition to this approach, the authors conducted experiments with two alternative methods: one focused on topical similarity, while the other involved a feature-based supervised machine learning classification task based on argumentation schemes, as mentioned earlier. The detection of segment-level clausal properties along with their boundary identification was performed using all three methods. The outcomes obtained were comparable to those achieved by existing methods in the field. Several other studies [12, 14, 51] have also shown that markers are crucial signals for ADUs. Markers as a lexical feature for classifying argument components with a feature-based multiclass classification were used by [12]. Later, 1131 argumentative markers were extracted from different datasets by [14] to check the efficacy of AM tasks. An improved span representation utilizing the information of extracted markers was proposed by them. The contribution of markers for AM tasks in the Reddit social discussion thread was also explored by [51]. 69 Reddit-specific markers were extracted, and selective masked language modeling (sMLM)

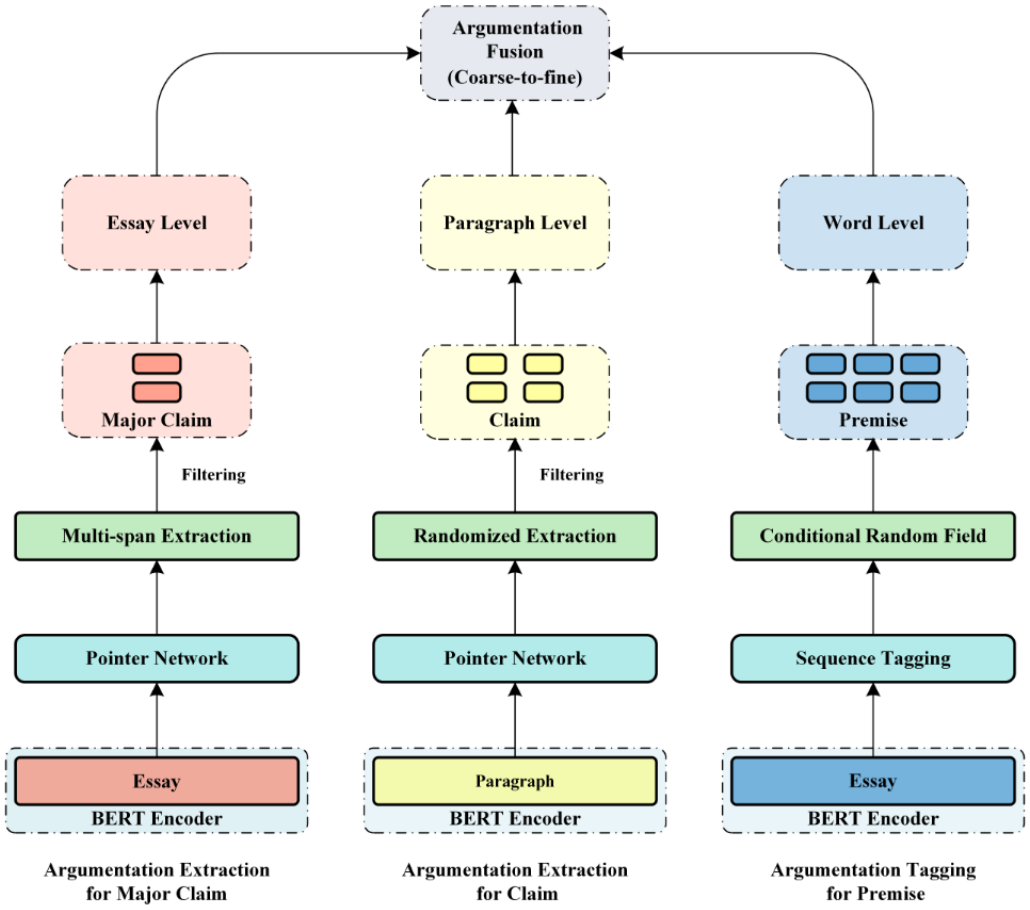

Fig. 2.3 Clausal Properties Identification at Multi-Scales [2]

was performed by them by masking those markers for domain adaptation. Later, the resultant model was used for Argument Component Identification. A template-based approach was designed by them to predict the marker-like tokens in masked positions to predict the Relation Type between given components.

In the era of traditional neural models, the classification of clauses has evolved into a token-level sequence tagging task. This is because proper boundary detection of the clauses becomes necessary when transitioning from the sentence level to the segment level. In an experiment [48], the authors implemented a multi-task learning setup where argument unit segmentation and premise-claim classification were jointly performed using a sequence tagging approach. The model employed Bi-LSTM-CRF for sequence modeling and utilized GloVe [52] embeddings. The experiment aimed to assess the effectiveness of multi-task learning compared to single-task learning in a low-resource setting and to enhance generalizability across domains. The results, obtained from six different domain datasets, demonstrated that multi-task learning outperformed single-task learning in the classification of argument components. Additionally, it successfully reduced the occurrence of invalid clause sequencing observed in previous methods.

Another method proposed in [2] with the AAE corpus [50] highlighted the varying scales (see Figure 2.3) of clauses in argumentation. They defined a Major Claim as a clause at the essay level, a Claim as a clause at the paragraph level, and a Premise as a clause at the token level component. To process each clause, they employed distinct inputs and anticipated candidate spans as outputs. For instance, since the Major Claim is an essay-level clause, the entire essay was fed into BERT. For Claims, they fed paragraphs, and for Premises, they fed essays into BERT. They utilized a pointer network [53] to identify the spans of Major Claim and Claim candidates, employing a Multi-span extraction strategy and a Randomized extraction strategy, respectively. However, for Premises, they employed a sequence-tagging approach using CRF with the standard BIO scheme to fetch the potential premise candidates. This work provides insights into considering the contextualization of different clausal elements in different levels.

Till now, several methods are discussed, where relation prediction is modeled jointly with ADU identification and ADU classification. In the next subsection, we will focus on the methods, that are particularly used for relation prediction only.

2.2.3 Identifying and Classifying Relational Properties (ARI & ARC Tasks)

One of the dedicated approaches for relation prediction was done in [54], where the authors filtered the argumentative sentences with the basis of sentence pair relation classification. More precisely, given two sentences, if the relation between them is support or attack, then both sentences are considered to be argumentative. To identify conflict relations, Peldszus *et al.* [55] examined the texts of counter-attacking nature. These instances include phrases like "Even though..." or "It has been claimed that...however...". The authors utilized these phrases to detect criticisms of the arguments and strengthen corresponding points. They identified textual segments as “proponent" or “opponent" using a linear log-loss model achieving an F-score of 0.64. Also, another work by Nguyen *et al.* [56] contends that simply examining the content of paired elements to establish relationships fails to exploit the available information to its fullest extent. The authors suggest an alternative method that incorporates contextual features derived from the surrounding sentences of both the source and target components, in addition to general topic information. The authors of [39] used the WordNet information to experiment with the semantic similarity between two clauses. They used topical similarity instead of semantic similarity to address this task. Although the results were promising, they were missing the broader context of knowledge.

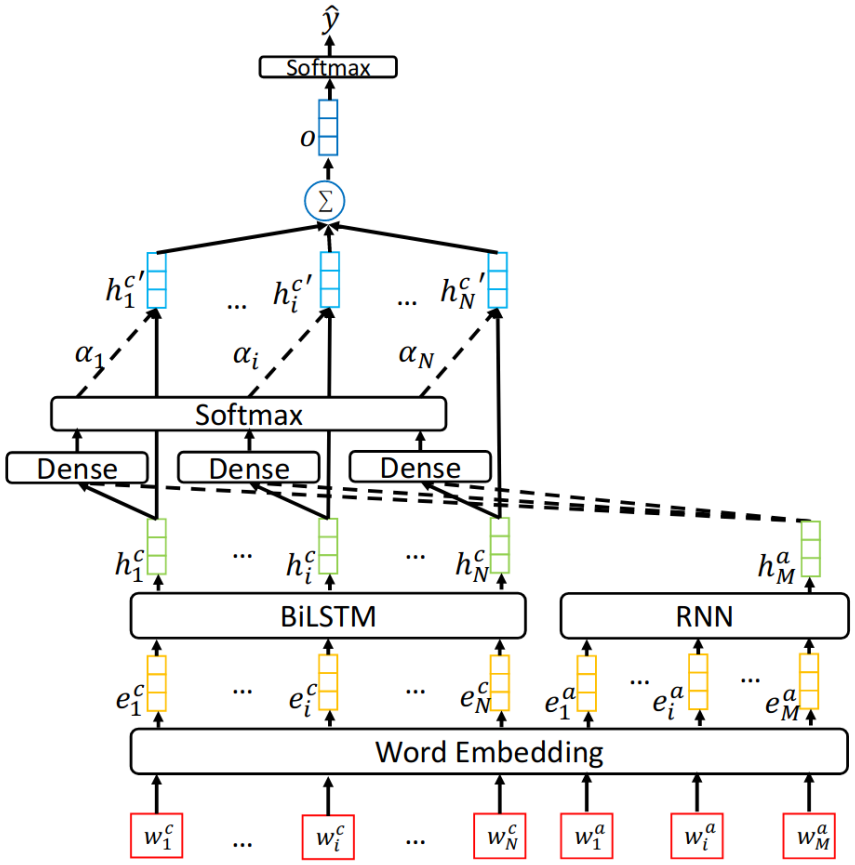

Fig. 2.4 Lexicon-guided neural network [3]

Coming to the neural models, an experiment in [57], the authors used two Bi-LSTM to encode two texts separately at the sentence level. The outputs of those Bi-LSTMs were concatenated and fed into a dense32 ReLU and then through a softmax layer to do the multi-class classification of support, attack, or no-relation. Another approach in [42], the authors focused on sentence-level relation prediction by utilizing a Bi-LSTM along with cosine similarity features and an attention mechanism. Both the topic and candidate sentences served as input for the Bi-LSTM, while the expected outputs were classified as support, attack, or no-relation. This experiment aimed to evaluate the cross-topic generalization of relation prediction using heterogeneous sources. The in-topic F1 score achieved a value of up to 0.741, while the cross-topic F1 score reached 0.658.

Another interesting approach was done in [3], which was based on lexicons (see Figure 2.4). Claim lexicons, sentiment lexicons, emotion lexicons, and WordNet were used to expand the individual topic sentence. Additionally, four topic sentence variants were created with the help of corresponding lexicons as mentioned above. Both the extended topic sentence and the candidate sentence were fed into a Bi-LSTM and RNN respectively. The outputs of Bi-LSTM and RNN were then concatenated and

No	Example
1	**[Camping]**$_c$ [*is a great way*]$_{oc}$ to [bring]$_{oa}$ **[families]**$_a$ [together]$_{oa}$
2	**[Campers]**$_c$ [*have an opportunity*]$_{oc}$ to try some [interesting]$_{oa}$ **[food]**$_a$
3	When **[families]**$_c$ go **[camping]**$_c$, they put the **[jobs]**$_a$ and **[sporting events]**$_a$ [*on hold*]$_{oa}$
4	**[Housing]**$_c$ [*did collapse*]$_{oc}$
5	**[These countries, especially China]**$_c$, [*are taking*]$_{oc}$ **[Americans' jobs]**$_a$
6	**[We]**$_c$, [*are losing*]$_{oc}$ [our]$_a$ [*good*]$_{oa}$ **[jobs]**$_a$ so many of them
7	**[Our jobs]**$_c$, [*are fleeing*]$_{oc}$ **[the country]**$_a$
8	By putting aside these events, the **[family]**$_c$ [*has an opportunity*]$_{oc}$ to [bond]$_{oa}$ their [relationships]$_a$
9	**[Cooking]**$_c$ over a fire makes **[burgers]**$_c$ and **[potatoes]**$_c$ [*taste better*]$_{oc}$ than can be found at **[fast]**$_a$**[food]**$_a$**[place]**$_a$
10	**[Advertising]**$_c$ **[alcohol]**$_a$, **[cigarettes]**$_a$, **[goods]**$_a$ and **[services]**$_a$ with **[adult content]**$_a$ [*should be prohibited*]$_{oc}$
11	**[Modern society]**$_c$ [*needs*]$_{oc}$ **[advertising]**$_a$
12	**[Ads]**$_c$ will [*keep*] us [*well informed*]$_{oc}$ about [*new*]$_{oa}$ **[products]**$_a$ and **[services]**$_a$
13	**[advertising]**$_c$ **[cigarettes]**$_a$ and **[alcohol]**$_a$ [*will definitely affect*]$_{oc}$ our children [*in negative way*]$_{oc}$

Table 2.4 Description of four fine-grained components (C, A, OC, and OA)[6]

passed through a softmax layer to enable multi-class classification for support, attack, and no-relation. This approach led to noticeable performance improvements.

A more fine-grained approach was taken in [6], where the authors decomposed the token sequences of premises and claims into four components (See Table 2.4) namely target concepts (C), aspects (A), opinion on concepts (OC), and opinion on aspects (OA). Now, the relation between premises and claims will be determined by the relation between all those four components. This experiment was performed with three datasets namely AAE [50], AMT [58], and US2016G1TV [59] at the segment level. First, they identified C, A, OC, and OA with two methods: (1) relation extraction with manual features and (2) sequence labeling with CNN. Then, they link the premise and conclusion according to the similarity (proposition level similarity and aspect-based similarity) between A and C present inside the premises and conclusions. Once the linked premise-conclusion pairs are identified, then the relation between them is identified with the agreement between the OC and OA present inside those claims and premises via identifying constrained synonyms (CS). The results were promising with F1 scores of 0.79, 0.74, and 0.64 for AAE, AMT, and US2016 corpora respectively.

A prompt-based approach was introduced in [4], where the authors were predicting relations between premise and claim (see Figure 2.5) by predicting discourse markers from a pre-defined manually constructed set. They made this list of discourse markers with supporting or attacking differentiation. The experiment was conducted with the Reddit discussion thread dataset [60]. They masked the discourse markers from the list and fine-tune the transformer-based models like BERT to let the model know about the general representation of the argumentative discussion of Reddit. Then, they performed sequence tagging to detect the premise and claim components at the segment level. Now, once the components are there, then they designed **prompt** like this: "*Thread sequence texts before components* **USER 1 said** *Component 1* [MASK], [MASK], [MASK] **USER 2 said** *Component 2*". In these masked positions,

Paper	Features	Model	Granularity	Type	Dataset	Result
[23]	Structural, Lexical, Syntactic, Indicators, Contextual	SVM, NB, DT, RF	Segment	Classification	AAE	F1 upto 0.722
[57]	GloVe Embeddings	Bi-LSTM	Sentence	Text-Pair Classification	Araucaria and SNLI	F1 upto 0.89
[42]	Cosine Similarity	Bi-LSTM with Attention	Sentence	Text-Pair Classification	Own Cross-Domain Corpus	F1: In-Domain 0.74 and Cross-Domain 0.658
[3]	Lexicons: Claim, Emotion, Sentiment and Wordnet	Bi-LSTM	Sentence	Text-Pair Classification	Same as above	F1: Cross-Domain 0.57
[6]	Similariy and WordNet	Statistical Classifiers	Segment	Fine-grained components classification	AAE, AMT, US2016G1	F1: AAE 0.79, AMT 0.77, US2016G1 0.64
[4]	BERT Embeddings	Fine-tuned BERT	Segment	Prompt-based marker prediction	CMV Reddit	F1: Upto 0.69

Table 2.5 A summary of different early approaches used for ***Relation Type Prediction***.

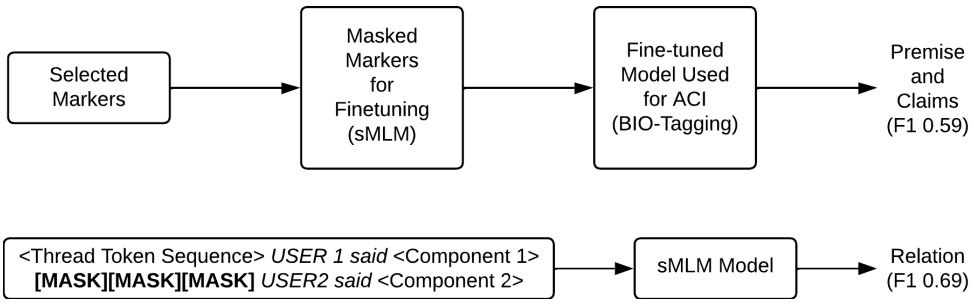

Fig. 2.5 Prompt-based approach for relation prediction [4]

the model will predict markers and that will be mapped to the previously constructed markers list to identify the relation.

With the advent of deep learning, transformer-based architectures such as BERT and RoBERTa demonstrated superior performance by learning contextual representations of argument pairs [61]. Multi-task learning frameworks further enhanced the ARC performance by jointly solving multiple argumentative tasks [62, 63]. Additionally, models incorporating external commonsense knowledge, such as ARK [64] and KE-RoBERTa [65], showed improvements by leveraging knowledge graphs like ConceptNet and WordNet. However, these approaches depend on predefined knowledge graphs, limiting their ability to generalize to unseen and implicit arguments. Recently, COMET-based approaches are introduced to generate commonsense inference chains to uncover implicit links between argumentative units [65]. While effective, these methods are constrained by the quality and coverage of pre-existing commonsense knowledge bases. Graph-based neural networks, such as DPGNN [66], have also been proposed to incorporate structural dependencies from pre-trained language models (PLMs). These methods improve reasoning but still struggle with capturing nuanced argumentative structures. Another recent method, DISARM [67], enhances relational argument classification using adversarial training and discourse marker detection. However, it lacks cross-domain evaluation and exhibits instability on smaller datasets, limiting its robustness. Despite these advances, the existing literature lacks RAM formulations that leverage the generative capabilities of LLMs. In particular, no prior work has explored how LLMs can be employed to model RAM tasks, or how their synthetic data generation abilities can be harnessed to address persistent challenges such as class imbalance.

2.2.4 Joint Modeling Approaches

Traditionally, AM research has approached its core tasks: ACI, ACC, ARI, and ARC either independently or sequentially in a pipeline manner [68, 65, 8]. Such formulations often suffer from error propagation and limited knowledge sharing, which restrict the

TH-4046_216101004

system's ability to capture the holistic nature of argumentative discourse. To mitigate these issues, a line of work has emerged that emphasizes joint modeling, where two or more subtasks are optimized within a shared framework. Early studies in this direction was explored by Stab *et al.* [12], who applied Integer Linear Programming (ILP) for joint optimization. Further, Niculae *et al.* [11] leveraged structured SVMs with factor graphs to capture interdependencies. Building upon this, neural approaches such as Pointer Networks [69] and residual networks with link-guided training [70] demonstrated the promise of deep learning in enhancing joint inference. More recent contributions employed span-based and attention-driven strategies, including span representation with LSTMs [68], task-specific parameterization with biaffine attention [71], and transition-based formulations with pre-trained language models [72]. Morio *et al.* [73] further extended this line by combining Longformer [74] with biaffine parsing in both single-task and multi-task setups, while Liu *et al.* [63] reframed AM subtasks as machine reading comprehension with a multi-hop reasoning strategy using BART.

2.2.5 End-to-End Approaches

Parallel to the joint formulations, there has been a recent growing shift toward fully end-to-end approaches that aim to unify the entire AM pipeline. A key challenge in this direction is the scarcity of annotated datasets, which motivated Morio *et al.* [73] to propose a cross-corpora multi-task approach using a span-biaffine architecture with Longformer [74], effectively modeling all four subtasks jointly. Their framework employed span classification with BIO tagging and pooling operations to construct span-level representations, thereby reducing the reliance on handcrafted pipelines. More recently, generative approaches have gained momentum, reframing AM as a structured text generation task. A generative approach was first taken in [18], by treating this task as a sequence-to-sequence generation task, focusing on the AAE [50] and CDCP [49] datasets. It involves performing all three subtasks of argument mining jointly at the same time. Given an input text, represented as $W = [w_1, w_2, ..., w_n]$, the model aimed to generate a sequence $Y = [3, 5, Claim, 10, 14, Premise, Support]$. In this sequence, tokens 3 to 5 corresponded to the *claim*, tokens 10 to 14 represented the *premise*, and the relation between them was identified as *support*. To achieve this, they employed a pointer mechanism, as discussed earlier, along with additional constraints known as the *"constrained pointer mechanism"*. Furthermore, they modified the original positional encoding of BART to suit this specific task. The results were promising, with a 0.75 F1 score achieved on the AAE dataset and 0.58 on the CDCP dataset, demonstrating successful identification of premises and claims. Building on the idea of structured prediction, the TANL framework [75] was adapted to AM by Kawarada *et al.* [76], who modeled ACC and ARC tasks within a text-to-text generation formula-

tion. Extending further, Sun *et al.* [77] integrated discourse structure-aware prefixes into BART [78], enabling the joint modeling of all four subtasks with task-specific prompts. These end-to-end generative approaches highlight a promising shift towards flexible and unified formulations, capable of capturing richer contextual dependencies compared to traditional joint models.

2.2.6 Biomedical Argument Mining

BAM has emerged as a prominent subfield of AM, driven by its potential applications in evidence-based healthcare, biomedical literature analysis, and clinical decision support. The pioneering work of Mayer *et al.* [79] introduced the first BAM dataset, annotated at the level of ACs, to facilitate automatic identification of argumentative discourse within biomedical abstracts. Building upon this, the same authors later released the `AbstRCT` dataset, which extended annotations to include both ACs and ARs, thereby enabling three central tasks in BAM: ACI, ACC, and ARI. Subsequent research has leveraged advanced contextualized word representations, such as various BERT-based models [80], to enhance performance across these tasks. Galassi *et al.* [81] proposed a multi-task learning framework that jointly optimizes AC and AR classification, effectively capturing shared argumentative structures within biomedical discourse. Similarly, the SeqMT framework [82] assumes that the ACI task is pre-resolved and focuses primarily on the ACC and ARI sub-tasks through a sequential multi-task formulation. More recent work by Liu *et al.* [83] introduced entity-aware modeling by incorporating entity co-reference and co-occurrence signals into transfer learning architectures, achieving significant improvements in ARI performance. These advances highlight the increasing integration of semantic and structural knowledge in BAM, underscoring the growing trend toward unified and context-rich representations of biomedical argumentation. Despite this progress, challenges remain concerning data sparsity, cross-domain generalization, and fine-grained relation modeling, motivating further exploration of generative and entity-enriched frameworks in subsequent chapters.

2.3 Datasets and Evaluation Metrics

2.3.1 Datasets

We conduct experiments on four widely used AM benchmarks that differ in domain, annotation granularity, and structural assumptions:

TH-4046_216101004

1. **Argument Annotated Essay (AAE)** [12]: This dataset contains 402 persuasive student essays annotated at the span level, resulting in 1,833 paragraphs. The annotation follows a tree-structured scheme, where each AC is allowed at most one outgoing AR. It defines three AC types, namely *MajorClaim*, *Claim*, and *Premise*, and comprises 6,089 ACs and 3,832 ARs in total. With approximately 147,000 tokens, AAE remains one of the most commonly used benchmarks in AM research.

2. **Fine-Grained Argument Annotated Essay (AAE-FG)** [19]: AAE-FG extends AAE by introducing a finer-grained AC taxonomy. Both *Claim* and *MajorClaim* are subdivided into *Fact*, *Value*, and *Policy*, while *Premise* is further categorized into *Common Ground*, *Testimony*, *Hypothetical Instance*, *Statistics*, *Real Example*, and *Others*. This results in nine distinct AC types, while the AR annotations remain unchanged from AAE. All numerical statistics, including the number of essays, paragraphs, ACs, and ARs, are identical to AAE.

3. **Consumer Debt Collection Practices (CDCP)** [11]: This dataset consists of 731 user comments collected from a US eRulemaking platform. Unlike AAE, it does not impose a tree constraint, allowing ACs to participate in multiple outgoing ARs. It defines five AC types, namely *Fact*, *Value*, *Policy*, *Testimony*, and *Reference*, along with two AR types, *Reason* and *Evidence*. In total, the dataset includes 4,931 ACs and 1,220 ARs, and its graph (non-tree) structure introduces additional challenges for AM models.

4. **Abstracts of Randomized Controlled Trials (AbstRCT)** [8]: This dataset comprises 700 abstracts across five disease domains: *Neoplasm*, *Glaucoma*, *Hypertension*, *Hepatitis B*, and *Diabetes*. It defines three AC types, namely *MajorClaim*, *Claim*, and *Evidence*, and three AR types: *Support*, *Attack*, and *Partial-Attack*. The *NEO-train* split contains 1,537 Evidence, 666 Claims, and 64 MajorClaims, together with 1,213 Support, 36 Attack, and 169 Partial-Attack relations. The *NEO-dev* split includes 218 Evidence, 99 Claims, and 9 MajorClaims, along with 186 Support, 8 Attack, and 25 Partial-Attack relations. The *NEO-test* split provides 438 Evidence, 228 Claims, and 20 MajorClaims paired with 364 Support, 16 Attack, and 44 Partial-Attack relations. Two additional test sets, *GLAU-test* and *MIX-test*, include 404 Evidence, 183 Claims, 7 MajorClaims and 388 Evidence, 182 Claims, 30 MajorClaims, with corresponding AR distributions of 334 Support, 7 Attack, 26 Partial-Attack and 305 Support, 3 Attack, 21 Partial-Attack, respectively. The dataset contains both tree and graph structures, making it a domain-specific benchmark for biomedical AM.

TH-4046_216101004

Together, these datasets cover both tree-structured and graph-structured argumentation settings, providing a diverse and challenging evaluation ground for the proposed frameworks.

2.3.2 Evaluation Metrics

To evaluate model performance, we use the **F1-score**, the harmonic mean of precision and recall, as the primary metric for all four AM tasks. Following prior work, we adopt *strict matching* in evaluation. A predicted AC is considered correct only if both its boundaries and type match the gold annotation exactly. Similarly, a predicted AR is correct only if its head and tail ACs, along with the relation type, exactly match the gold standard. This strict protocol ensures fair comparison with existing methods.

2.4 Summary

This chapter presented a comprehensive overview of existing research in AM, outlining its progression from early feature-based and rule-driven approaches to modern end-to-end and generative frameworks. Initial studies primarily focused on independent sub-tasks such as ACI and ARI, where pipeline-based models dominated but often suffered from error propagation and weak structural coherence. The introduction of neural architectures, particularly those employing contextual embeddings and attention mechanisms, marked a significant shift by enabling joint and multi-task formulations that captured dependencies among argumentative units. Later, end-to-end and generative models further advanced the field by unifying multiple AM tasks within a single framework, effectively addressing limitations of modular architectures. In parallel, domain-specific research in BAM has demonstrated the adaptability of AM techniques to evidence-based biomedical discourse. Studies in this area highlight the importance of contextual, semantic, and entity-level cues for capturing fine-grained argumentative relations, motivating more integrated and knowledge-enriched modeling strategies. The literature reflects a clear movement toward unified, generative, and structure-aware formulations that better capture the semantic and procedural depth of argumentation. These developments provide the foundation for the subsequent chapters, which focus on advancing end-to-end AM through autoregressive and entity-enriched generative frameworks.

TH-4046_216101004

TH-4046_216101004

Chapter 3

Exploration of Marker-Based Approaches in Argument Mining through Augmented Natural Language

Chapter Highlights

- This chapter investigates the integration of discourse and argumentative markers into a generative formulation of AM.
- **Marker-Based Fine-Tuning:** Marker distributions are introduced through auxiliary pretraining, guiding the model to recognize marker-rich contexts.
- **Marker-Enhanced argTANL (ME-argTANL):** Marker information is directly embedded into the target sequence, enabling explicit conditioning of generation on marker cues.
- Comprehensive experiments are conducted on three benchmark datasets: **AAE**, **AAE-FG**, and **CDCP**. Results demonstrate that ME-argTANL achieves strong performance as compared to the competitive baselines.
- The publication related to this chapter is as follows:
  1. **Nilmadhab Das**, Vishal Choudhary, V. Vijaya Saradhi, Ashish Anand, "Exploration of Marker-Based Approaches in Argument Mining through Augmented Natural Language," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Italy, 2025.

3.1 Introduction

While the earlier chapters introduced the fundamental concepts and challenges in AM, this chapter presents a unified methodology that addresses end-to-end argument mining

TH-4046_216101004

in a streamlined unified manner. Traditionally, AM is decomposed into four subtasks [11]: ACI, ACC, ARI, and ARC, as mentioned in the previous chapters. In the current chapter, however, we considered only ACC and ARC. This choice was intentional. First, solving ACC implicitly resolves ACI, since identifying the type of a component requires simultaneously locating its span. Likewise, resolving ARC implicitly covers ARI, as determining the type of relation between two components inherently requires identifying whether a relation exists. Building on this rationale and following the same formulation of [76], this chapter adopts a grouped view of the subtasks. We denote the combined responsibilities of ACI and ACC as Argument Component Extraction (ACE), and the combined responsibilities of ARI and ARC as Argumentative Relation Classification (ARC). Consequently, end-to-end AM, termed ACRE, entails jointly solving both ACE and ARC in a single unified framework. However, in the subsequent chapters, we revisit the four sub-tasks individually.

In the recent times, AM contributes to a variety of downstream applications such as debate analysis [84], automated essay scoring [85], and customer review analysis [86]. However, the primary challenge in AM lies in effectively handling the longer sequence length of ACs and their associated ARs [7]. Defining boundaries for ACs is more intricate compared to tasks like Named Entity Recognition (NER) or Parts-of-Speech (POS) tagging, where the target text span consists of a few tokens only. Also, every AC has certain underlying contexts of argumentativeness and is related to another AC of the same context. Variations in argument representations across domains pose another challenge [87]. Previous approaches to this task have primarily utilized extractive methods [18] to address these challenges, which have shown limited performance. Recently, end-to-end frameworks based on dependency parsing [17] have gained popularity due to their ability to streamline the extraction process into a single unified model. However, these methods often require intricate post-processing steps, adding to the overall complexity of the solution. Given the limitations of these recent approaches, we propose an alternative method based on a text-to-text generation paradigm. This new approach reduces complexity by eliminating the need for elaborate post-processing.

Several generative frameworks have recently been proposed for various NLP tasks, including NER [88], joint entity and relation extraction [89], and co-reference resolution [90]. One such framework is Translation between Augmented Natural Language (TANL)[91], which addresses a range of NLP tasks within a unified framework. This framework generates outputs by augmenting the original input text with class labels specific to downstream tasks. "*[Tolkien | **person**]'s epic novel [The Lord of the Rings | **book**] was published in 1954-1955.*" Here, labels such as *person* and *book* are embedded within the original text. With this kind of output format, complex post-processing steps are reduced. However, among all the NLP tasks solved by the

TH-4046_216101004

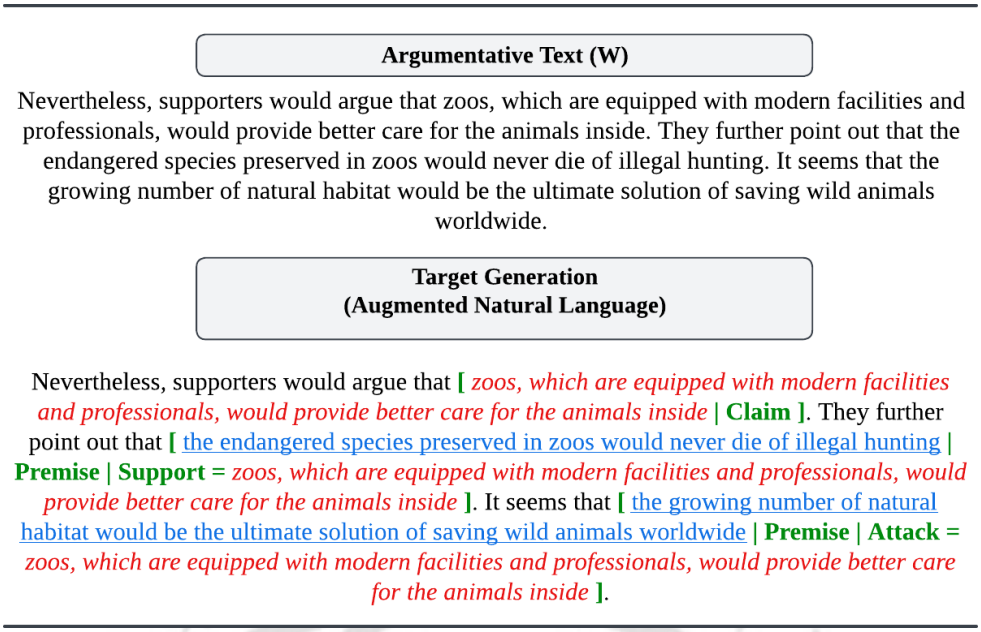

Fig. 3.1 An overview of the proposed generative end-to-end argument mining framework ***arg*TANL**. Claims are highlighted in *italics* and marked in red. Premises are underlined and marked in blue. Augmented labels are represented with **bold** font and marked in green.

TANL framework, the target output spans (e.g., *Tolkien* or *The Lord of the Rings*) are relatively shorter as compared to the target spans of AM tasks, where each AC span is distributed across extended sections of text. Thus, it is a meaningful direction to explore the effectiveness of TANL in solving AM tasks that consist of longer target spans.

Studies in [92] have demonstrated that markers are strongly correlated with improvements in AM task performance as they reliably indicate the presence of ACs in the text. Two distinct types of markers are categorized: *(a) Argumentative Markers* and *(b) Discourse Markers (DMs)*. Argumentative Markers are typically phrases such as *"I strongly agree that," "But, I deny the point that," or "However, this clearly proves that,"* conveying the argumentativeness of the discourse. In contrast, DMs are single-token connectives like *"But," "And," or "However,"* representing the rhetorical structure of language. The set of argumentative markers is extracted using the rule-based methods by [14], while the set of DMs is constructed from a specific web-corpus by [93]. However, rule-based extraction yields a large number of irrelevant spans lacking generalizability. We employ a similar extraction method for *Argumentative Markers*, incorporating an additional manual filtering step to eliminate irrelevant or non-generic spans.

We propose the following. (1) ***arg*TANL**, a TANL framework for the downstream task of computational AM. In this, both ACs and ARs are label-augmented to form an ANL-based generation sequence (See Figure 3.1). This framework handles longer

TH-4046_216101004

target AC spans than the target spans of previously attempted downstream tasks. (2) **Marker-Based Fine-Tuning of *arg*TANL** to enhance its performance. From the different marker sets *(e.g., Argumentative Markers and DMs)*, the marker information is learned first with specifically designed fine-tuning strategies. Then with this learned marker information, the ***arg*TANL** output is learned. (3) **Marker Enhanced *arg*TANL (*ME-arg*TANL)**, an enhanced *arg*TANL framework by incorporating the marker knowledge inside the output text itself. This is distinctly different from the TANL framework, where in addition to the class labels of ACs and ARs, the marker information is also included in the output. *ME-arg*TANL allows to perform two tasks jointly: (a) Marker Identification and (b) AC and AR identification. With this framework, the model receives a strong signal from the markers, which are present just before the AC, to guide the identification of that AC inside the same generation sequence. This, in turn, enhances the overall performance of the AM task. Whenever the marker information is present inside the AC span the *ME-arg*TANL framework is redundant, In that case, the Marker-Based Fine-Tuning framework is appropriate.

We conduct a comprehensive set of experiments to evaluate the performance of the proposed framework, organized into several subsections. We first analyze the main results with a detailed comparison of *arg*TANL, Marker-Based Fine-Tuned *arg*TANL, and *ME-arg*TANL to highlight the improvements brought by the knowledge of markers. Then we compare the *Component-Only and End-to-End* variants to determine their relative effectiveness. Next, we analyze the *Impact of input length* on the performance of the *arg*TANL. Then, we conduct an ablation study by introducing **Abbreviated *arg*TANL**. Here, repeated full AC spans are replaced with unique "ID" tokens, reducing output sequence length and improving efficiency in handling longer paragraphs. This study explores the performance trade-offs between the standard and abbreviated formulations. Finally, an *Error Analysis* is performed to find the scope for further improvement. Through these extensive experiments upon three standard benchmarks of AM literature, the proposed approach achieves competitive results in several datasets and surpasses various important baselines. In summary, the main contributions of this chapter are:

1. A generative task formulation in the form of ***arg*TANL** framework to solve *End-to-End AM* along with *Component-only* variant.

2. We enhance *arg*TANL with a specialized ***Marker-Based Fine-Tuning*** by leveraging the knowledge of markers to improve the AM task performance.

3. We propose ***ME-arg*TANL**, an advanced version of *arg*TANL that integrates marker information directly into the output sequence. It identifies markers and argumentative structures jointly to deliver superior AM task performance.

4. An extensive performance analysis along with a target-oriented ablation study with **Abbreviated *arg*TANL** to justify the strength of the proposed formulation.

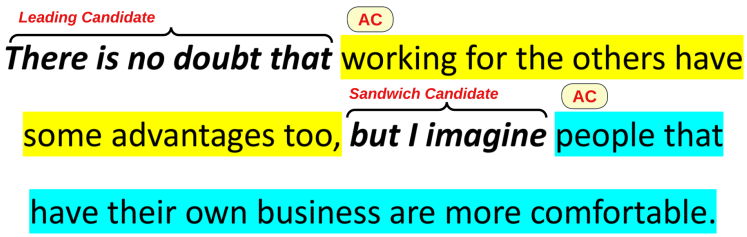

Fig. 3.2 An example sentence from AAE corpus describing both ways of marker extraction. Here, extracted ***marker candidates*** are in bold italics, and ACs are highlighted.

3.2 Task Formulation

We represent argumentative text as $W = w_1, w_2, w_3, ..., w_n$, where n is the total number of tokens in W. For a text-span $w_i, w_{i+1}, w_{i+2}, \ldots, w_j$ in W, we write it as $w_{i:j}$. We define a set of AC types as $T^c = \{t_1^c, t_2^c, t_3^c, ..., t_{n_c}^c\}$ and a set of AR types as $T^r = \{t_1^r, t_2^r, t_3^r, ..., t_{n_r}^r\}$, where n_c and n_r refer to a total number of possible AC and AR types respectively. We use a set of symbol tokens, $S = \{[,], =, |\}$ in the ANL, where "[" marks the start of a component, "]" marks the end of a component, " = " is used for relation assignments, and "|" is used to separate labels. Subsequent sections discuss different formulations of the proposed task.

3.2.1 ACE task: Component-only Variant

For any given argumentative text $w_{1:n}$, objective of the *ACE* is to extract a set of ACs as $C = \{C_i | C_i = (c_i, c_i^s, c_i^e)\}$, where C_i is the i^{th} AC, $c_i \in T^c$, and c_i^s and c_i^e refer to the relative start and end token indices of c_i respective to W. Here, we generate the label-augmented text for ACs only such as, in Figure 3.1, for the head, AC is *[the endangered species preserved in zoos would never die of illegal hunting | Premise]* and for tail AC is *[zoos, which are equipped with modern facilities and professionals, would provide better care for the animals inside | Claim]*. The main motivation for this variant is to assess our proposed method's ability to identify ACs alone when the relational information is muted.

3.2.2 ACRE task: End-To-End Argument Mining

The proposed end-to-end formulation jointly frames the *ACE* and *ARC* tasks in the following manner. We define ARs as $R = \{R_i | R_i = (r_i, r_i^h, r_i^t)\}$, where R_i is the i^{th} AR corresponding to AR type $r_i \in T^r$, and r_i^h and r_i^t refer to the head and tail ACs respectively. r_i^h and r_i^t are connected with relation type r_i. If two components $C_p, C_q \in$

C are related with $R_k \in R$ as $r_k^h = C_p$ and $r_k^t = C_q$, then for the token spans of C_p i.e. $w_{c_p^s:c_p^e}$ and C_q i.e. $w_{c_q^s:c_q^e}$, the model will generate augmented labels as $[w_{c_p^s:c_p^e}|c_p|r_k = w_{c_q^s:c_q^e}]$ and $[w_{c_q^s:c_q^e}|c_q]$ respectively. The rest of the tokens in W will be rewritten as it is. We refer to this joint formulation as *ACRE*. Figure 3.1 illustrates the *ACRE* formulation.

3.3 Methodology

In this section, we provide an overview of the *arg*TANL framework, the marker extraction process, the marker-based fine-tuning approach, and the *ME-arg*TANL framework.

3.3.1 The *arg*TANL Framework

This framework adopts the TANL formulation (Figure 3.1) with specific customization tailored for the AM task. Given argumentative text as input, the framework generates ANL consisting of ACs and ARs as label-augmented texts enclosed with symbol tokens, as described in Section 3.2. This approach leverages the strengths of the TANL framework while introducing modifications to effectively capture the complex relationships and structures inherent in argumentative texts. The integration of symbolic tokens ensures precise boundary definition and clear distinction between different ACs and ARs, facilitating accurate identification within the generated sequences.

3.3.2 Argumentative Marker Extraction

An argumentative marker typically signals the beginning of an AC. However, not every marker is always followed by an AC, and an AC may not always be preceded by a marker. Considering this phenomenon, we extract two types of (See Figure 3.2) potential marker candidates from any argumentative text: *(i) Leading candidates:* by extracting tokens from the start of a sentence to the beginning of an AC, and *(ii) Sandwich candidates:* by extracting tokens after the end of an AC until the start of another AC in the same sentence. However, these strategies yielded some spans, which were non-generic. For example, "*In spite of the importance of* ***sports activities***" or "*Nevertheless, opponents of* ***online-degrees*** *would argue that*" are having non-generic spans. These spans lacked generalizability across diverse topics, being tailored solely to their respective subjects. To enhance generalizability, we introduce an additional layer of manual filtering to eliminate topic-dependent spans from the pool of extracted marker candidates. Initially, we identified 2925 marker candidates. After manual

Table 3.1 Description of the initial step of ***Marker-Based Fine-Tuning*** of the *arg*TANL framework.

Strategy	Input Sequence	Target Generation Sequence
A-MKT	Last but not least, students have ... difficulties.	**[Last but not least,** | **marker]** students have ... difficulties.
SM-MKT	<*extra_id_0*> <*extra_id_1*> <*extra_id_2*> <*extra_id_3*> <*extra_id_4*> students have ... difficulties.	<*extra_id_0*> **Last** <*extra_id_1*> **but** <*extra_id_2*> **not** <*extra_id_3*> **least** <*extra_id_4*> **,** <*extra_id_5*>
E-MKT	Last but not least, students have ... difficulties.	[**-1,-1,-1,-1,-1,**0,0,0,0,0,0,0]
D-MKT	Motivations for playing cricket are vastly different. It is a well-crafted game.	Motivations for playing cricket are vastly different. **Truly,** it is a well-crafted game.

filtering and removing duplicates, we refine the list to 1072 unique *argumentative markers*.

3.3.3 Marker-Based Fine-Tuning of *arg*TANL

This technique strategically divides the *arg*TANL output generation into two significant steps. The initial step leverages the extracted markers from various corpora to execute Marker-Based Fine-Tuning. This involves implementing four distinct generative fine-tuning strategies, each utilizing varied input and output combinations (see Table 3.1). The objective is to acquaint the model with the nuanced representations of markers within the argumentative text. Notably, neither the markers from the test data nor the test data itself were used during this initial fine-tuning process.

In the second step, the models derived from the initial step undergo additional fine-tuning to perform the *arg*TANL task. This way, we get four unique resultant models, each distinguished by the applied fine-tuning strategy. Below are the details of different Marker-Based Fine-Tuning strategies performed as the first step of this technique:

3.3.3.1 Augmented Marker Knowledge Transfer (A-MKT)

This strategy takes plain text input and fine-tunes the model to generate ANL where only *markers* are augmented. ACs and ARs are not augmented.

3.3.3.2 Span-Masked Marker Knowledge Transfer (SM-MKT)

It is a self-supervised denoising fine-tuning strategy by *masking the span of markers*. We replace every token in the marker span with sentinel tokens. Here, the target generation sequence is formed by concatenating the sentinel tokens and the corresponding marker tokens.

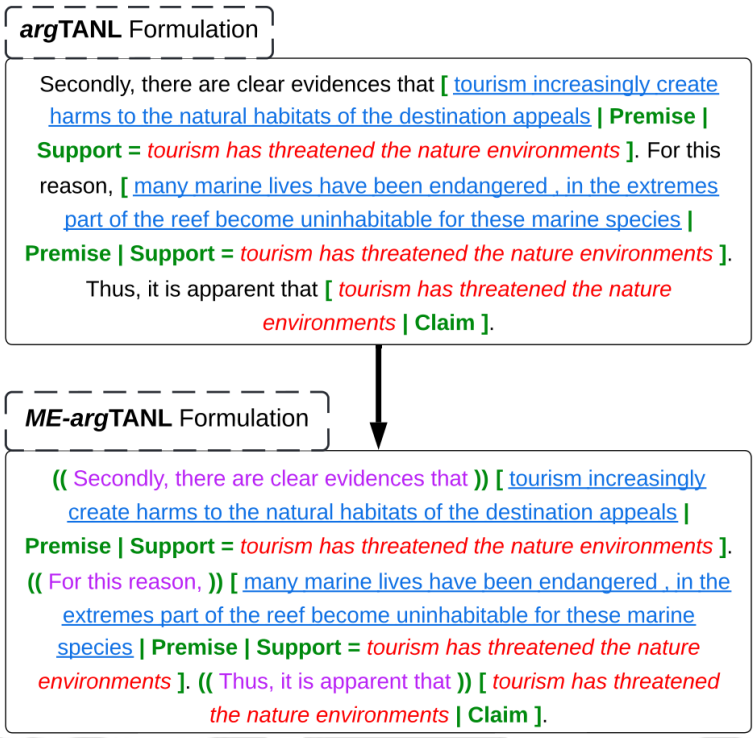

Fig. 3.3 Description of ***ME-arg*TANL** Formulation. Markers are enclosed with the "(( ))" symbol in violet color. Claims are highlighted in red *italics*. Premises are underlined in blue. Augmented labels are represented in green with **bold** font.

3.3.3.3 Marker Knowledge Transfer through Encoding (E-MKT)

Unlike the above strategies, which are based upon text-to-text generation, it is a *text-to-sequence* generation strategy. Here, we generate the labels of marker tokens in terms of a numeric sequence of 0's and -1's, where -1 and 0 are replacing the markers and non-markers of the input text, respectively.

3.3.3.4 Discourse Marker Knowledge Transfer (D-MKT)

To check the effectiveness of single-token DMs over multiple-token markers, we propose this fine-tuning strategy. Using sentence pairs from a standard DM corpus, we generate a target sequence in the following format: *(Sentence 1 + DM + Sentence 2)*, where the concatenated input is *(Sentence 1 + Sentence 2)*. Thus the model can achieve the capability of generating the probable *"connective"* between a pair of sentences.

3.3.4 The *ME-arg*TANL Framework

This framework modifies the *arg*TANL to improve the performance of AM tasks by uniquely embedding marker knowledge into the same generation sequence. We include

the marker span using *"(( ))"* to form an unique marker span as *"((Marker))"* (See Figure 3.3). These double braces serve as a special enclosing symbol, indicating the task difference between the *arg*TANL task, and the task of marker identification. Given that marker identification occurs within the same target sequence and the generation process is autoregressive, with markers appearing mostly on the left side of the target AC span, there is a strong likelihood that the augmented markers will be generated first. Subsequently, these markers will serve as a guiding signal for the generation of ACs and ARs, which are present immediately afterward. This type of sequential generation will enrich the model by engaging it in two types of structurally independent tasks simultaneously.

3.4 Experimental Setup

3.4.1 Datasets

We consider these three datasets: **AAE**, **AAE-FG** and **CDCP** to evaluate the proposed method.

Alongside, we use the `Discovery` [93] corpus as the primary source of DMs. It is used for the sole purpose of the initial fine-tuning of the *Marker-Based Fine-Tuning* only for its D-MKT strategy. Extracted from the *DepCC* web corpus [94], it features 1.74 million pairs of adjacent sentences *(Sen1, Sen2)* with 174 DMs, consolidating 10k pairs per DM. All DMs occur at the beginning of *Sen2*.

Notably, the *arg*TANL framework and the *Marker-Based Fine-Tuning* of *arg*TANL are designed to be applicable across all datasets. Whereas the *ME-arg*TANL framework is only designed for both versions of the AAE corpus as this is the only dataset that contains markers outside the component boundary. The CDCP dataset includes the DMs inside the AC span boundaries. So, the formation of the *ME-arg*TANL output is not trivially possible for this dataset.

3.4.2 Training Details

All experiments were performed using the *T5-Base* model [95] on Nvidia A100 GPUs, and each reported score is obtained by averaging results over 5 independent runs. The hyperparameter settings for the *arg*TANL framework are as follows: For the CDCP dataset, a batch size of 4 is used, while for AAE and AAE-FG, the batch size is set to 8. Across all datasets, we utilize the AdamW optimizer with a learning rate of 0.0005. The maximum input length is set to 1024 tokens for CDCP and 512 tokens for AAE

and AAE-FG. In the end-to-end and component-only training scenarios, we run for more than 10000 steps, with checkpoints saved every 200 steps. During inference, beam search is used with a beam length of 8 across all datasets. For all strategies (A-MKT, SM-MKT, D-MKT, and E-MKT) of Marker-Based Fine-Tuning, we use a batch size of 16, the AdamW optimizer with a learning rate of 0.0005, a maximum input length of 512 tokens, and 10 epochs of training. We use the following libraries: (i) TANL framework[1], and (ii) HuggingFace's Transformers[2].

3.4.3 Baseline Methods

We consider several important baselines to investigate the efficacy of our end-to-end AM formulation on each dataset as follows:

1. **ILP** [96]: Rich feature based approach to perform *joint inference* over the AM sub-tasks optimized by *Integer Linear Programming (ILP)*.
2. **BLCC** [97]: Based upon *Bi-LSTM-CNN-CRF (BLCC)* to formulate this task as a sequence tagging problem.
3. **LSTM-ER** [97]: An adapted version of an end-to-end relation extraction model with sequential LSTM [98].
4. **LSTM-Parser** [97]: A dependency parsing approach built on stacked LSTM [99].
5. **BiPAM** [100]: Another dependency parsing approach with customized biaffine operation based upon BERT-base [101].
6. **BiPAM-syn** [100]: An enhanced version of BiPAM with the inclusion of *syntactic* information.
7. **BART-B** [18]: A generative approach to *text-to-sequence generation* with Bidirectional and Auto-Regressive Transformer (BART) [78].
8. **RPE-CPM** [18]: An extended version of BART-B that introduces a modified positional encoding scheme together with a pointer-based decoding constraint.
9. **Span-Biaffine-ST** [17]: Latest dependency parsing approach based upon Longformer [74] along with *Span-Biaffine* architecture for information sharing among sub-tasks. Since our approach is based upon a *single-task setup (ST)*, we take the ST version of this work.
10. **SP-AM** [76]: Adapted TANL framework for *Structured Prediction* for end-to-end AM based upon text-to-text generation.
11. **DENIM** [77]: A discourse structure aware generative framework with multitask prompt-tuning.

[1]https://github.com/amazon-science/tanl
[2]https://github.com/huggingface

For the AAE benchmark, we evaluate the proposed method against all these baselines. For the AAE-FG dataset, we benchmark our results by running the most competitive open-source baseline **Span-Biaffine-ST**, using our implementation. For the CDCP benchmark, we use the following baselines: **BiPAM**, **BART-B**, **RPE-CPM**, **CPM-only** (without RPE)[18], **SP-AM**, and **Span-Biaffine-ST**.

3.4.4 Performance Metrics for Evaluation

After generating the target ANL, we post-process the sequence by removing the symbol tokens to get a cleaned text. Finally, for a comprehensive evaluation, AC and AR tuples are created, including their types and corresponding boundaries. Following [97], we evaluate the results with *micro-F1* score for both ACE and ACRE tasks, where an exact match with the gold label is considered as a true label.

3.5 Results and Discussion

In our discussion, we refer to the ***arg*TANL** framework as $T5_{argTANL}$, ***ME-arg*TANL** as $T5_{ME-argTANL}$, and the ***Marker-Based Fine-Tuning*** as $T5_{MFS}$. Here, MFS indicates a specific Marker-Based Fine-Tuning Strategy: A-MKT, E-MKT, SM-MKT, or D-MKT.

Table 3.2 compares the performance of the proposed frameworks with the baselines on the ACRE task in terms of micro F1-scores for component classification and relation classification. For the AAE dataset, $T5_{ME-argTANL}$ turns out to be the best among the proposed frameworks, surpassing *arg*TANL and all the *MFS-based* strategies, for component classification. While $T5_{ME-argTANL}$ outperforms several baselines for component and relation classifications, it gave a competitive performance to the current state-of-the-art models for component classification. We observe a similar performance on the AAE-FG dataset. For the AAE-FG dataset, $T5_{ME-argTANL}$ surpasses the current state-of-the-art result for component classification.

The likely reason behind the relatively poor performance of *MFS-based* strategies is that models can detect markers within and outside the AC spans. However, in the second step of *arg*TANL output generation, containing ACs and ARs, there is a high probability that some ACs might get split into smaller segments due to the presence of marker-like tokens within the AC spans. This erroneous splitting increases the number of potential permutations of ARs with these split ACs. Conversely, in the $T5_{ME-argTANL}$ model, both the markers and AC/AR labels are generated within the same target sequence. This allows the model to understand markers' relative positions and corresponding ACs. These markers act as contextual guides with the immediately preceding markers to automatically exclude markers inside the AC spans. This

Table 3.2 Experiment results on ACRE task with comparable baselines. Best scores are marked in **bold**. * indicates the baseline results we produced by running their open-source code.

Corpus	Model	Params	Micro-F1	
			ACE	ARC
CDCP	BiPAM	110M	41.15	10.34
	BART-B	139M	56.15	13.76
	RPE-CPM	139M	57.72	16.57
	CPM-only	139M	58.13	15.11
	Span-Biaffine-ST	149M	**68.90**	**31.94**
	SP-AM *(Flan-T5-Base)*	220M	66.80	23.19
	$T5_{argTANL}$ **(Ours)**	220M	66.56	19.81
	$T5_{E-MKT}$ **(Ours)**	220M	57.31	7.97
	$T5_{A-MKT}$ **(Ours)**	220M	61.06	13.86
	$T5_{SM-MKT}$ **(Ours)**	220M	64.17	16.68
	$T5_{D-MKT}$ **(Ours)**	220M	67.49	20.68
AAE-FG	Span-Biaffine-ST*	149M	59.54	**54.75**
	$T5_{argTANL}$ **(Ours)**	220M	60.41	32.14
	$T5_{E-MKT}$ **(Ours)**	220M	55.24	25.83
	$T5_{A-MKT}$ **(Ours)**	220M	57.66	27.82
	$T5_{SM-MKT}$ **(Ours)**	220M	59.37	30.54
	$T5_{D-MKT}$ **(Ours)**	220M	59.71	31.03
	$T5_{ME-argTANL}$ **(Ours)**	220M	**61.39**	33.12
AAE	LSTM-Parser	-	58.86	35.63
	ILP	-	62.61	34.74
	BLCC	-	66.69	39.83
	LSTM-ER	-	70.83	45.52
	BiPAM	110M	72.90	45.90
	BiPAM-Syn	110M	73.50	46.40
	BART-B	139M	73.61	47.93
	RPE-CPM	139M	75.94	50.08
	SP-AM *(Flan-T5-Base)*	220M	75.55	58.51
	DENIM	139M	76.50	58.51
	Span-Biaffine-ST	149M	76.48	**59.55**
	$T5_{argTANL}$ **(Ours)**	220M	75.93	50.56
	$T5_{E-MKT}$ **(Ours)**	220M	73.06	45.89
	$T5_{A-MKT}$ **(Ours)**	220M	74.22	48.01
	$T5_{SM-MKT}$ **(Ours)**	220M	75.91	49.08
	$T5_{D-MKT}$ **(Ours)**	220M	76.45	49.91
	$T5_{ME-argTANL}$ **(Ours)**	220M	**77.14**	52.12

TH-4046_216101004

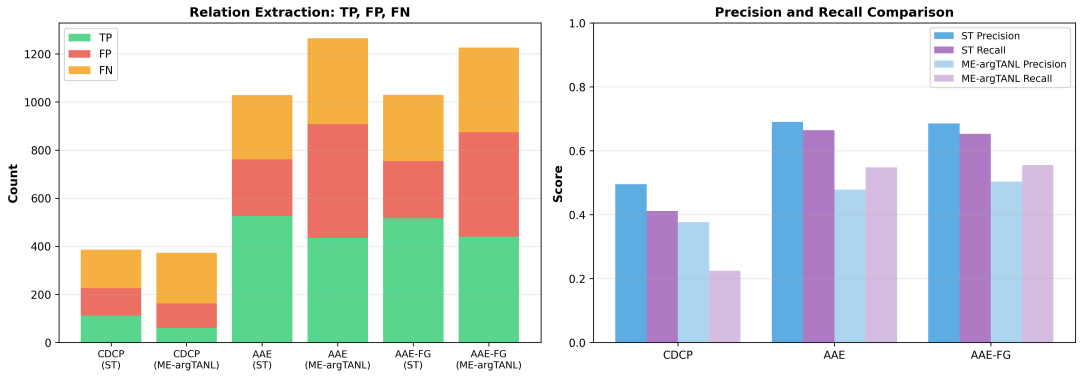

Fig. 3.4 Comparison of TP, FP, and FN counts along with Precision and Recall scores between the ST baseline and the proposed *ME-arg*TANL framework across the CDCP, AAE, and AAE-FG datasets.

underscores the efficiency of the ***ME-arg*TANL** by directly including argumentative markers within the target generation process, along with the augmented ACs and ARs, as opposed to employing any of *MFS-based* technique.

D-MKT produces the best result for the CDCP dataset among the *arg*TANL frameworks. Though it could outperform most of the baselines, it gave the second-best performance in terms of micro-F1 score for the component classification. It shows the strength of using the ***MFS-based*** approach, where the model benefits from the contextual cues provided by DMs for identifying ACs. Across all the datasets, none of the proposed frameworks could beat the state-of-the-art result on relation classification within the joint formulation.

The above results indicate that the specific nature of argumentative markers, which are highly beneficial in AAE dataset, might not align as effectively with the argumentative structures present in the CDCP dataset. The A-MKT, SM-MKT, and E-MKT strategies are likely affected by *catastrophic forgetting* [102], a phenomenon where the model partially forgets previously learned knowledge during the target task fine-tuning. The poor performance of E-MKT across all datasets indicates that the knowledge of the relative position of markers is not beneficial. Similarly, the A-MKT strategy, which involves similar tasks set up in both steps of fine-tuning, proves to be sub-optimal. In contrast, the span-masking-based strategy SM-MKT demonstrates superior performance compared to both E-MKT and A-MKT.

3.5.1 Detailed Failure Analysis of the ARC Task

Our marker-enhanced *arg*TANL models perform well on the ACC task. However, they fall short of the strong Span-Biaffine Single-Task (ST) baseline (the best one among the compared baselines) on the ARC task across several datasets. To understand why,

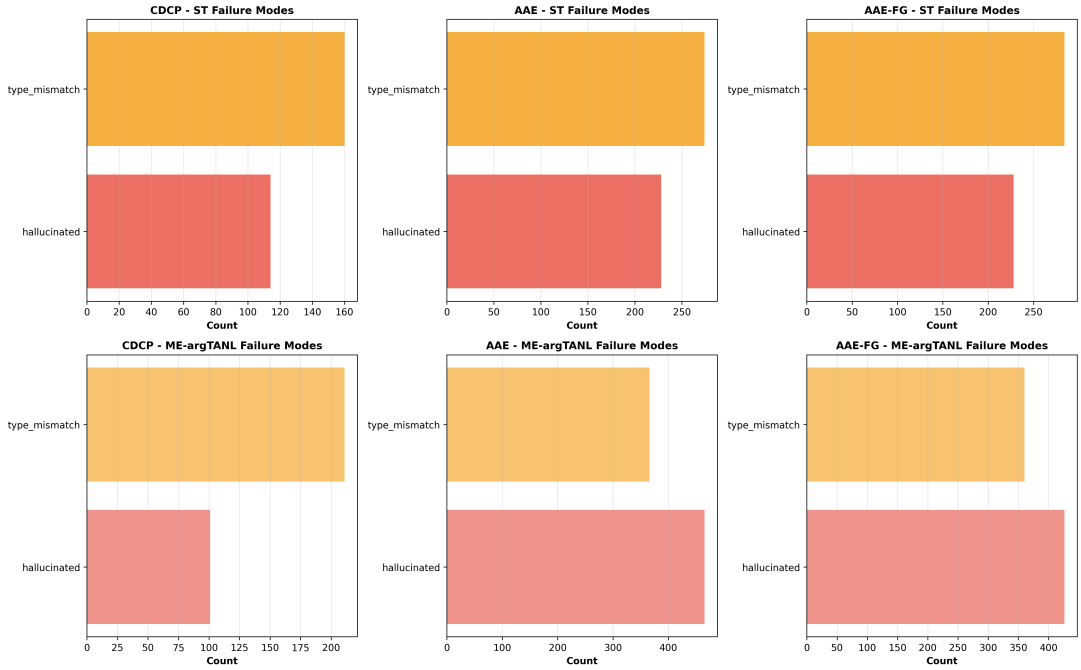

Fig. 3.5 Distribution of dominant failure modes across the CDCP, AAE, and AAE-FG datasets for both ST and *ME-arg*TANL frameworks. The analysis highlights hallucinated relations and relation-type mismatch errors.

we looked closely at the ARC results. We examined the True Positives (TP), False Positives (FP), and False Negatives (FN). We also analyzed the most common types of classification errors. Figures 3.4 and 3.5 compare our TANL framework with the ST baseline on the CDCP, AAE, and AAE-FG datasets.

Figure 3.4 shows the TP, FP, and FN counts, along with Precision and Recall scores for the ARC task. Compared to the ST baseline, the *ME-arg*TANL framework creates more false positives. At the same time, it misses several real argumentative relations. The generative model sometimes boosts recall by guessing extra relations. However, the high number of false positives heavily hurts precision. This leads to a lower overall score on the ARC task. This issue stands out clearly in the CDCP dataset. Here, our framework predicts relations that make logical sense but are structurally wrong. The *ME-arg*TANL model generates the entire argument structure step-by-step. Because of this, it often connects related ACs even when no actual relation exists in the data. This creates many “hallucinated" or fake relations. These hallucinations directly lower the ARC Micro-F1 score compared to the ST baseline.

Figure 3.5 takes a closer look at these errors. Across all datasets, we see two main types of mistakes:

1. **Hallucinated Relations:**

   The model predicts relations that do not exist in the actual data. This happens often with *ME-arg*TANL models. The decoder tries to build the whole structure based on word meanings learned during training. As a result, it frequently

connects topics that seem similar, even without a real link. These fake relations increase false positives and hurt precision.

2. **Relation Type Mismatches:**

   The model finds the right pair of ACs, but guesses the wrong relation type. For instance, it might label a *Support* relation as an *Attack*. This happens a lot in the CDCP dataset. In CDCP, arguments are often implied, and clue words are weak or unclear. This shows that simple marker cues are not enough. We need deeper meaning to correctly tell relation types apart.

The AAE and AAE-FG datasets contain more explicit argumentative structures than CDCP. Their discourse markers align better with relation boundaries, which reduces the performance gap between *ME-arg*TANL and ST. However, ST still performs better on the ARC task because it models pairwise relations more directly under stronger structural constraints. In contrast, the autoregressive decoding process of *ME-arg*TANL is more prone to hallucinated relations, relation-type confusion, and error propagation. Overall, this analysis shows that marker-based cues alone are insufficient for robust relation classification, especially in cases involving implicit reasoning and long-range dependencies. These findings motivated the later chapters of this thesis, where semantically grounded key phrases and entity-aware representations are introduced to improve relational reasoning.

3.5.2 Component Only vs End-to-End Variants

Table 3.3 compares the performance of models on the ACE task in the two formulations. Analyzing the argumentative structure of AAE and AAE-FG datasets reveals that argument components always form a tree, indicating a single parent for each child. In this well-structured dataset, the performance of the models drops in the *Component-only* variant compared to the *end-to-end* variant. The CDCP dataset contains many isolated components. It does not necessarily form a tree but can form a graph (e.g., $t_1^c \rightarrow t_2^c$, $t_2^c \rightarrow t_3^c$, and a transitive relation $t_1^c \rightarrow t_3^c$). We observe the similar performance of models, except E-MKT, in the two variants for CDCP corpus. This result indicates that ACs and ARs benefit from mutual feature information, suggesting synergy in an end-to-end setup. For *Component-only* variant, the D-MKT strategy proves to be beneficial for all the datasets. When comparing the compatibility of the E-MKT with other *MFS* strategies, it appears more suitable for the Component-only variant rather than the end-to-end for the ACE task. It exhibits the minimum performance difference in the Component-only setup for both versions of the AAE corpus and even shows performance improvements for the CDCP corpus.

Table 3.3 Performance difference on the ACE task: Component-Only vs. End-to-End variants. Among all techniques, minimum differences between these two variants are marked in **bold**.

Corpus	Model	Micro-F1 of ACE		Diff. (±)
		Comp-Only	End-to-End	
AAE	$T5_{argTANL}$	74.12	75.93	-1.81
	$T5_{ME-argTANL}$	74.48	77.14	-2.66
	$T5_{E-MKT}$	71.64	73.06	**-1.42**
	$T5_{A-MKT}$	72.20	74.22	-2.62
	$T5_{SM-MKT}$	73.63	75.91	-2.28
	$T5_{D-MKT}$	**74.87**	76.45	-1.58
AAE-FG	$T5_{argTANL}$	59.20	60.41	-1.21
	$T5_{ME-argTANL}$	58.90	61.39	-2.49
	$T5_{E-MKT}$	55.15	55.24	**-0.09**
	$T5_{A-MKT}$	56.08	57.66	-1.58
	$T5_{SM-MKT}$	57.32	59.37	-2.05
	$T5_{D-MKT}$	**59.29**	59.71	-0.42
CDCP	$T5_{argTANL}$	66.40	66.56	-0.16
	$T5_{E-MKT}$	59.05	57.31	**+1.74**
	$T5_{A-MKT}$	60.43	61.06	-0.63
	$T5_{SM-MKT}$	62.76	64.17	-1.41
	$T5_{D-MKT}$	**66.66**	67.49	-0.83

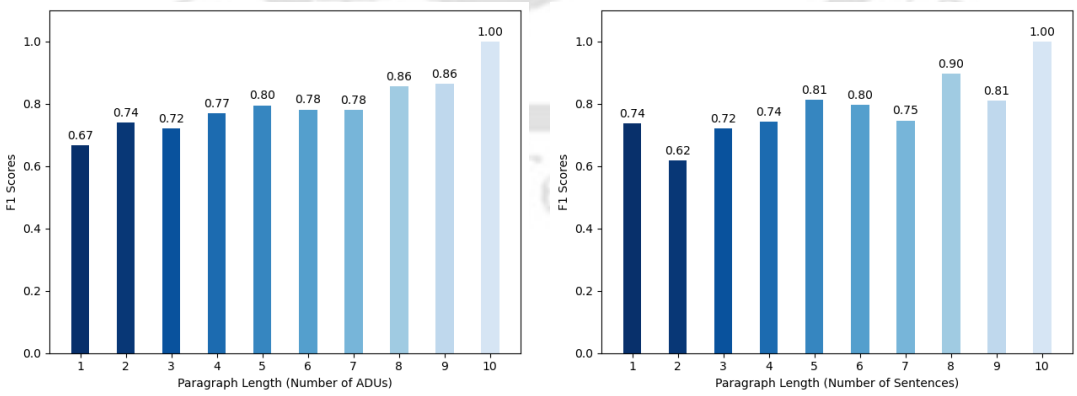

Fig. 3.6 Performance on the ACE task of the End-to-End variant with the varying number of ADUs (left) and of sentences (right) in a paragraph.

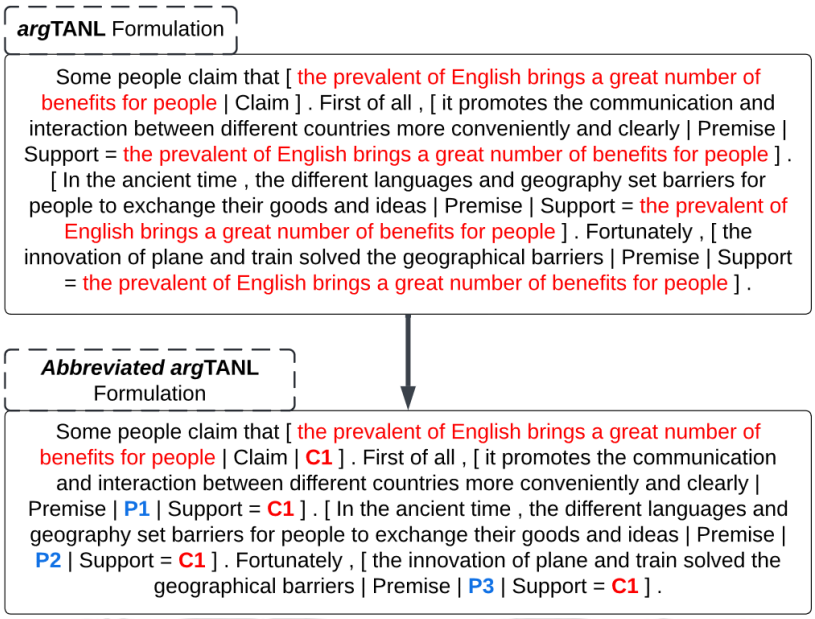

Fig. 3.7 Representation of **Abbreviated *arg*TANL** with *"ID"* tokens replacing repeating spans for a shorter ANL output format.

3.5.3 Performance Analysis Based on the Length of the Input

We assess the performance of *arg*TANL on the ACRE task for extracting ACs for the AAE dataset. Our analysis considers both the number of input text sentences and the number of ADUs. As illustrated in Figure 3.4, the performance generally improves with increasing input length. However, the trend is more consistently monotonic when paragraph length is measured in terms of ADUs compared to sentences. This suggests that argumentative density, represented through ADUs, is more indicative of argTANL performance than sentence count alone.

3.5.4 Ablation with the Abbreviated *arg*TANL framework

This representation (See Figure 3.7) is designed to minimize redundant and repeated AC spans connected to the right side of the *Relation-type*. Here, we add an extra *"ID"* token to every unique AC present in a paragraph in an ascending manner. For example, if *n* number of ACs of type *Premise* and *m* number of *Claims* are present in a paragraph, then, we will create abbreviated labels in the form of *"ID"* tokens as *P1, P2, P3...* for all *n* number of *Premises* and *C1, C2, C3...* for all *m* number of *Claims*. These tokens will then be appended just after the original AC label as a *"unique ID"* of the original label. Later, whenever the same AC is repeated during the relation assignment with *Relation-type*, only the abbreviated tokens will be used for the same. This way we are keeping the related AC connections intact while reducing the size of

Table 3.4 Performance comparison between *argTANL* and *Abbreviated argTANL* formulations.

	Micro-F1 of ACE			Micro-F1 of ARC		
Dataset	***arg*TANL**	**Abbr.**	**Diff.**	***arg*TANL**	**Abbr.**	**Diff.**
AAE	75.93	75.48	-0.45	50.56	47.32	-3.24
AAE-FG	60.41	59.50	-0.91	32.14	28.44	-3.70
CDCP	64.78	65.16	0.38	20.65	18.20	-2.45

the output sequence, allowing longer paragraphs to fit within the permissible output length.

For all the datasets: AAE, AAE-FG and CDCP, the *Abbreviated* formulation achieved impressive average reductions in output sequence length of 23.56%, 23.01%, and 16.45%, respectively. As shown in Table 3.4, this formulation resulted in competitive performance for the ACE task, showcasing its efficiency. While the ARC task experienced a noticeable decline in performance across all datasets with this formulation, the substantial reduction in output length represents a compelling trade-off. This suggests that this formulation effectively balances performance and efficiency, capturing essential information with shorter spans in the form of *"IDs"*. However, for tasks that demand a deeper understanding of relational context, the *arg*TANL framework, with its longer spans, remains superior. Overall, the *Abbreviated* formulation presents pragmatic and resourceful insights, offering significant benefits in terms of output length reduction while maintaining competitive performance for certain tasks.

3.5.5 Error Analysis

The proposed method sometimes produces invalid ANLs as the generation is not fully controllable. We identified the following three major types of ANL-related erroneous generation (see Table 3.5): *(i) Invalid Token:* The generated ANL consists of some out-of-vocabulary tokens or out-of-context text spans (*Hallucinations*). *(ii) Invalid Format:* The invalid ANL format includes mismatched brackets, symbols, or corrupted text. *(iii) Invalid Component:* The tail component connected with the relation in ANL is invalid if it is a span of text from the non-component regions. Importantly, these errors are not very frequent, and erroneous generations are discarded as negative results without undergoing any additional post-processing. Among all the datasets, D-MKT is the most effective in generating error-free ANL, whereas E-MKT produces the highest number of errors for the ACRE task.

TH-4046_216101004

Table 3.5 Error analysis for the ACRE task. **IT, IC,** and **IF** refer to *Invalid Token, Invalid Component,* and *Invalid Format* respectively.

Corpus	Model	IT	IC	IF
AAE	$T5_{argTANL}$	2.95	4.51	1.11
	$T5_{E-MKT}$	5.45	6.62	3.39
	$T5_{A-MKT}$	2.61	5.4	1.05
	$T5_{SM-MKT}$	3.06	5.23	1.39
	$T5_{D-MKT}$	**2.39**	**4.9**	**0.8**
	$T5_{ME-argTANL}$	4.95	5.57	1.00
AAE-FG	$T5_{argTANL}$	3.23	6.07	1.05
	$T5_{E-MKT}$	10.47	10.41	7.01
	$T5_{A-MKT}$	3.39	8.85	**1.00**
	$T5_{SM-MKT}$	3.45	**5.29**	1.33
	$T5_{D-MKT}$	**2.95**	5.73	1.22
	$T5_{ME-argTANL}$	5.45	7.29	1.05
CDCP	$T5_{argTANL}$	11.33	**4.93**	7.6
	$T5_{E-MKT}$	27.33	15.2	**5.73**
	$T5_{A-MKT}$	13.6	8.93	7.33
	$T5_{SM-MKT}$	12.53	6.4	7.2
	$T5_{D-MKT}$	**9.6**	**4.93**	7.06

3.6 Conclusion

This chapter introduces *arg*TANL framework, a generative approach to AM. We demonstrate a text-to-text generation model could effectively handle the complexities of *End-to-End* AM tasks. We further enhance the *arg*TANL framework through *Marker-Based Fine-Tuning*, which utilizes argumentative markers and DMs to improve the performance. This fine-tuning step enables the model to better recognize and extract ACs by learning the marker information. Additionally, we propose *ME-arg*TANL framework, which directly integrates marker knowledge into the output sequence. *ME-arg*TANL framework simultaneously identifies markers and argumentative structures. This dual-task approach yields superior performance compared to traditional methods. Through extensive experimentation on multiple AM benchmarks, we validate the proposed approaches *arg*TANL, Marker-Based Fine-Tuned *arg*TANL, and *ME-arg*TANL by showcasing strong performance. The ablation study on Abbreviated *arg*TANL provides additional evidence of the efficiency and effectiveness of our framework, especially in managing longer sequences. Thus, our research highlights the potential of generative models in the AM domain. This also paves the way for future research in integrating domain-specific knowledge into generative models for complex NLP tasks.

TH-4046_216101004

TH-4046_216101004

Chapter 4

On the Role of Key Phrases in Argument Mining

Chapter Highlights

- This chapter addresses the limitations of surface-level cues by focusing on the deep conceptual connections between argumentative components, introducing the concept of **Related Key Phrases**.
- **LLM-Based Key Phrase Extraction:** A prompt-based strategy is employed, utilizing a Large Language Model (LLM) to automatically extract the semantic key phrases that form a conceptual bridge between related AC pairs.
- **Key Phrase-Enhanced ANL:** The extracted key phrases are integrated directly into the target ANL sequence. This allows the generative model to explicitly learn and leverage the internal semantic links when modeling argumentative relations.
- The framework is evaluated on the joint formulation of **ACC** and **ARI** across three structurally diverse benchmarks: **AAE**, **AAE-FG**, and **CDCP**. The proposed method establishes a new SoTA on all datasets.
- The publication related to this chapter is as follows:
  1. **Nilmadhab Das**, V. Vijaya Saradhi, Ashish Anand, "On the Role of Key Phrases in Argument Mining," in *Findings of the 2025 Conference of the Nations of Americas Chapter of the Association for Computational Linguistics: **NAACL**, New Mexico, 2025.*

4.1 Introduction

In the preceding chapter, we introduced a generative framework, argTANL, which reframed end-to-end AM as a text-to-text generation task. That work demonstrated

TH-4046_216101004

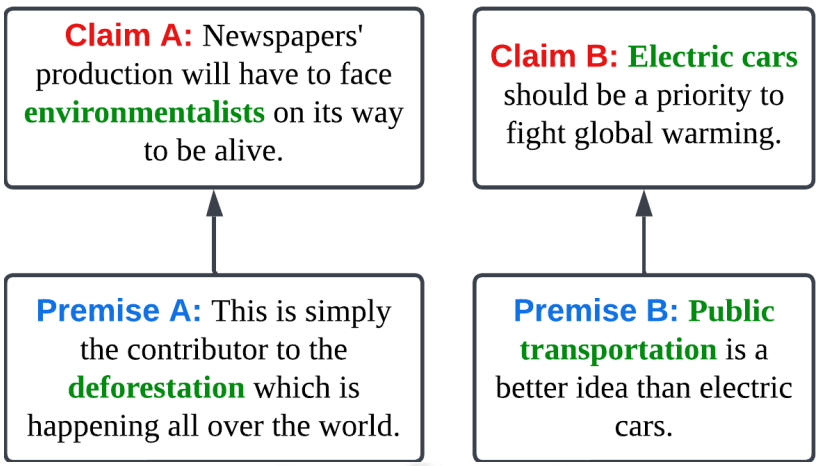

Fig. 4.1 Examples of *Related Key Phrases* between related AC pairs, highlighted in green. Claim A is supported by Premise A, with two key phrases serving as conceptual bridges between these components. Similarly, Premise B attacks Claim B, as they are connected by conceptually opposing key phrases.

that by leveraging explicit linguistic signals, specifically argumentative and discourse markers, we could improve a model's ability to jointly identify ACs and their corresponding ARs. This marker-based approach proved effective for streamlining the end-to-end pipeline and capturing the overall structure of an argument.

However, that methodology, while successful, primarily relies on surface-level cues that signal the presence or boundary of an argumentative unit. It does not fully address the more complex challenge of modeling the deep conceptual connection that underpins the relationship between two components. For instance, while a marker like "because" indicates a supporting relationship, it does not capture why the premise logically supports the claim based on its content. This leaves a critical gap: how to model the internal semantic links that form the true backbone of an argumentative relation, especially when explicit markers are absent. Alongside, the previous method also produced limited performance for the most difficult ARI task across datasets.

This chapter delves deeper into this challenge by shifting focus from external linguistic signals to the internal semantic content of the arguments themselves. We posit that the most crucial links between related ACs are often encapsulated in *Related Key Phrases*, which are conceptual bridges that semantically connect a premise to a claim. For example, in figure 4.1, the link between Premise A discussing *"deforestation"* and Claim A about *"environmentalists"* is a core element of their argumentative relationship that previous methods, including the marker-based approach, often overlook. Similarly, both Claim B and Premise B contain the phrase *"Electric cars"*, suggesting a potential connection between these ACs. Yet, this similarity does not effectively convey the attacking nature of the relationship between them. Instead, it is the conceptual contrast

between *"Electric cars"* in Claim B and *"Public transportation"* in Premise B that reveals the true argumentative nature of the relation, where opposing ideas are at play.

To explore this hypothesis, this chapter presents a unified approach to jointly model a subset of the key tasks in AM: ACC and ARI, given that the ACI task is already solved. Our method leverages a text-to-text generation framework, where both input and output are structured using the ANL format. In the input, AC spans are augmented and enclosed with special symbols. On the output side, AC types and their corresponding related spans are embedded with augmented labels, also using special symbols. To invoke the knowledge of *related key phrases*, we first prompt an open-source LLM, guiding it in extracting these phrases between pairs of ACs from the standard AM datasets. Such extracted phrases are then appended to the ANL output sequence, embedding this knowledge directly into the text-to-text generation task. Through extensive experiments across multiple structurally diverse standard AM benchmarks, our proposed approach achieves SoTA results for both tasks, outperforming all existing baselines. These results highlight the strong potential of our method in effectively addressing two distinct AM tasks jointly. In summary:

1. We propose a joint modeling of two key AM tasks, *ACC* and *ARI*, using an *ANL-based* text-to-text generation framework to model both tree and graph (non-tree) structured arguments.
2. We integrate the explicit knowledge in terms of *related key phrases* within the target ANL to strengthen its relational representation.
3. We utilize *LLM-based prompting* techniques to extract *related key phrases* between related AC pairs, which in turn enhances the proposed task performance.
4. We conduct extensive experiments achieving SoTA results across datasets and demonstrate strong noise resilience through a *Noise Adaptability Study*. We also assess the *impact of external knowledge* and the *utility of joint modeling* with an *in-depth argumentative structural analysis*.

4.2 Proposed Method

The proposed method consists of two main steps. *First,* the input ANL is represented with special markers to indicate argumentative spans. For example: “Nevertheless, supporters would argue that **[Argument 1]**. They further point out that **[Argument 2]**." Here, the special tokens “[" and “]" denote the start and end of each argument. The output representation further extends this format to include argument classification and relationships: “Nevertheless, supporters would argue that **[Argument 1 | Claim | Argument 2]**". This indicates that **Argument 1** is classified as a **Claim** and is related to **Argument 2**. *Next,* from each argument pair, related key phrases are extracted

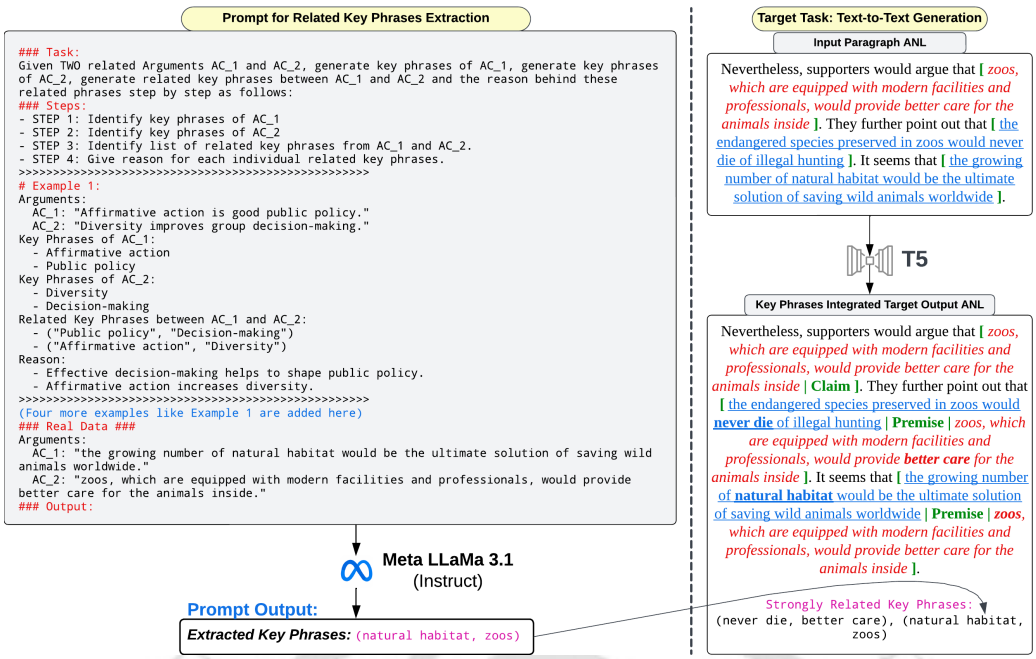

Fig. 4.2 Illustration of the proposed method. On the left, the prompt description for extracting *Related Key Phrases* using the LLM in a 5-shot setting is shown. On the right, the input/output ANL configurations for the proposed generation task are presented. Claim spans are marked in red, Premises are in blue, and augmented labels are in green. The *Related Key Phrases* within the AC spans are in **bold** and are explicitly appended at the end of the output.

using `Lamma-3.1-instruct` in a few-shot setup. Then, these extracted key phrases are appended with the target ANL to produce the final ANL output.

This structured input and output ANL are then used to fine-tune a T5 model. During inference, T5 classifies AC spans, predicts argument relations, and generates key phrases as a by-product. Figure 4.2 illustrates the overall process. Further details are discussed in subsequent sections.

4.2.1 Key-Phrases Extraction with LLM

Utilizing an open-sourced LLM *Meta-LLaMa-3.1-instruct*, we employ a prompt-based strategy to extract *related key phrases* from the related ACs. To enhance the quality of the extracted key phrases, we adopt a few-shot prompting approach, where we manually create five carefully crafted examples from a standard AM dataset. These examples illustrate both the input and the expected output, helping guide the model towards producing the desired results. To prevent the model from generating irrelevant or random text, we design a well-structured instruction composed of four distinct steps, each aimed at constraining the extraction process (See Figure 4.2). The first step consists of extracting key phrases from *AC*_1, focusing on the most relevant

phrases that reflect its core argument. In the second step, the model performs a similar extraction from the *AC*_2. Notably, both *AC*_1 and *AC*_2 may contain more than one central point. Next, in the third step, those extracted phrases from both the ACs are matched by identifying conceptual links to establish a connection between these arguments. In the fourth step, the model provides an explanation of how and why the chosen pairs of related key phrases are linked. The reasoning is expressed through meaningful sentences that articulate the logical or thematic connections, ensuring that the extracted links are relevant and coherent. By reasoning this way, the model is forced to reduce the likelihood of extracting unrelated key phrases.

4.2.2 Joint Formulation of ACC & ARI

We define the proposed ANL formatting using five core elements: (i) the input paragraph $W = w_1, w_2, w_3, \ldots, w_n$, where n denotes the total number of tokens in W, and a contiguous span covering tokens w_i through w_j is represented as $w_{i:j}$. (ii) a list of ACs, $C = \{(c_i, t_i, s_i, e_i)\}_{i=1}^{n}$, where c_i represents the AC span, t_i specifies its type (e.g., Claim, Premise), s_i is the start index, and e_i is the end index of the AC; (iii) a list of ARs, $R = \{(h_j, t_j)\}_{j=1}^{m}$, where $h_j, t_j \in C$ are the head and tail ACs, respectively; (iv) a list of LLM-extracted related key phrases from all the ARs is denoted by $KP = \{(e_{1_h}, e_{1_t}), (e_{2_h}, e_{2_t}), \ldots, (e_{k_h}, e_{k_t})\}$, where e_{i_h} and e_{i_t} denote the related key phrases extracted from related head and tail ACs, respectively; (v) a set of special symbols $S = \{[,], (,), |, =\}$, where "[" marks the start of an augmented label, "]" marks the end of an augmented label, "(" and ")" are used to enclose the phrases, "|" separates different augmented labels, and " = " connects the related AC spans. Now, consider two arbitrary ACs C_p and C_q from the set C, which are related by an AR R_k from R, where $h_k = C_p$ and $t_k = C_q$. The spans of C_p and C_q are denoted by $w_{c_p^s:c_p^e}$ and $w_{c_q^s:c_q^e}$, respectively. In the input W, the spans of those ACs are constructed as: $[w_{c_p^s:c_p^e}]$ and $[w_{c_q^s:c_q^e}]$. And, in the output, the augmented labels are created as: $[w_{c_q^s:c_q^e} \,|\, c_q \,|\, w_{c_p^s:c_p^e}]$ to make connection between the pairs of ACs, and for the head AC C_p, the augmented label is written as: $[w_{c_p^s:c_p^e} \,|\, c_p]$. The remaining tokens in W are rewritten as they are for both input and output. Finally, all the related key-phrase from all the ARs present in W are appended at the end of the label-augmented paragraph as:

`Strongly Related Key Phrases:` (e_{1_h}, e_{1_t}), (e_{2_h}, e_{2_t}), ..., (e_{k_h}, e_{k_t}).

An illustrative example of the proposed formulation is presented in Figure 4.2. The figure demonstrates both the ANL-formatted input and the corresponding target output used for the joint ACC and ARI generation task. In the input paragraph ANL, the complete argumentative component spans are explicitly enclosed within square brackets to preserve their exact boundaries. For example, the Claim span highlighted in red is represented as: `[ zoos, which are equipped with modern facilities and`

`professionals, would provide better care for the animals inside ]`. Similarly, the two Premise spans highlighted in blue are represented as: `[ the endangered species preserved in zoos would never die of illegal hunting ]` and `[ the growing number of natural habitat would be the ultimate solution of saving wild animals worldwide ]`. These bracketed spans allow the model to identify the exact argumentative component boundaries within the paragraph context. In the target output ANL, the same spans are augmented with structural labels and relational information. The Claim span is generated as: `[ zoos, which are equipped with modern facilities and professionals, would provide better care for the animals inside | Claim ]`, where the appended label explicitly denotes the component type. The first Premise span is generated as: `[ the endangered species preserved in zoos would never die of illegal hunting | Premise | zoos, which are equipped with modern facilities and professionals, would provide better care for the animals inside ]`. Likewise, the second Premise span is generated as: `[ the growing number of natural habitat would be the ultimate solution of saving wild animals worldwide | Premise | zoos, which are equipped with modern facilities and professionals, would provide better care for the animals inside ]`. In both cases, the first segment corresponds to the complete Premise span, the second segment specifies the argumentative component type, and the final segment explicitly encodes the related Claim span, thereby jointly modeling ACC and ARI within a unified generation sequence. Finally, the related key phrases extracted using the LLM-based phrase extraction module are appended at the end of the output as: `Strongly Related Key Phrases: (never die, better care), (natural habitat, zoos).`

4.3 Experimental Setup

4.3.1 Datasets

We evaluate our proposed method on these three AM benchmarks: **AAE**, **AAE-FG** and **CDCP**. Both AAE and AAE-FG are tree-structured, while CDCP is graph-structured.

4.3.2 Implementation Details

We utilize the Flan-T5 model family [103] in three variants: *Base* (220M), *XL* (3B), and *XXL* (11B) for all our experiments. A batch size of 32 is used for `AAE` and `AAE-FG`, and that of 16 is used for `CDCP`. In each case a maximum input/output sequence length of 1024 is kept. We apply a learning rate of 0.0005 with the AdamW optimizer. All

experiments are conducted on a single A100 GPU over 10000 steps, with checkpoints taken every 400 steps. The results are averaged over three independent runs. For the *XL* and *XXL* variants, we incorporate QLoRA adapters [104] for parameter-efficient fine-tuning. The QLoRA configuration uses a rank (r) of 16, a LoRA alpha of 32, and a dropout of 0.05. The adapters are applied to the query (q), key (k), value (v), output (o), and feed-forward network layers (wo, wi_0, wi_1). The base model is loaded in 4-bit using the NormalFloat4 (`nf4`) quantization type with double quantization, while the computation is performed using `torch.bfloat16`.

4.3.3 Evaluation

Following the prior studies [72, 63], we evaluate the performance of the proposed method using the *Macro-Averaged F1* score for both ACC and ARI tasks. We treat an AC span as correct only when it exactly matches the gold boundaries, and we ignore partial overlaps to ensure strict consistency with prior benchmarks.

4.3.4 Baselines

We use the following SoTA joint models as baseline systems in our experiments:

Joint-ILP [12]: An argumentation structure parser that performs joint optimization of ACs and ARs using Integer Linear Programming (ILP).

St-SVM [11]: A structured SVM that models AM as an inference problem in both full and strict factor graph.

Joint-Ptr-Net [69]: This joint model leverages a Pointer Network architecture to simultaneously address ACC and ARI tasks.

Deep-Res-LG [70]: It utilizes a residual network with link-guided training.

Span-LSTM [68]: An LSTM-based span representation with argumentative markers for improved AC and AR processing.

TSP-PLBA [71]: It employs task-specific parameterization for encoding ACs and a biaffine attention module for ARs.

BERT-Trans [72]: A neural model that employs a transition-based approach.

SB-Parser [73]: Latest dependency parsing approach using Longformer combined with a Span-Biaffine(SB) architecture for sub-task information sharing.

MRC-GEN [63]: A multi-hop generative formulation converting the AM tasks into machine reading comprehension.

Error Type	Related AC Spans	Extracted Key Phrases	Correct Key Phrases
Generated New Words	AC_1: The better a person feels, the better his brain works. AC_2: Putting physical activities in early steps of human development would finally lead to mentally healthy society.	*(Better feelings, Mentally healthy society)*	*(Better his brain works, Mentally healthy society)*
Smaller or Larger Phrases	AC_1: The more cars and motorbikes are on roads, the more seriously the ozone layer is damaged. AC_2: This is sure to lead to more carbon emitted into the atmosphere, which can cause skin cancer.	*(Cars, Carbon emitted)*	*(Cars and motorbikes, Carbon emitted)*
Unrelated Phrases	AC_1: Roommate turns down the music or television volume at night time. AC_2: Consideration is always important in relationship.	*(Nighttime, Relationship)*	*(Roommate, Relationship)*

Table 4.1 Examples of different erroneous extraction of related key phrases using *Meta-LLaMa-3.1-instruct*.

PITA [105]: A generative multitask formulation with prompt-tuning for efficient task interactions.

4.4 Results and Discussion

4.4.1 Comparison with Baseline Models

Table 4.2 compares the proposed method with the SoTA baselines. Our approach consistently outperforms all existing baselines by a significant margin in both ACC and ARI tasks, achieving new SoTA results across all datasets. While most baselines perform well on the tree-structured `AAE` dataset, they struggle with the graph-structured `CDCP` dataset, particularly for the ARI task. The mix of transitive and non-transitive relations in `CDCP` pose challenges that our method handles relatively better. In this dataset, the proposed method marginally outperforms the latest baseline by 0.6% F1 score for the ACC task with an impressive relative improvement of F1 score of 9% for the challenging ARI task. Such improvements indicate the strength of our method in managing complex relationships. In the case of the tree-structured `AAE` dataset, it gave similar F1 scores or marginal improvement over the compared baselines for both the ACC and ARI tasks. This shows that the proposed approach works equally well for both tree-structured and graph-structured data. For the `AAE-FG` dataset, we benchmark our results against the *SB-Parser* baseline. Considering the original data content of `AAE` is classified into nine AC classes in `AAE-FG` as compared to three AC classes in the original one, this baseline struggles to classify those fine-grained AC classes correctly, resulting in poor performance in the ACC task. In contrast, our method achieves a significant 17.38% F1 score gain in ACC and a 1.65% F1 score improvement in ARI as compared to the baseline. Such strong performance of our ***ANL-based method***

Corpus	Model	Macro-F1		AVG
		ACC	ARI	
CDCP	Deep-Res-LG	65.3	29.3	47.3
	St-SVM (strict)	73.2	26.7	50.0
	TSP-PLBA	78.9	34.0	56.4
	BERT-Trans	82.5	37.3	59.9
	SB-Parser	82.3	40.1	61.2
	PITA	83.6	44.9	64.3
	Base **(Ours)**	77.9	35.3	56.6
	XL **(Ours)**	**84.2**	53.4	68.8
	XXL **(Ours)**	**84.2** (+0.6)	**53.9** (+9.0)	**69.0** (+4.7)
AAE-FG	SB-Parser*	56.6	67.8	62.2
	Base **(Ours)**	71.4	66.1	68.8
	XL **(Ours)**	**74.0** (+17.4)	**69.4**	**71.7** (+9.5)
	XXL **(Ours)**	73.3	**69.4** (+1.6)	71.3
AAE	Joint-ILP	82.6	58.5	70.6
	St-SVM (full)	77.6	60.1	68.9
	Joint-Ptr-Net	84.9	60.8	72.9
	Span-LSTM	85.7	67.8	76.8
	SB-Parser	86.8	69.3	78.1
	BERT-Trans	88.4	70.6	79.5
	MRC-GEN	89.2	70.9	80.1
	PITA	88.3	**73.5**	80.9
	Base **(Ours)**	87.4	69.3	78.4
	XL **(Ours)**	89.4	72.7	81.1
	XXL **(Ours)**	**89.5** (+0.3)	**73.5**	**81.5** (+0.6)

Table 4.2 Comparison of experimental results against the baselines. Best scores are marked in **bold**. Recent SoTA results are underlined. * indicates the baseline results produced by running the corresponding open-source code with original hyperparameters. All values are rounded to one decimal place. *Base*, *XL*, and *XXL* refer to different variants of the *Flan-T5* model.

TH-4046_216101004

Corpus	Variant	Macro-F1	
		ACC	ARI
CDCP	With KP	77.88	**35.28**
	Without KP	**80.58** (+2.7)	34.27 (-1.01)
AAE-FG	With KP	**71.39**	**66.13**
	Without KP	69.83 (-1.56)	64.07 (-2.06)
AAE	With KP	**87.44**	**69.26**
	Without KP	87.20 (-0.24)	68.02 (-1.24)

Table 4.3 Experimental results of *"with"* or *"without"* the related key phrases (KP) in the target ANL with *Flan-T5-Base*. Best scores are marked in **bold**.

across diverse datasets, whether tree-structured or not, highlights its generalizability to manage complex argumentative structures, all while achieving SoTA results.

4.4.2 Impact of Related Key Phrases

To assess the contribution of the related key phrases, we perform the proposed task *without using key phrase information* in the output ANL. For this, we use the *Flan-T5-Base* model. As shown in Table 4.3, omitting key phrases leads to a decline in performance in most of the cases across all datasets. This highlights the ***positive impact of key phrase information*** within the ANL in improving both ACC and ARI tasks. By generating this information, the model gains an internal understanding of the phrase-level AR connections, enabling it to better distinguish between related ACs. As a result, the likelihood of incorrectly associating non-related ACs is reduced, which significantly boosts ARI performance. Noticeably, the drop in ARI performance is more substantial than in ACC for all datasets. However, there is an exception: the `CDCP` dataset actually performs better *without key phrases* in the ACC task. This exception could be due to 207 out of 731 paragraphs lacking ARs, meaning a significant portion of the target ANL lacks key phrase information. This increases the difficulty for the model to generalize across both types of ANL sequences in the same training set: *those that contain key phrases* and *those that do not*. This complex distribution negatively impacts the ACC task performance when the *key phrase* information is on.

4.4.3 Human Evaluation of the Extracted Key Phrases

Since standard AM benchmarks lack explicit annotations for related key phrases, we conducted a manual evaluation. Two annotators (authors of this paper) assessed 10% of randomly selected related ACs and their key phrases from the `AAE` and `CDCP` training

sets, classifying them as *Correct* or *Wrong*. Out of 506 key phrases analyzed, 451 and 442 were correct, yielding an average accuracy of **88.24%**. The main error types are summarized in Table 4.1.

Generated New Words: The model generates a new phrase instead of extracting it directly from the span, resulting in a mismatch with the original span words. This error is the most frequent one.

Smaller or Larger Phrases: The extracted spans are either too short or too long. Very few instances of this type of error were found.

Unrelated Phrases: The extracted phrases are not semantically related to the corresponding ACs.

We relied entirely on the capabilities of the LLM for related key phrase extraction tasks and did not manually filter the erroneous extractions, leaving them unchanged as they were utilized during training for the proposed generation task. Despite using these silver-standard key phrases, we achieved SoTA results, highlighting the strength of the proposed method even when the utilized key phrases are partially noisy.

4.4.4 Joint vs. Standalone Formulation

To evaluate the importance of solving ACC and ARI tasks together rather than handling them independently, we propose two standalone task formulations by modifying the original ANL: *(i) ACC Only*, which focuses solely on the ACC task; *(ii) ARI Only*, which handles only the ARI task. In the *ACC Only* formulation, we remove the related AC span $w_{c_p^s:c_p^e}$ from the original augmented label $[w_{c_q^s:c_q^e} \mid c_q \mid w_{c_p^s:c_p^e}]$, creating a new augmented label $[w_{c_q^s:c_q^e} \mid c_q]$ in the target ANL sequence. In the *ARI Only* formulation, the AC type c_q is removed to make a new augmented label as $[w_{c_q^s:c_q^e} \mid w_{c_p^s:c_p^e}]$, which only contains the two related AC spans without their class labels. The input ANL format remains unchanged for both formulations. We formulate the target ANL in both *with* and *without* key phrase information settings.

Table 4.4 shows the joint setting outperforms the standalone formulations in nearly all cases. This demonstrates the advantage of joint modeling over treating the AM tasks independently. The most significant drop of 4.2% F1 scores was observed for the ARI task on `AAE` while using key phrases. Performance drop is also observed for the ACC task, with a maximum drop of 1.65% F1 scores on AAE. The joint formulation benefits from the mutual feature sharing across different tasks formulated in the same target ANL, uplifting each others' performances. In contrast, the standalone formulations miss out on these advantages, leading to lower performance. However, we observe two exceptions, one each in `CDCP` and `AAE-FG`. For the `CDCP`, the standalone model using

KP Status	Corpus	Variants	Macro-F1	
			ACC	ARI
with KP	CDCP	Joint	77.88	**35.28**
		ACC Only	**79.71** (+1.83)	–
		ARI Only	–	33.73 (-1.55)
	AAE-FG	Joint	**71.39**	**66.13**
		ACC Only	70.29 (-1.1)	–
		ARI Only	–	65.06 (-1.07)
	AAE	Joint	**87.44**	**69.26**
		ACC Only	86.75 (-0.69)	–
		ARI Only	–	65.06 (-4.2)
w/o KP	CDCP	Joint	**80.58**	**34.27**
		ACC Only	79.83 (-0.75)	–
		ARI Only	–	33.22 (-1.05)
	AAE-FG	Joint	69.83	64.07
		ACC Only	69.83	–
		ARI Only	–	**66.12** (+2.05)
	AAE	Joint	**87.20**	**68.02**
		ACC Only	85.55 (-1.65)	–
		ARI Only	–	66.12 (-1.9)

Table 4.4 Joint Vs. Standalone formulation of AM tasks including *"with"* and *"without"* Key Phrases (KP) with *Flan-T5-Base*. Best scores are in **bold**.

Corpus	Variant	Macro-F1	
		ACC	ARI
CDCP	No Noise	**77.88**	35.28
	Noise-Inf	75.11 (-2.77)	28.29 (-6.99)
	Noise-FT	77.65 (-0.23)	**36.81** (+1.53)
AAE-FG	No Noise	71.39	**66.13**
	Noise-Inf	70.81 (-0.58)	60.89 (-5.24)
	Noise-FT	**71.48** (+0.09)	64.02 (-2.11)
AAE	No Noise	87.44	**69.26**
	Noise-Inf	86.14 (-1.3)	65.09 (-4.17)
	Noise-FT	**87.50** (+0.06)	68.02 (-1.24)

Table 4.5 Performance comparison under different noise setups using *Flan-T5-Base*. *No Noise* refers to training and inference without noise, *Noise-Inf* adds noise only during inference, and *Noise-FT* includes noise in both training and inference.

key phrases performs better on the ACC task. This is likely due to the fact that 207 out of 731 paragraphs lack ARs in the joint formulation, complicating generalization across ANL sequences *with* and *without* ARs. In contrast, the exclusion of ARs in the standalone formulation enhances the performance in this dataset. In the AAE-FG dataset, the standalone ARI task yields better results than the joint formulation with muted key phrase information. The higher number of AC types in this dataset may lead to increased number of misclassified ACs, resulting in *error propagation* with inaccurate relation identification in the joint formulation. However, by removing AC-type information in the standalone ARI task, the model improves performance by reducing the chance of error propagation.

4.4.5 Noise Adaptability Study

This analysis aims to evaluate the robustness of our proposed method to noise by introducing noisy sentences into the input ANL. These noisy sentences are unrelated to the paragraph's content, often presenting an opposing viewpoint or being distinctly off-topic. We use few-shot prompting containing manually created noise with *Meta-LLaMa-3.1-instruct* to generate noisy sentences. On average, the AAE and CDCP datasets contain 72.35 and 111.2 tokens per paragraph, respectively. Inserting a single noisy sentence in a paragraph adds roughly 18.57% noisy tokens to AAE and 17.2% to CDCP. For this significant amount of noise injection, we define two experimental setups: **(I)** *Train the model with noise and evaluate with noisy inputs.* **(II)** *Train the model without noise and evaluate with noisy inputs.* Table 4.5 presents the results under these

Corpus	Model	Macro-F1		AVG
		ACC	ARI	
CDCP	**Flan-T5 (Fine-Tuned)**	**80.58**	**34.27**	**38.51**
	LLaMa *(5-shot)*	39.96	8.50	16.41
	LLaMa *(10-shot)*	52.95	8.88	20.82
	LLaMa *(20-shot)*	52.40	7.64	20.29
AAE-FG	**Flan-T5 (Fine-Tuned)**	**69.83**	**64.07**	**44.88**
	LLaMa *(5-shot)*	25.73	38.75	21.76
	LLaMa *(10-shot)*	37.61	40.95	26.40
	LLaMa *(20-shot)*	39.52	43.95	28.08
AAE	**Flan-T5 (Fine-Tuned)**	**87.20**	**68.02**	**51.97**
	LLaMa *(5-shot)*	43.56	19.80	21.37
	LLaMa *(10-shot)*	42.53	27.36	23.51
	LLaMa *(20-shot)*	42.46	39.59	27.62

Table 4.6 Performance comparison of Few-shot *Meta-LLaMa-3.1-instruct* Vs. Fine-tuned *Flan-T5-Base*.

conditions. The performance on the ACC task remains notably stable across datasets in both setups, highlighting the resilience of our method in noisy environments. However, for the ARI task, there is a significant drop in performance when noise is introduced during inference without prior exposure in training. When the model is trained with noise, the drop is notably smaller.

4.4.6 Few-shot Decoder-based LLM vs Fine-tuned Flan-T5-Base

Table 4.6 presents a comparison between the performance of 5-shot, 10-shot, and 20-shot experiments using the SoTA decoder-based LLM *Meta-LLaMa-3.1-instruct* and the fine-tuned *Flan-T5-Base*, both evaluated without key-phrase information. Although the few-shot performance improves as the number of examples increases, it consistently falls short of the fine-tuning approach, which outperforms it by a substantial margin across all datasets. It is worth noting that we did not conduct an extensive search for the optimal prompt in the AM task, as determining the most effective prompt can be quite challenging. Instead, we use the same input-output combination employed in the text-to-text generation of the proposed model to construct the few-shot prompts.

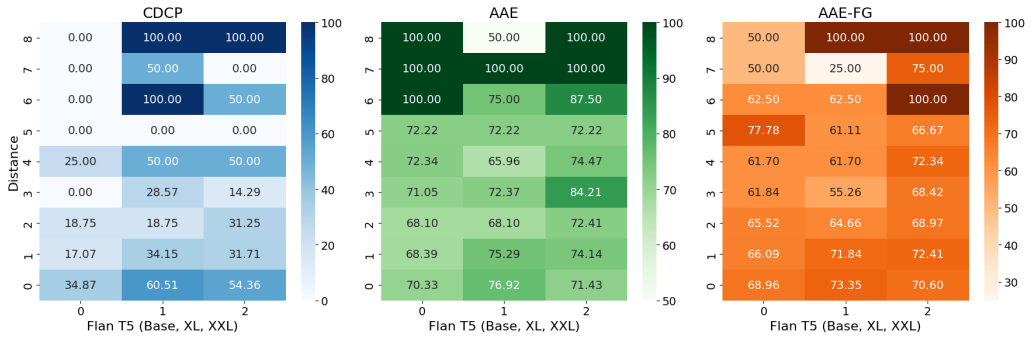

Fig. 4.3 Performance comparison of the ARI task in capturing long-range relations, based on the distance between related ACs, measured by the number of components separating them. The X-axis represents different Flan-T5 variants *(Base, XL, XXL)*, while the Y-axis indicates the distance.

Dataset	Model	2-length (%)	3-length (%)	4-length (%)	AVG (%)
CDCP	Base	6.52	3.57	0.00	3.36
	XL	**26.09**	**25.00**	**33.33**	**28.81**
	XXL	10.87	7.14	0.00	6.67
AAE-FG	Base	38.44	27.06	0.00	21.83
	XL	**46.26**	35.29	6.25	29.93
	XXL	43.54	**35.88**	**18.75**	**32.06**
AAE	Base	39.46	30.00	**18.75**	29.40
	XL	**50.68**	**40.59**	6.25	**32.51**
	XXL	43.88	33.53	12.50	29.97

Table 4.7 Performance analysis of *n-length* chains in terms of Accuracy(%), where $n = \{2,3,4\}$.

Ground Truth ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **Premise** | the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies]. `Strongly Related Key Phrases: (Merits, Advertising expenses)`
Flan-T5-Base generated ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **MajorClaim**].
Flan-T5-XXL generated ANL	That is not to say, however, that advertisements have no downsides. Of course, [the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies | **Claim**]. Yet, [its merits still outweigh these downsides | **Premise** | the advertising expenses lead to a higher product price and some of them express fake information, creating information asymmetry between consumers and companies]. `Strongly Related Key Phrases: (Merits, Advertisement expense)`

Table 4.8 Errors where missing key phrases in the generated sequence cause incorrect AC classification with *Flan-T5-base*, while the larger *Flan-T5-XXL* generates the correct ANL sequence. Incorrect generations are marked in red, and correct ones in green.

4.4.7 Argumentative Structural Analysis

We perform an in-depth performance study of our proposed approach in three important argumentative structural aspects as follows:

(I) Analysing the Relation Chains: Consider an argumentative paragraph containing multiple ACs, denoted as C_1, C_2, C_3, C_4, and C_5. These ACs may form a sequential chain such as $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5$, with C_5 being the root AC. Each consecutive AC is related to the previous one, thereby forming a *4-length* chain, as four relations connect these five ACs. Several such *n-length* chains are present in a paragraph in both AAE and CDCP, where $n = \{2, 3, 4\}$. In such configurations, relation identification between two closely connected ACs becomes challenging. Because, given that all ACs in the chain are (in-)directly related through the root, there is a high likelihood of incorrectly identifying relations, such as predicting $C_3 \rightarrow C_5$ instead of the correct $C_4 \rightarrow C_5$. Table 4.7 reports the performance of different length chains across different datasets with all three model variants. Considering the inherent difficulties, the results are promising in both tree and graph-structured datasets. As the chain length increases, the larger models mostly show better capability in detecting these chains correctly. Notably, we consider the counts of smaller sub-chains that are part of the bigger chains in our analysis.

(II) Capturing Long-Range Relations: In an argumentative paragraph, some ACs are related despite being far apart. For instance, one AC may appear at the start, while the other is found near the end. We compare the long-range relation identification performance based on the number of intermediate ACs between the linked components. Figure 4.3 shows a heatmap comparing model performance across varying distances. For both versions of the AAE dataset, most model variants excel at capturing these distant connections. However, in CDCP, the *Base* model struggles, and the *XL* model outperforms the *XXL* variant. Short-range relations with few or no intermediate ACs are more prevalent across the datasets, making them easier for most models to identify. Interestingly, the strong performance on long-range connections, despite their presence in lower numbers across datasets, showcases the effectiveness of our approach.

(III) Performance of Transitive Relations: The inherent challenge of the ARI task for graph-structured arguments lies in the presence of additional transitive relations. We evaluate the performance of identifying these transitive relations on the graph-structured CDCP corpus across different model variants. The *base* variant identified 9 out of 31 ground truth transitive relations, achieving an accuracy of 29.03%. The *XL* variant performed better, correctly identifying 14 out of 31 transitive relations with an accuracy of 45.16%. The *XXL* variant exhibited the best results, detecting 17 out of 31 transitive relations, yielding an accuracy of 54.84%. These findings highlight the strength of our method in handling complex graph argumentative structures.

TH-4046_216101004

4.4.8 Error Analysis

In all datasets, there are a few instances where the *Base* model either fails to capture or mistakenly captures relational dependencies. One such instance is shown in Table 4.8. This issue arises due to the erroneous generation or missing key phrases. In contrast, the larger *XXL* model effectively generates correct key phrases, enabling accurate identification of AC types and their spans, thus reducing errors. Also, a small number of cases exhibit unclosed brackets in the augmented labels in the generated ANL sequences. Since these errors are rare, we choose not to address them and retain the generated output as it is.

4.5 Conclusion

This chapter introduces an ANL-based generative framework for jointly modeling ACC and ARI tasks in AM. By utilizing few-shot LLM prompting to extract related key phrases between AC pairs, we enrich the output ANL to improve the capability of capturing complex relations. With extensive experiments across multiple datasets, the proposed approach achieves SoTA results and handles both tree- and graph-structured data with ease, demonstrating strength and generalizability. Moreover, it handles noise effectively and is capable of capturing longer relation chains, long-range relational dependencies and transitive relations. Despite the strong performance of our ANL-based method for AM tasks, several limitations remain. First, the accuracy of the extracted key phrases is closely tied to the quality of the prompts and the performance of the LLM. As the LLM's output is not fully predictable, it can occasionally generate irrelevant or incomplete key phrases. These are discarded, and we do not attempt further extraction from these instances, leaving them empty during the construction of the proposed target ANL. Second, the method's reliance on key phrases introduces variability in performance, particularly in datasets like `CDCP`, where the inclusion of key phrases sometimes leads to performance drops. A more dynamic strategy for key phrase integration, tailored to dataset characteristics, could improve results. Another limitation is the challenge of handling transitive relations in graph-structured arguments. While our method performs well, the accuracy for detecting transitive links remains moderate (up to 54.84%), indicating room for improvement in more complex, non-hierarchical structures. Finally, the joint modeling of ACC and ARI tasks can lead to error propagation, especially in datasets with a large number of AC types, such as `AAE-FG`, where mistakes in one task may negatively impact the other. Refining error mitigation strategies could help reduce these issues. Addressing these limitations in future work will enhance the robustness and adaptability of the proposed method further.

TH-4046_216101004

TH-4046_216101004

Chapter 5

Ent-BAM: A Generative Framework for Biomedical Argument Mining Enriched with Entity Information

Chapter Highlights

- This chapter focuses on enriching biomedical argument mining (BAM) with domain-specific cues by integrating **Outcome entity** information within the ANL-based generative framework.
- **LLM-Based Entity Extraction:** A prompt-based strategy is employed to extract (co-)occurred *Outcome* entities between related ACs from the standard BAM dataset to model the BAM.
- **Entity-Enriched ANL:** The extracted entities are prepended to the target ANL sequence, enabling the model to jointly solve all four BAM tasks: **ACI**, **ACC**, **ARI**, and **ARC**.
- The proposed **Ent-BAM** framework enhances the conditional generation process by aligning entity information with argumentative labels, achieving SoTA results on the standard BAM benchmark.
- The publication related to this chapter is as follows:
  1. **Nilmadhab Das**, Yash Sunil Choudhary, V. Vijaya Saradhi, Ashish Anand, "Ent-BAM: A Generative Framework for Biomedical Argument Mining Enriched with Entity Information," Accepted in ***IEEE Transactions on Artificial Intelligence (TAI), 2025***.

5.1 Introduction

In the preceding chapter, we proposed a key phrase-aware generative framework that enhanced argumentative relation modeling by introducing *Related Key Phrases*. These

TH-4046_216101004

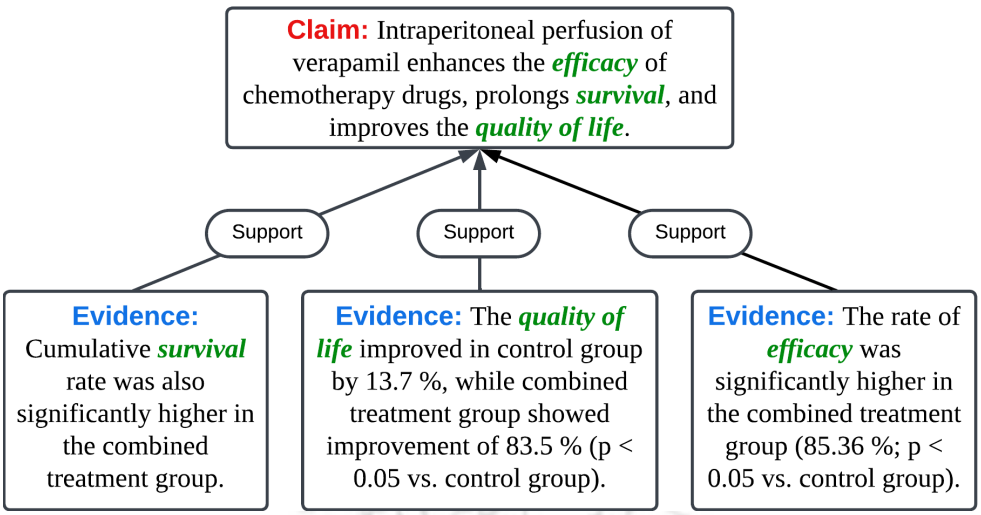

Fig. 5.1 An example of argumentative structure from standard BAM dataset highlighting the occurrences of entities (green) in individual ACs and co-occurrences among related pairs of ACs.

were semantic bridges that connect conceptually linked ACs. This approach demonstrated that enriching ANL sequences with contextual key phrases allows generative models to better capture the internal semantics between ACs, thereby improving relation identification and classification. While this framework effectively handled generic argumentative texts, it remained limited to open-domain AM and did not incorporate domain-specific knowledge critical to specialized fields such as biomedical literature.

BAM presents unique challenges that demand domain-informed reasoning. The ACs in biomedical texts such as clinical trial abstracts are often grounded in PICO elements. It focuses on four key entities: *Patients/Population (P)*, which represents the group of individuals being studied; *Intervention (I)*, referring to the treatment or procedure applied; *Control/Comparison (C)*, which serves as a baseline or alternative to the intervention; and *Outcome (O)*, describing the results or effects observed. These entities are also present in the standard BAM dataset [8], consisting of abstracts of clinical trial reports. A study by Liu *et al.* [83] on this dataset reveals that these entities have significant occurrences in ACs and co-occurrences in related pairs of ACs in AM literature (See Figure 5.1). Similar entities within a pair of ACs strongly suggest a high likelihood that an AR exists between these ACs. The presence of many such entities within a single AC indicates that the AC is the central argument of an argumentative structure. Moreover, in a pair of related ACs, if they share at least one common entity and one of the ACs contains multiple distinct entities, the direction of the AR is likely to flow from the AC with fewer entities to the AC with more entities. Another important observation shows that the ACs in this dataset primarily describe experimental results and conclusions of the clinical trials. As a result, the *Outcome (O)* entity appears most frequently within these ACs [106], emphasizing the significance of *Outcome (O)* entities over other PICO elements in this BAM dataset.

Despite the importance of PICO entities in BAM literature, leveraging these informative rhetorical cues in a BAM system is not straightforward. The primary challenge lies in the lack of PICO entity (co-)occurrence annotations in standard BAM datasets, which makes it difficult to utilise this valuable information effectively. Liu *et al.* [83] made an effort to utilise the *Outcome* entities in BAM for the ARI task through a transfer learning approach. Their method extracted PICO information from an external dataset using an entity-cluster extraction module and later mapped the *Outcome* entities only into the BAM dataset. However, this approach is inherently tedious, as it involves multiple engineering steps and addresses only the ARI task. This leaves significant room to explore efficient methods to incorporate the information of *Outcome* entities in solving all key BAM tasks more comprehensively.

The use of large language models (LLMs) as external knowledge sources has been demonstrated to be highly effective in several studies [107]. Recently, numerous approaches have been developed to leverage LLMs for NLP applications by eliciting external knowledge through prompting. For instance, Wadhwa *et al.* [108] utilised GPT-3 to generate *explanations* detailing why specific tuples of entities are related. These explanations were then appended to the target generation text, providing additional context for the relation extraction task and yielding significant performance improvements. Building on this idea, a similar strategy was applied to NER tasks by Jiang *et al.*[109], leading to a performance gain. Building on the proven capability of LLMs to extract relevant task-specific knowledge, we employ a prompt-based strategy to leverage LLMs for extracting *Outcome* (co-)occurred entities from the BAM dataset. This approach addresses the lack of explicit annotations for such entities in the standard BAM corpus.

Recently, *ANL-based* frameworks [91] have been employed to transform various structured prediction tasks such as *Joint Entity and Relation Extraction*, into text-to-text generation problems, resulting in improved performance. In this setup, plain text is given as input, and target labels are embedded within the generated output to form the ANL sequence. Kawarada *et al.* [76] first apply this approach to BAM tasks, but without incorporating biomedical-specific information, which limited performance on BAM datasets. To address this gap, we use ANL-based formulation with biomedical-specific customizations to improve the modeling of BAM tasks.

In this chapter, we propose **Ent-BAM** *(**B**iomedical **A**rgument **M**ining using **Ent**ities)*, an ANL-based generative framework designed for BAM that jointly addresses all four key BAM tasks. This framework leverages the previously LLM-extracted *Outcome* (co-)occurred entities to construct a biomedical entity-enriched target ANL. Such a biomedical entity-enriched target ANL models the relational dependencies in a simple yet efficient way with biomedical-specific information. Ent-BAM divides the whole process into distinct steps. The first step extracts relevant entities

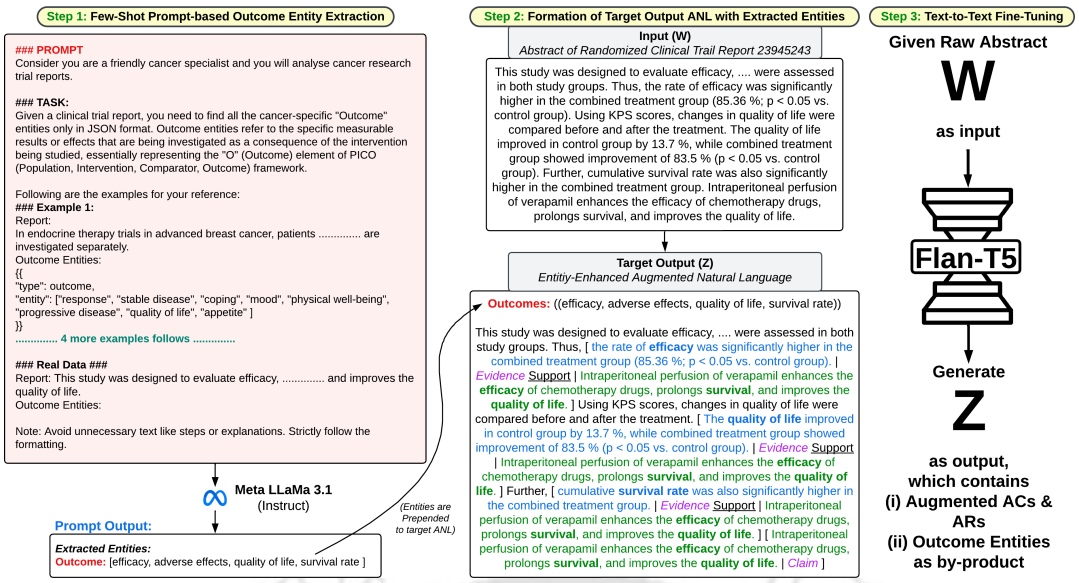

Fig. 5.2 An overview of the **Ent-BAM** framework. Step 1 extracts entities using a prompt-based strategy. Step 2 formulates the target ANL output consisting of augmented AC and AR labels with extracted *Outcomes* entities. *Claim* and *Evidence* are marked in *Green* and *Blue* respectively, with AR labels underlined and AC labels *italicized*. Step 3 describes the proposed text-to-text fine-tuning using an encoder-decoder model.

with the help of a LLM. The second step prepares the training data in a desired format. It transforms the raw text of a clinical trial report into a *label-augmented* target ANL output, embedding both AC and AR labels. Subsequently, the (co-)occurrence information of *Outcome* entities, extracted in the first step and related to the corresponding ACs, is prepended to the ANL, forming the final target output. The third and final step fine-tunes an encoder-decoder model. Given the raw clinical trial report as input, an encoder-decoder model is then trained to generate the proposed target ANL output, which contains both augmented ACs/ARs and the *Outcome* entities. By incorporating entity information in this manner, the model enhances its conditional generation capabilities for AC and AR labels by generating additional *Outcome* entity information alongside the label-augmented output. Experimental results demonstrate improvements over recent baselines, achieving state-of-the-art (SoTA) results on the standard BAM dataset. We summarise the main contributions of this chapter as follows:

1. We propose **Ent-BAM**, an ANL-based framework for joint modelling of all four key BAM tasks: ACI, ACC, ARI and ARC, where the generation is guided by (co-)occurred *Outcome* entity information.

2. We prompt an LLM to extract (co-)occurred *Outcome* entities from the standard BAM dataset, where the explicit annotation of such entities is absent.

3. We conduct extensive experiments with ablation studies and error analyses, achieving SoTA results that highlight the strength of our approach.

TH-4046_216101004

5.2 Ent-BAM Framework

This framework is built on three key steps. The first step extracts *Outcome* entities from a standard BAM dataset using a prompt-instructed method powered by an LLM. The next step utilizes the extracted entities to create an entity-enhanced ANL. The final step is to fine-tune an encoder-decoder model for a text-to-text generation with raw clinical trials as input and the proposed entity-enhanced ANL as output. Figure 5.2 illustrates all steps of the Ent-BAM. Subsequent sections discuss each of the steps in detail.

5.2.1 Prompt-Based Entity Extraction

We implement a *Few-Shot* prompt-based extraction strategy with an open-sourced LLM, *Meta-LLaMa-3.1-instruct* to extract the *Outcome* entities (See Figure 5.2 step 1). As the dataset mainly contains cancer-related trials, we instruct the LLM to assume the role of a friendly cancer specialist tasked with analyzing research trial reports. Given a specific report and five reference examples, the LLM is instructed to extract the *Outcome* entities from it. The final output is structured in a JSON format, containing a list of the identified *Outcome* entities. The prompt emphasizes strict formatting adherence and discourages unnecessary text or explanations.

5.2.2 Entity Enhanced Target ANL Formulation and Fine-Tuning

We first introduce notations used in describing the step two of the proposed method. A paragraph W with n number of tokens is written as $W = w_1, w_2, w_3, ..., w_n$. For a text-span $w_i, w_{i+1}, w_{i+2}, \ldots, w_j$ in W, we write it as $w_{i:j}$. AC is indicated as c_p and its type as t_p. An AC span is indicated as $w_{c_p^s:c_p^e}$ with c_p^s and c_p^e indicates start and end indices of the AC c_p in W. r_{pq} indicates the AR type (e.g., Support, Attack) between the head AC c_p and the tail AC c_q. We also use a set of special symbols $S = \{[,], ((,)), |\}$ in the formatted target ANL output. "[" marks the start of an augmented label, "]" marks the end of an augmented label, "((" and "))" are used to enclose the entities, and "|" is used to separate different augmented labels in the output.

In a paragraph W, if two ACs c_p, c_q are related with r_{pq} type, and c_p as a head AC and c_q as tail AC, then for the token spans of c_p i.e. $w_{c_p^s:c_p^e}$ and c_q i.e. $w_{c_q^s:c_q^e}$, the model will generate augmented labels of this relation pairs as $[w_{c_q^s:c_q^e} \,|\, t_q \, r_{pq} \,|\, w_{c_p^s:c_p^e}]$ and for the head AC as $[w_{c_p^s:c_p^e} \,|\, t_p]$. The remaining tokens in W will be rewritten as they are. At last, the whole constructed sequence will be appended after $E =$ `Outcomes:` $((e_1, e_2, ..., e_z))$ to form final target output ANL Z.

As shown in step 2 of Figure 5.2, the first augmented label of the output text, "[*the rate of efficiency ...* | *Evidence Support* | *Intraperitoneal perfusion of verapamil ...*]" indicates that *"the rate of efficiency ..."* is an AC of type *Evidence* and it *Supports* another AC *"Intraperitoneal perfusion of verapamil ..."*, which is of type *Claim*, which is originally present at the end of the paragraph with an augmented label "[*Intraperitoneal perfusion of verapamil ...* | *Claim*]". This way, augmented labels are created for all the ACs in the paragraph. Within these AC spans, entities such as *"efficacy," "adverse effects," "quality of life,"* and *"survival rate"* are the extracted *Outcome* entities. These extracted entities are added at the beginning of the target output, forming the final proposed output.

5.2.3 Proposed Text-to-Text Fine-Tuning

Given the raw clinical trial paragraph W as input, an encoder-decoder model is fine-tuned to generate the previously constructed proposed ANL output Z to perform two tasks in a single target output: (I) Generation of label-augmented ACs and ARs, and (II) Generation of *Outcome* entities as a by-product. During the inference, the flattened ANL is post-processed to get the final ACs and ARs. We ignore the generated *Outcome* entities as they are not part of the argumentative structure.

5.3 Experimental Setup

5.3.1 Datasets

We consider the standard BAM dataset **AbstRCT** to evaluate the proposed method.

5.3.2 Implementation Details

We employ Flan-T5 [103] with four different model sizes: *Base* (220M), *Large* (770M), *XL* (3B), and *XXL* (11B). We perform all experiments using an NVIDIA A100 GPU with a maximum input/output sequence length of 1024. For the *Base* model, we use a batch size of 8, while a batch size of 16 is employed for the other model variants. The AdamW optimizer is applied with a linear learning rate decay starting from 0.0005. Results are averaged across three independent runs. Apart from the *Base* model, QLoRA adapters [110] are utilized during fine-tuning. The QLoRA configuration uses a rank (r) of 16 and a LoRA alpha of 32, with a dropout rate of 0.05 and no bias. It is set up for sequence-to-sequence language modeling (SEQ_2_SEQ_LM) and targets specific modules including query, value, key, and output layers. The model

Test Split	Baseline	Model	Micro F1 Scores of Key BAM Tasks				AVG
			ACI	ACC	ARI	ARC	
NEO-test	GMAM [18]	BART-Base	72.67	65.35	34.88	33.64	51.63
	ST [17]	Longformer	70.29	64.16	39.35	38.38	53.04
	DENIM [77]	BART-Base	77.52	70.19	41.79	40.46	57.49
	ANL-AM# [76]	Flan-T5-Base	78.71	72.29	38.66	37.14	56.70
		Flan-T5-Large	77.09	72.20	43.24	42.50	58.75
		Flan-T5-XL	78.22	72.02	43.34	42.58	59.04
		Flan-T5-XXL	78.34	72.90	47.20	45.96	61.10
	Ent-BAM (Ours)	Qwen-3-8B-Instruct	76.41	69.55	43.24	41.08	57.57
		LLaMa-3.1-8B-Instruct	70.90	64.62	39.72	36.75	52.99
		Flan-T5-Base	79.28	73.81	43.19	41.26	59.38
		Flan-T5-Large	79.77	74.58	45.54	44.21	61.02
		Flan-T5-XL	80.59	75.49	52.68	49.83	64.64
		Flan-T5-XXL	**81.67**	**76.04**	**54.56**	**51.57**	**65.96** (+4.86)
GLAU-test	ST* [17]	Longformer	59.36	55.58	17.89	17.19	37.51
	DENIM* [77]	BART-Base	78.14	72.75	42.37	41.52	58.70
	ANL-AM# [76]	Flan-T5-Base	79.64	73.71	37.76	36.75	56.96
		Flan-T5-Large	81.41	76.31	42.05	41.79	60.39
		Flan-T5-XL	82.56	76.88	45.64	44.87	62.48
		Flan-T5-XXL	81.87	75.72	49.34	48.81	63.93
	Ent-BAM (Ours)	Qwen-3-8B-Instruct	79.02	72.31	44.92	42.74	59.74
		LLaMa-3.1-8B-Instruct	75.46	69.41	42.56	41.60	57.25
		Flan-T5-Base	80.27	74.66	40.19	39.71	58.70
		Flan-T5-Large	80.79	75.12	41.50	40.35	59.44
		Flan-T5-XL	82.29	**77.81**	**50.73**	**49.47**	**65.07**(+1.14)
		Flan-T5-XXL	**82.88**	77.37	50.47	48.66	64.84
MIX-test	ST* [17]	Longformer	60.69	56.34	23.05	22.39	40.62
	DENIM* [77]	BART-Base	75.24	68.68	39.28	37.79	55.25
	ANL-AM# [76]	Flan-T5-Base	74.89	68.22	40.22	37.97	55.32
		Flan-T5-Large	74.43	70.60	40.90	40.62	56.63
		Flan-T5-XL	76.66	71.44	42.02	41.13	57.81
		Flan-T5-XXL	78.43	73.23	43.69	43.40	59.68
	Ent-BAM (Ours)	Qwen-3-8B-Instruct	75.10	68.30	40.00	39.03	55.60
		LLaMa-3.1-8B-Instruct	72.60	67.63	40.33	39.33	54.97
		Flan-T5-Base	77.03	70.59	41.00	39.16	56.94
		Flan-T5-Large	77.91	72.77	41.33	40.15	58.04
		Flan-T5-XL	78.40	**74.09**	47.48	45.42	61.34
		Flan-T5-XXL	**79.33**	73.70	**49.15**	**46.86**	**62.26** (+2.58)

Table 5.1 Comparison of **Ent-BAM** against the baselines in terms of *Micro F1* scores across all test splits of the `AbstRCT` dataset. The highest scores are in **Bold**, and improvements over SoTA are highlighted within brackets in **Green**. Scores marked with * are reproduced using public code from the respective baselines; those marked with # are from our reimplementation due to unavailable source code. In the AVG column, underlined scores compare **Ent-BAM** with the most recent SoTA model of equivalent scale across each test set.

loads in 4-bit precision (load_in_4bit) with NF4 quantization (bnb_4bit_quant_type), double quantization enabled (bnb_4bit_use_double_quant), and computes using the torch.bfloat16 data type (bnb_4bit_compute_dtype). In addition to the Flan-T5 model family, we also report results using two recent instruction-tuned decoder-only models: **LLaMa-3.1-8B-Instruct**[1] and **Qwen-3-8B-Instruct**[2], to ensure the generalisation ability of our method across models. These are fine-tuned via the `Unsloth`[3] framework with the similar QLoRA configuration as described above.

5.3.3 Prompt-Based Entity Extraction

We extract the *Outcome* entities solely from the *NEO-train* dataset using a *Few-Shot* prompt-based extraction strategy. The dev and all three test splits are not used. These entities are only required in the training process to enhance the model. During this fine-tuning step of the model Flan-T5, the generation of these entities, as an auxiliary task, helps the model to enrich the conditional probabilities of the augmented ACs and ARs.

5.3.4 Baselines and Evaluation

We compare our method with the following SoTA baselines. **GMAM** [18] is a generative model using positional encoding and a pointer mechanism to solve all key BAM tasks. **ST** [17] enhances task sharing with *Longformer* and *Biaffine Attention*, which solves all key BAM tasks. **ANL-AM** [76] applies a text-to-text ANL strategy with *Flan-T5*, which originally solves ACC and ARC only. **DENIM** [77] uses discourse-aware prefixes and multi-task prompting to solve all key BAM tasks.

Most baselines report results on the *NEO-test* set only. To ensure a thorough comparison, we ran the latest strong baselines, ST and DENIM, on the *GLAU-test* and *MIX-test* sets using their open-source code and original hyperparameters. For ANL-AM, the authors' code is unavailable, so we reimplemented it using the reported hyperparameters for all three test splits. As a result, our scores for the ACC and ARC tasks on the *NEO-test* slightly differed from those reported in the original paper, which only provides results for these two tasks. We report the results of all four tasks with our implementation in Table 5.1. Following prior work [17], we evaluate all four BAM tasks using the *Micro-F1* metric. An AC prediction is counted as correct only when its span exactly matches the gold annotation. Partial span overlaps are not considered, ensuring strict consistency with earlier evaluation settings.

[1] https://github.com/meta-llama/llama3
[2] https://github.com/QwenLM/Qwen3
[3] https://unsloth.ai/

5.4 Results and Discussion

5.4.1 Main Results

Table 5.1 presents the comprehensive performance comparison of the proposed Ent-BAM framework with the latest SoTA baselines across all test splits of the `AbstRCT` dataset. As the Ent-BAM framework addresses all four key BAM tasks, we compute the average performance to compare those tasks in one place, similar to the studies in [77]. The results demonstrate that Ent-BAM consistently surpasses all existing baselines, establishing new SoTA results across all test splits. Specifically, in the *NEO-test*, *GLAU-test*, and *MIX-test* splits, our method outperforms the average Micro F1 scores of the most recent SoTA baseline DENIM by significant margins of 4.86%, 1.14%, and 2.58%, respectively. These improvements across different diseases underscore the cross-disease generalizability of our approach, even when trained exclusively on the *NEO-train* set. We also observe that, compared to the non-generative baseline ST, the generative baselines DENIM and ANL-AM, along with the proposed Ent-BAM, consistently outperform in solving BAM tasks across all test splits. This highlights the advantages of adopting a generative approach for BAM. However, the GMAM baseline struggles to effectively model relational dependencies, leading to weaker performance in the ARI and ARC tasks. In contrast, Ent-BAM excels in handling relational dependencies, delivering robust performance across all tasks, including ARI and ARC.

Upon analysing the performance of decoder-only models, namely *Qwen-3-8B-Instruct* and *LLaMa-3.1-8B-Instruct*, the results suggest that these models demonstrate competitive performance compared to earlier baselines such as ST and GMAM. However, their effectiveness lags behind that of the Flan-T5 variants of similar or smaller scale. This is evident across all test splits and especially pronounced in the ARI and ARC tasks, where relational reasoning is crucial. For instance, on the *NEO-test* split, Qwen achieves an ARC score of 41.08 compared to 51.57 with Flan-T5-XXL, and LLaMa attains only 36.75 in ARI, reflecting limitations in modelling the argument relations. This suggests that, despite their strong general-purpose capabilities, decoder-only models are less suited to BAM tasks within the current setup. On the other hand, with the increase in model parameters of Flan-T5, the performance of Ent-BAM improves. Especially for the models with higher parameters, Ent-BAM captures the relations dependencies very well. For instance, in the comparatively complex ARC task, the performance improvements from Base to XXL across the *NEO-test*, *GLAU-test*, and *MIX-test* splits are substantial: 10.31%, 8.95%, and 7.7%, respectively. Whereas the improvements for ACC tasks are 2.23%, 2.71%, and 3.11%, respectively. This suggests the utility of using larger encoder-decoder models in mod-

TH-4046_216101004

Dataset	Model	ACI	ACC	ARI	ARC	AVG
NEO-test	Ent-BAM	79.28	73.81	43.19	41.26	**59.38**
	Ent-BAM(-OE)	78.19	71.38	41.42	39.32	57.57 (-1.81)
GLAU-test	Ent-BAM	80.27	74.66	40.19	39.71	**58.70**
	Ent-BAM(-OE)	80.07	75.06	40.01	39.36	58.62 (-0.08)
MIX-test	Ent-BAM	77.03	70.59	41.00	39.16	**56.94**
	Ent-BAM(-OE)	76.68	70.67	39.21	37.28	55.96 (-0.98)

Table 5.2 Results of ablation experiments across all test splits in *Micro F1* Scores. Ent-BAM(-OE) is without the list of *Outcome* entities.

elling the argumentative structural dependencies, making them especially suitable for BAM tasks requiring complex relational reasoning.

5.4.2 Ablation Study

We conduct an ablation study by removing *Outcome* entities from the ANL. We refer to this variant as Ent-BAM(-OE). It uses the *Flan-T5-Base* model. Table 5.2 shows the average performance across all BAM tasks in Ent-BAM(-OE) consistently lagged behind the original Ent-BAM variant across all test splits. The drop is more pronounced in *NEO-test* and *MIX-test* splits, emphasizing the *positive impact* of integrating ***Outcome Entity (Co-)Occurrence information*** in Ent-BAM. When the model generates *Outcome* entities at the beginning of the sequence, it implicitly provides some guidance for identifying and classifying ACs and ARs. Interestingly, the Ent-BAM(-OE) still outperforms the recent SoTA baseline DENIM in *NEO-test* and *MIX-test* splits, highlighting the effectiveness of the ***ANL-based formulation*** for jointly addressing all BAM tasks.

Additionally, we perform another ablation study to assess the utility of the proposed target ANL format (repeated AC spans for relation aggregation). We design an *Abbreviated* format (see Figure 5.3) that replaces repeated AC spans with *ID-token* identifiers. For a paragraph containing *n* Evidences and *m* Claims, we assign tokens such as *E1, E2, ...* for Evidences and *C1, C2, ...* for Claims, assigned in order of appearance. These tokens are used consistently during AR-type assignment, replacing repeated AC spans. This approach reduces output length and preserves relational links via token references. However, results using *Flan-T5-Base* (Table 5.3) show a consistent performance drop across tasks and splits. The added indirection appears to hinder the accurate localisation of ACs and their relations. This suggests a trade-off between sequence compression and performance, highlighting the benefit of the original span-based relational representation in this generative AM modelling.

Proposed Target Output

Outcomes: ((remission rate, disease control, quality of life, response rate, progression etc.))

[Responses (> or = 50% improvement) were seen in 3 (20%) of 15 patients with an initial dose at 6.5 mg/m(2) per day (95% confidence interval [CI], 0%-40%), 15 (54%) of 28 patients at 300 mg/m(2) per day (95% CI, 35%-72%), and 10 (67%) of 15 patients at above 300 mg/m(2) per day (95% CI, 43%-91%). | **Evidence** Support | Bexarotene (Targretin capsules) (the first retinoid X ..... doses of 300 mg/m(2) per day.] [The rate of progressive disease was 47%, 21%, and 13% at the same dose levels, respectively. | **Evidence** Support | Bexarotene (Targretin capsules) (the first retinoid X ..... doses of 300 mg/m(2) per day.] [Eight (73%) of 11 patients crossing over from 6.5 mg/m(2) per day to higher doses subsequently responded. | **Evidence** Support | Bexarotene (Targretin capsules) (the first retinoid X ..... doses of 300 mg/m(2) per day.] [Bexarotene (Targretin capsules) (the first retinoid X ..... doses of 300 mg/m(2) per day. | **Claim**]

Abbreviated Target Output

Outcomes: ((remission rate, disease control, quality of life, response rate, progression etc.))

[Responses (> or = 50% improvement) were seen in 3 (20%) of 15 patients with an initial dose at 6.5 mg/m(2) per day (95% confidence interval [CI], 0%-40%), 15 (54%) of 28 patients at 300 mg/m(2) per day (95% CI, 35%-72%), and 10 (67%) of 15 patients at above 300 mg/m(2) per day (95% CI, 43%-91%). | **E1** Support | **C1**] [The rate of progressive disease was 47%, 21%, and 13% at the same dose levels, respectively. | **E2** Support | **C1**] [Eight (73%) of 11 patients crossing over from 6.5 mg/m(2) per day to higher doses subsequently responded. | **E3** Support | **C1**] [Bexarotene (Targretin capsules) (the first retinoid X ..... doses of 300 mg/m(2) per day. | **C1**]

Fig. 5.3 Representation of Abbreviated ANL output using “ID-tokens" to replace repeated AC spans.

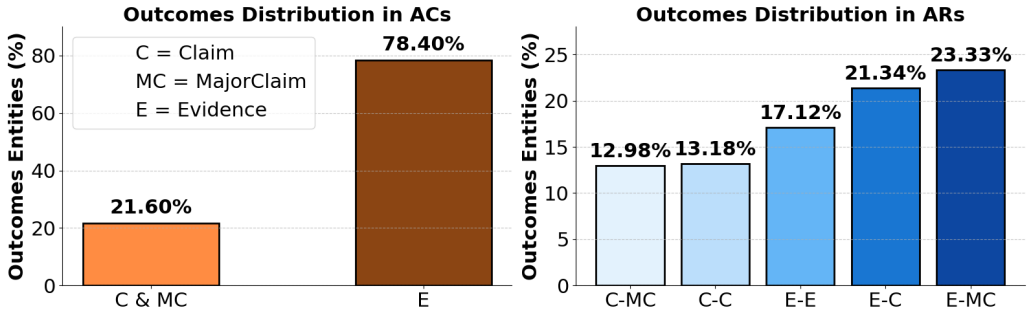

Fig. 5.4 The left figure illustrates the percentage occurrences of *Outcome* entities across different ACs in the *NEO-train* split, while the right figure presents their distribution within various related pairs of ACs.

Dataset	Model	ACI	ACC	ARI	ARC	AVG
NEO-test	Ent-BAM	79.28	73.81	43.19	41.26	**59.39**
	Ent-BAM*(Abbr)*	67.94	59.91	32.65	30.61	47.78 (-11.61)
GLAU-test	Ent-BAM	80.27	74.66	40.19	39.71	**58.71**
	Ent-BAM*(Abbr)*	71.45	64.08	35.49	34.08	51.78 (-6.93)
MIX-test	Ent-BAM	77.03	70.59	41.00	39.16	**56.95**
	Ent-BAM*(Abbr)*	67.16	58.39	34.55	32.11	48.55 (-8.40)

Table 5.3 Comparison between Ent-BAM and its *Abbreviated Output Variant* across all test splits *(Using Flan-T5-Base)*.

Dataset	Model	ACI	ACC	ARI	ARC	AVG
NEO-test	Ent-BAM	79.28	73.81	43.19	41.26	**59.39**
	with Zero-Shot	77.74	72.20	39.89	37.73	56.39 (-3.00)
	with TransforMED	76.69	69.63	40.10	37.98	56.10 (-3.29)
GLAU-test	Ent-BAM	80.27	74.66	40.19	39.71	58.71
	with Zero-Shot	78.08	72.71	39.75	39.44	57.00 (-1.71)
	with TransforMED	78.45	73.02	42.46	41.53	**58.87** (+0.16)
MIX-test	Ent-BAM	77.03	70.59	41.00	39.16	**56.95**
	with Zero-Shot	76.25	70.29	39.11	37.22	55.22 (-1.73)
	with TransforMED	72.89	66.71	36.67	34.82	52.77 (-4.18)

Table 5.4 Comparison of Ent-BAM *(Few-Shot)* with *Zero-Shot*, and *TransforMED* extraction approaches using Flan-T5-Base.

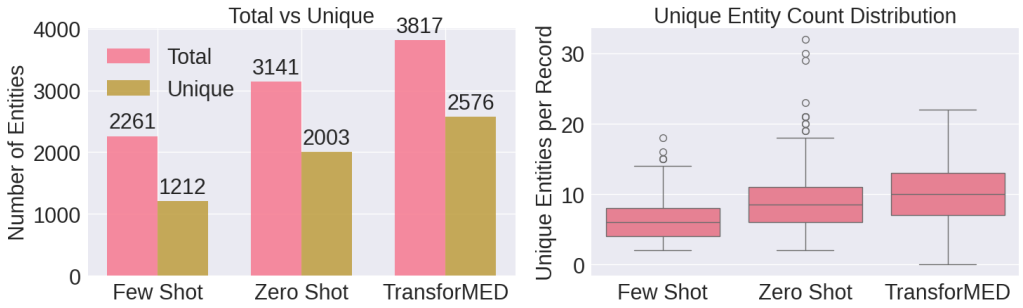

Fig. 5.5 Comparison of *Outcome* entity counts, and distributions across extraction methods, revealing redundancy and noise in *Zero-Shot* and *TransforMED* outputs.

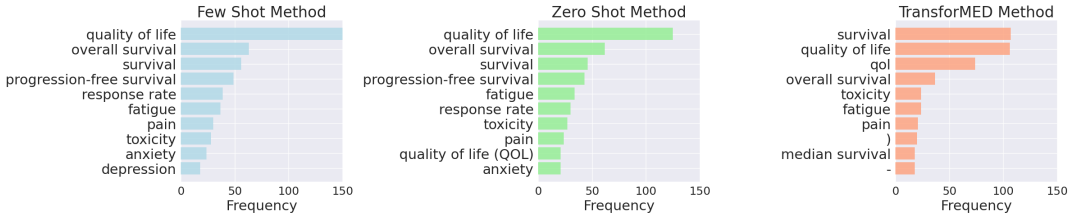

Fig. 5.6 Statistics of predominantly occurring *Outcome* entities in the *NEO-train* split using different methods of extraction.

5.4.3 Analysis and Evaluation of the Extracted *Outcome* entities

Figure 5.4 shows that *Outcome* entities appear far more frequently in *Evidence* ACs than in *Claims* or *MajorClaims*. Among AR pairs, they are most common in *Evidence-MajorClaim*, while *Claim-MajorClaim* shows fewer occurrences. As noted by [83], current biomedical datasets often lack cancer-specific PICO elements like “Quality of Life" and “Survival Rate," making direct quality comparisons difficult. To address this, we conducted a human evaluation of 10% of the training data. Two annotators reviewed the extracted entities, achieving a Krippendorff’s Alpha of 0.69 and an Inter-Annotator Agreement of 94.47%, indicating strong reliability. This suggests that our silver-standard entities are of good quality and beneficial to Ent-BAM’s performance. While some entities may be missed due to prompt-based limitations, this analysis focused solely on evaluating the quality of those extracted.

To assess the effectiveness of our proposed *Few-Shot Outcome* extraction method, we compare it with two alternative strategies: (i) a *Zero-Shot* variant, which follows the same pipeline but omits examples in the prompt; and (ii) *TransforMED*, a recent traditional BIO-tagging approach that uses an encoder-based model trained on biomedical texts to extract PICO elements [111]. In both cases, the extracted *Outcome* entities are integrated into the final ANL target output of Ent-BAM. Both models are fine-tuned using the *Flan-T5-Base* backbone, and the results are summarised in Table 5.4. Across *NEO-test* and *MIX-test*, Ent-BAM with *Few-Shot* extraction achieves the highest performance, demonstrating the robustness of our method. Although *TransforMED* slightly outperforms it on *GLAU-test*, the overall results highlight the reliability and generalizability of the proposed *Few-Shot* approach for extracting *Outcome* entities over other approaches. Figure 5.5 further shows that *Zero-Shot* and *TransforMED* produce more number of unique entities due to redundancy or noise. This makes it more difficult for the downstream model to understand the extracted information, ultimately reducing performance. It is illustrated in Figure 5.6, where *Few-Shot* yields clean, well-normalized entities, while *Zero-Shot* method includes duplicates such as *quality of life"* and *quality of life (QOL)"*. Similarly, *TransforMED* also yields both full and abbreviated forms *(e.g., “qol")* of it, as well as irrelevant tokens like *“)"* and “-".

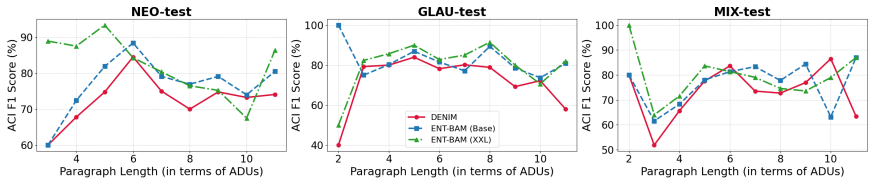

Fig. 5.7 Comparison of ACI Micro F1 Scores (%) between DENIM and Ent-BAM (Base & XXL) across paragraphs of varying lengths (in terms of the number of ADUs present in that paragraph) for different test splits of `AbstRCT` dataset.

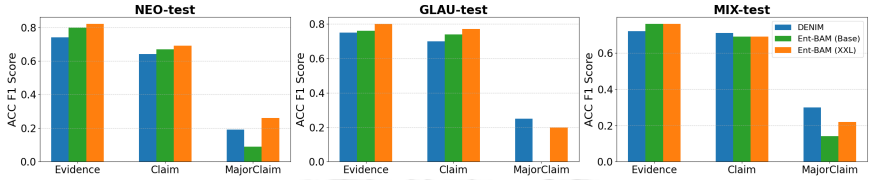

Fig. 5.8 Performance Comparison of Micro F1 Scores for DENIM, Ent-BAM (Base), and Ent-BAM (XXL) on the ACC Task across different splits of the `AbstRCT` Dataset.

5.4.4 Performance Analysis of All Four BAM Tasks

We conduct a comprehensive comparison of Ent-BAM (Base and XXL) against the SoTA baseline DENIM across all four BAM tasks: ACI, ACC, ARI, and ARC in the subsequent sections to discuss the trends over paragraph lengths, relation distances, and class-wise performances of ACs and ARs.

ACI Task for Varied Length Paragraphs: Figure 5.7 compares the ACI performance of Ent-BAM (Base & XXL) against DENIM across varying paragraph lengths in three test splits. Both Base & XXL variants consistently outperform DENIM, with the XXL model achieving peak F1 scores around 90% on moderate-length paragraphs (e.g., 5–6 ADUs) in the NEO and GLAU splits. Ent-BAM (Base) also maintains strong performance, often within 5–10 points of the XXL variant, while significantly outperforming DENIM across all length ranges. DENIM, in contrast, exhibits sharp drops in shorter (2–4 ADUs) and longer (10–11 ADUs) paragraphs, particularly in the GLAU and MIX splits. Thus, Ent-BAM demonstrates stability and generalizability across all test splits, confirming its robustness across diseases and length variations.

Classwise ACC Task Performance Comparison: Figure 5.8 compares the performance of the three models: DENIM, Ent-BAM (Base), and Ent-BAM (XXL) on the ACC task. Ent-BAM (XXL) consistently achieves the best Micro F1 scores across all dataset splits, followed closely by Ent-BAM (Base), while DENIM lags slightly behind but remains competitive. A similar pattern is observed in the *Claim* category also, where Ent-BAM (XXL) and Ent-BAM (Base) perform closely, with a slight advantage for the larger model. DENIM trails behind the Ent-BAM models but maintains reasonable performance across all dataset splits. In contrast, *MajorClaim* extraction

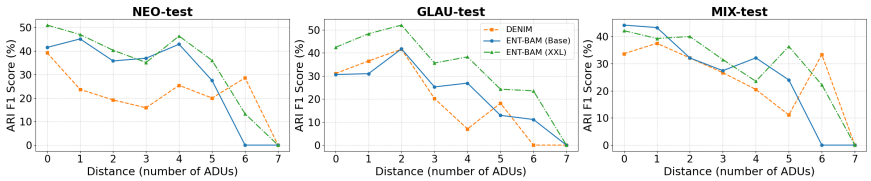

Fig. 5.9 Performance analysis of ARI task Micro F1 Scores (%) for DENIM and Ent-BAM (Base & XXL) across varying distances (in terms of number of intermediate ADUs) in different test splits of `AbstRCT` dataset.

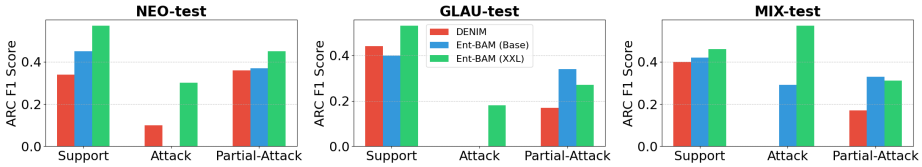

Fig. 5.10 Comparison of Micro F1 Scores of ARC task of different AR classes across different splits of the `AbstRCT` dataset, highlighting the performance of Ent-BAM (Base & XXL) and DENIM.

poses a significant challenge for all models, largely due to the scarcity of training examples and the resulting imbalance in class distribution.

ARI Task for Varied Distance Relations: In argumentative paragraphs, the distance between related ACs varies from zero ADUs to a few ADUs. Some related ACs are directly linked with no intermediate components (0 intermediate ADUs), forming straightforward connections, while others are moderately far apart, involving a few intermediate ADUs. The most challenging cases arise when one ADU appears near the beginning of the paragraph and the other near the end. Identifying such varied-distant relations is critical for understanding complex argumentative structures. As shown in Figure 5.9, Ent-BAM (Base and XXL) consistently outperforms DENIM across all splits. In the NEO-test split, Ent-BAM (XXL) maintains strong performance for close-range links (0–2 ADUs apart), while Base remains robust up to (4-5 ADUs apart), clearly outperforming DENIM throughout. In the GLAU-test split, Ent-BAM (XXL) shows resilience even as distance increases, sustaining over 45% F1 at 2–3 ADUs,

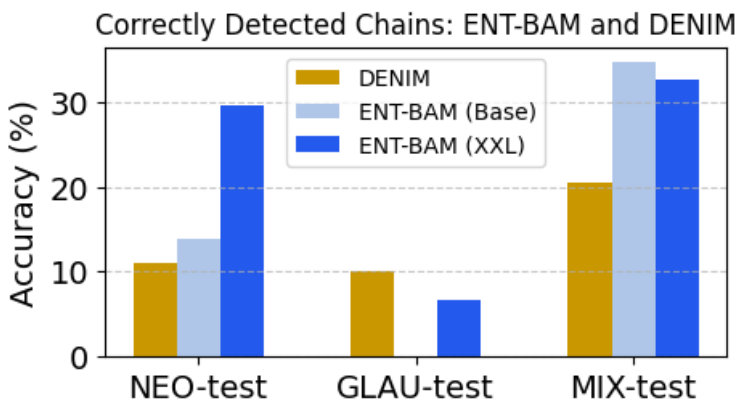

Fig. 5.11 Accuracy percentage of DENIM and Ent-BAM (Base & XXL) on relation chain detection across different test splits of `AbstRCT` dataset.

whereas both DENIM and Ent-BAM (Base) drop sharply beyond short distances. Similarly, in the MIX-test split, Ent-BAM variants handle both short and moderate distances more effectively than DENIM. These results highlight Ent-BAM's ability to model both local and long-range argumentative links.

ARC Task for Different AR Classes: Figure 5.10 compares the performance of two Ent-BAM variants and DENIM on the ARC task. In the *NEO-test* split, Ent-BAM (XXL) achieves the highest Micro F1 Scores for all AR classes. Although Ent-BAM variants gave better performance than DENIM on the *Attack* and *Partial-Attack* relations, all models found classifying *support* relation easier than the other two AR classes. Relatively poor performance could be due to the imbalance in AR class distribution. The superior performance of Ent-BAM (XXL) across all relation types, including the imbalanced *Attack & Partial-Attack* AR classes, indicates that larger models are better equipped to handle the class imbalanced splits more efficiently.

5.4.5 Performance Analysis of Relational Chain of Reasoning

Modelling the relational dependencies of an argumentative structure has always been a challenge in BAM literature. As pointed out by Liu *et. al.*[63], the structure of an argumentative text can be regarded as an answer to a *why* question. Therefore, the entire argument structure resembles a chain of reasoning, outlining a sequence of steps *(a path constructed by a series of consecutive premises/claims)* leading to a specific conclusion. Therefore, properly handling these *chains of reasoning* adds complexity to efficiently modelling the relational structure. In order to analyze this *chain of reasoning*, we define a chain like this: Consider an argumentative paragraph comprising multiple ACs, denoted as C_1, C_2, C_3, C_4, and C_5. These ACs may form a reasoning chain such as $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5$, with C_5 as the root AC, where each AC is directly connected to the preceding one. In such configurations, identifying relations between adjacent ACs poses a significant challenge. This difficulty arises because all ACs in the chain are (in-)directly connected through the root AC, increasing the likelihood of misidentifying relations. For example, one might incorrectly predict $C_3 \rightarrow C_5$ instead of the correct $C_4 \rightarrow C_5$. In the following paragraphs, we analyze and compare the performance of identifying these chains in both Ent-BAM and DENIM frameworks for all splits of the *AbstRCT* dataset. Only exact chain matches are considered correct, excluding any partial matches.

As shown in Fig. 5.11, in the *NEO-test* split, Ent-BAM (XXL) significantly outperformed both the DENIM and Ent-BAM (Base), achieving a correctly predicted chain accuracy of 29.62%, compared to 11.11% for DENIM and 13.88% for Ent-BAM (base) model. This underscores the efficacy of the larger model in capturing complex argumentative structures in biomedical contexts. For *GLAU-test* split, all models

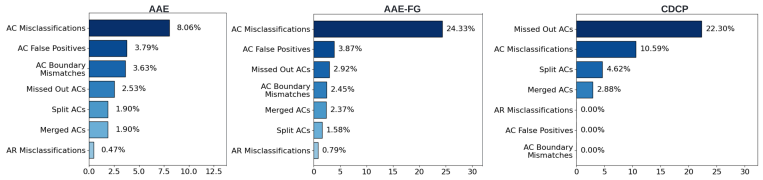

Fig. 5.12 Error distribution across different test splits of the AbstRCT dataset for the Ent-BAM (Base) framework, highlighting the percentage contribution of various error categories.

showed relatively lower accuracy, with DENIM at 10.00% and Ent-BAM (XXL) at 6.66%. The poor performance of Ent-BAM (base) suggests that smaller models may entirely fail to capture certain relationships, leading to zero correct predictions. For the *MIX-test* split, Ent-BAM (base) demonstrates the superior chain-detection ability, achieving the highest accuracy of 34.69% and Ent-BAM (XXL) closely following at 32.65%. In contrast, DENIM achieved only 20.40%, further emphasizing its limited capacity in detecting chains for diverse diseases.

5.4.6 Error Analysis

We performed error analysis on all three test splits and observed the following errors with Ent-BAM (Base): *(i) Missing Brackets* refers to the erroneous enclosing of augmented labels, where the brackets are opened, but not closed properly or the closing brackets are missing or multiple closing brackets are generated; *(ii) AC Boundary Mismatches*, where predicted ACs do not match the ground truth, either being too short or too long; *(iii) Missed Out ACs* means, potential ACs remain undetected during the inference; *(iv) AC False Positives* implies that the identified ACs are not present in the set of true ACs; *(v) AC Misclassifications* tells how many number of ACs are wrongly classified; *(vi) AR Misclassifications* are wrongly classified ARs; *(vii) Split ACs* refers to the number of ACs are split into multiple smaller AC segments; *(viii) Merged ACs* are just opposite to Split ACs, where two or more true ACs are incorrectly merged into a bigger ACs in the prediction; and *(ix) Invalid Tokens* refers to the generated ANL consists of some out-of-vocabulary tokens or out-of-context text spans.

Figure 5.12 shows the error distribution across the three test splits of the AbstRCT dataset. *Missing Brackets* is the most common error, especially in *GLAU-test* (16.00%), followed by *NEO-test* (10.00%) and *MIX-test* (9.00%). In *NEO-test*, both *Missed Out ACs* (8.08%) and *AC False Positives* (7.64%) occur. This shows that our system is highly sensitive. It can detect many valid ACs but sometimes selects extra text spans. For the *MIX-test* split, *AC False Positives* (8.91%) are the most frequent. *AC Boundary Mismatches* appear consistently, ranging from 4.99% to 7.11%. This suggests that the models sometimes miss the exact start or end of an annotation. Despite this, our overall performance is strong, and these challenges provide clear directions for future

improvements. *AC Misclassifications* are slightly higher in *MIX-test* (5.21%) than in the other domains. Less common issues such as *AR Misclassifications*, *Split ACs*, and *Merged ACs* are also present. These rare errors highlight some domain-specific challenges that we are eager to address in our next development phase.

5.5 Conclusion and Discussion

We introduce Ent-BAM, an ANL-based framework for jointly modeling all four key BAM tasks. This work addresses the influence of PICO elements, specifically the *Outcome* entities in solving BAM tasks. By identifying the entity (co-)occurrence phenomenon inside arguments, this work models the relational dependencies of argumentative structures in an efficient way to include the biomedical-specific information in a text-to-text generation model to efficiently guide the generation process. Through a same target generation sequence, two tasks are performed, first the generation of augmented ACs/ARs and, second, an auxiliary task to generate the *Outcome* entities alongside. By utilizing the *Few-Shot* capability of LLM, Ent-BAM addresses the absence of annotated *Outcome* entities in standard BAM dataset. Since the LLM output is not fully controllable, it may sometimes produce noisy or incomplete results. Thus in our case, the format of the *Few-Shot* prompt choice is experimental. Upon performing several extraction prompting experiments, we chose the best prompt that provided us with good extraction results (Exact prompt description is provided in the supplementary sheet). However, using the extracted entities from poorly constructed prompts might lead to inconsistent results in all BAM tasks, with an increasing chance of error propagation. Human evaluation and comparative experiments with two alternative extraction strategies indicate that the quality of the *Few-Shot* extracted silver-standard entities is sufficient to improve overall task performance. Through extensive experiments and ablation study, Ent-BAM demonstrates its strong capability, setting SoTA results for all test splits of the `AbstRCT` dataset.

TH-4046_216101004

Chapter 6

End-to-End Argument Mining through Autoregressive Argumentative Structure Prediction

Chapter Highlights

- This chapter addresses a core limitation of previous ANL-based generative approaches, which **flatten** complex argumentative structures into linear text sequences, often leading to repeated redundant AC spans and inefficiency when connecting the related AC pairs.
- **Autoregressive Argumentative Structure Prediction (AASP):** An end-to-end framework is introduced that moves beyond text generation. It models argument mining as the prediction of a sequence of **structure-building actions**, constructing the argumentative structure step-by-step.
- **Action-Based Formulation:** The framework uses a conditional language model to autoregressively predict a constrained set of actions: (i) **Span-Identifying**, (ii) **Boundary-Pairing**, and (iii) **Type-Labeling** to explicitly capture the dependencies and reasoning flow within the argument.
- The framework is evaluated on all four key AM tasks (**ACI, ACC, ARI, ARC**) across all standard benchmarks. The AASP framework achieves new SoTA results, which is currently the highest performing framework in the literature.
- The publication related to this chapter is as follows:
  1. **Nilmadhab Das**, Vishal Vaibhav, Yash Sunil Choudhary, V. Vijaya Saradhi, Ashish Anand, "End-to-End Argument Mining through Autoregressive Argumentative Structure Prediction," in *Proceedings of the International Joint Conference on Neural Networks (**IJCNN**)*, Italy, 2025.

TH-4046_216101004

6.1 Introduction

The preceding chapters have systematically explored the enhancement of generative frameworks for AM. Chapter 3 introduced surface-level linguistic signals, such as discourse and argumentative markers, which can effectively guide a text-to-text model in identifying argumentative structures. Subsequently, Chapter 4 addressed the limitations of relying solely on external cues by delving into the internal semantics of arguments. It introduced a method for modeling the conceptual links between components using "Related Key Phrases," thereby enriching the generative output with explicit semantic information for the enhanced relation modeling.

While these approaches have progressively advanced the SoTA, they share a fundamental methodological limitation inherent to their text-to-text formulation. Both frameworks are constrained to represent the intrinsically hierarchical or graph-like nature of argumentation by flattening it into a linear text sequence. This process is often inefficient and redundant, requiring the model to repeatedly generate the full text of an AC simply to establish its relationship with another. For complex arguments with long chains of reasoning or multiple inter-dependencies, this text-flattening approach becomes cumbersome and may struggle to capture the holistic structure effectively.

This chapter proposes a paradigm shift to overcome this core representational bottleneck. Instead of framing AM as a task of generating an augmented text string, we reformulate it as a process of direct structure construction. This approach moves beyond the limitations of flattened sequences by modeling the argument's structure as it is, a graph of interconnected components.

To achieve this, we adapt a recently introduced structured prediction framework called the *Autoregressive Structured Prediction (ASP) framework* [112]. It has achieved strong performance in several structured prediction tasks like NER, co-reference resolution, and relation extraction. It does so through a constrained set of structure-building actions with a conditional language model. The ASP framework has been tested on foundational NLP tasks only. This framework, though achieved convincing performance, has not been applied in complex reasoning tasks such as AM. ASP framework is applicable in AM tasks as the standard datasets of AM are having tree or graph structures. We note the following differences between foundational structured prediction tasks and AM tasks: **(I)** The target output spans in these foundational tasks are relatively shorter. In contrast, AM tasks involve longer spans, where each AC often spans across extended sections of text. **(II)** In relation extraction or co-reference resolution, the distance between two related tokens (or entities) is typically smaller. On the other hand, AM tasks often involve related arguments that can span large distances. For example, one argument may be located at the beginning of a paragraph, while the other is at the end, with multiple intermediate ACs. These distinctions make it a

TH-4046_216101004

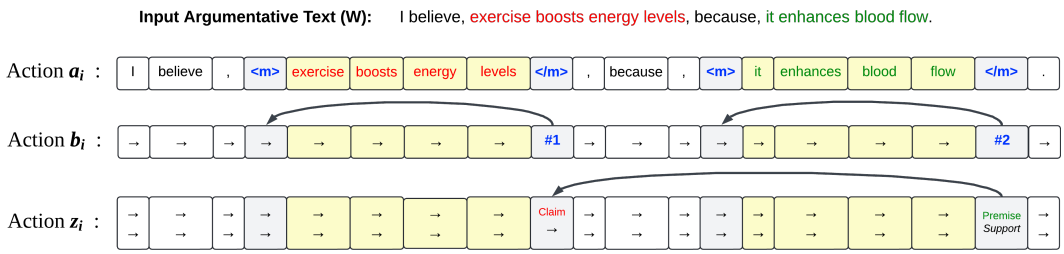

Fig. 6.1 Illustration of the decoding process in the AASP framework for end-to-end argument mining. The process involves three main actions step-by-step: (1) Action a_i constructs the target sequence for span-identifying actions; (2) Action b_i finalizes the AC span boundaries by matching the <m> and </m> tokens; and (3) Action z_i connects related ACs by linking their </m> tokens and classifies both the AC types and the types of their ARs.

meaningful and promising avenue to investigate the capability of the ASP framework in addressing AM tasks.

Thus, in this chapter, we propose to use the ASP framework in solving all four key AM tasks and demonstrate its success for the ***End-to-End*** AM framework. We name it as ***Autoregressive Argumentative Structure Prediction (AASP)***. To accommodate longer spans and distantly related ACs, we modify the feed-forward networks of the original ASP framework to a larger size. This modification is the key component in adaption the ASP framework to handle varying argumentative relational structures present in different AM corpora. Through AASP, we exploit the autoregressive phenomenon of a conditional language model for the AM task-specific constrained set of actions, including *(i) Span-Identifying Actions*, which involve the identification of AC spans; *(ii) Boundary-Pairing Actions*, which finalize the span boundaries; and *(iii) Type-Labeling Actions*, which classify AC types, connect related ACs, and determine their AR types.

Training a conditional language model to predict these structure-building actions exposes the argumentative structure such a way that it allows the model to capture the dependencies within the ACs better. This explicit modelling contributes to the learning of intra-structure relationships between ACs by utilizing the semantic and contextual knowledge of the language model. Experimenting with three standard structurally distinct AM benchmarks, AASP achieves SoTA results in two benchmarks and produces higher Micro-F1 scores on relational AM tasks in the rest of the benchmark. In summary, the key contributions of this chapter are:

1. We propose *Autoregressive Argumentative Structure Prediction* **(AASP)**, an end-to-end framework that jointly solves all four key tasks of AM using a conditional language model.

2. By adapting the ASP framework for the AM tasks, we solve the ACI task using *Span-Identifying* and *Boundary-Pairing* actions, and the ACC, ARI and ARC tasks using the *Type-Labeling* actions.

TH-4046_216101004

3. Utilizing larger FFNs to execute all these actions enhances the ability to handle longer AC spans and distantly related ACs, leading to performance improvements, as demonstrated in the *Ablation Study*.

4. We conduct extensive experiments across three diverse AM benchmarks, demonstrating the robust performance of AASP in all four key AM tasks. Our results further highlight that AASP handles relational dependencies better, thereby capturing the *chain of reasoning* more effectively as a byproduct.

6.2 Proposed Method

We define the argumentative paragraph as, $W = w_1, w_2, w_3, \ldots, w_n$, where n is the total number of tokens in W. We denote a text span from w_i to w_j in W as $w_{i:j}$. Among the four key AM tasks, *ACI* aims to identify the set of argumentative spans, $C_{ACI} = \{(s_i, e_i)\}_{i=1}^{n}$, where each span $w_{s_i:e_i}$ corresponds to a potential AC. Second, *ACC* assigns a type t_i (e.g., Premise, Claim) to each identified span $(s_i, e_i) \in C_{ACI}$, resulting in $C_{ACC} = \{(t_i, s_i, e_i)\}_{i=1}^{n}$. Third, *ARI* detects the set of relations $R_{ARI} = \{(h_j, t_j)\}_{j=1}^{m}$, where (h_j, t_j) represents a connection between two ACs, with h_j denoting the head AC (the source of the relation) and t_j denoting the tail AC (the target of the relation), where $h_j, t_j \in C_{ACC}$. Finally, *ARC* categorizes each identified relation $(h_j, t_j) \in R_{ARI}$ into a specific type r_j (e.g., Support, Attack), yielding $R_{ARC} = \{(h_j, t_j, r_j)\}_{j=1}^{m}$.

Through the *AASP framework*, we reformulate all these AM tasks as the prediction of a sequence of actions, where we construct the argumentative structure step-by-step with a conditional language model as given in Fig. 6.1. Building on the original formulation proposed in [112], ASP formulation is adapted upon the four key tasks of AM for *(i) Construction of action sequences* and *(ii) Conditional modelling of actions*. We discuss the details in subsequent sections to maintain the coherency.

6.2.1 Construction of Action Sequences

Given W, the goal is to predict a sequence of tuples $\langle span_i, boundary_i, type\text{-}label_i \rangle$.

Tuples denote action space. If the tuple $\langle \text{span}_i, \text{boundary}_i, \text{type-label}_i \rangle$ is denoted as y_i, then the goal is to predict the sequence $y_1, y_2, \ldots, y_N$ and the action space Y_i is given as,

$$Y_i = A \times B_i \times Z_i \tag{1}$$

Components of action spaces are described below.

Span-Identifying Actions:

These actions take symbols from the set $A = \{\langle m \rangle, \rightarrow, \langle /m \rangle\}$, where:

TH-4046_216101004

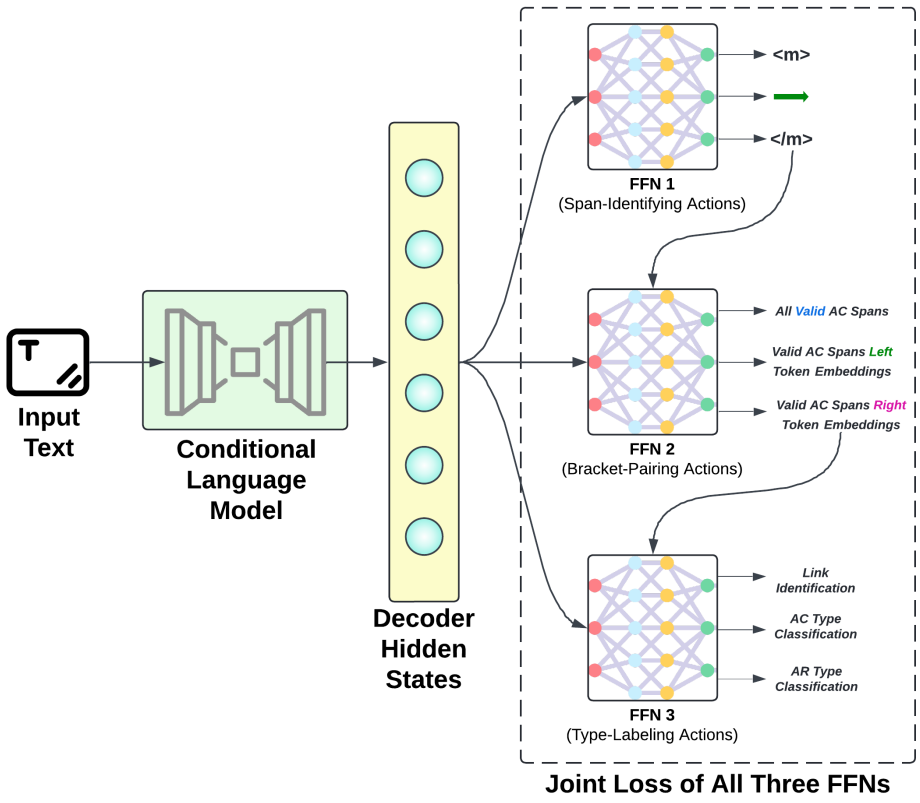

Fig. 6.2 Model Architecture of AASP

- $\langle m \rangle$ marks the beginning of an AC span,
- $\langle /m \rangle$ marks the end of the span, and
- $\rightarrow$ copies the input text as it appears from left to right, irrespective of whether it is an AC or Non-AC span.

These actions are performed using FFN 1 of Fig. 6.2.

Boundary-Pairing Actions: These actions denoted with set B_n given as,

$$B_n = \{i \mid i < j \wedge a_i = \langle m \rangle \wedge a_j = \langle /m \rangle \ \& \ i, j \in \{1, \ldots, n\}\} \tag{2}$$

Boundary-pairing actions connect the beginning of an AC span with the end of that AC span; that is, they connect the start action $\langle m \rangle$ with the end action $\langle /m \rangle$. Specifically, the sequence produced by the *Span-Identifying Actions*, is scanned from left to right, identifying each $\langle /m \rangle$ and immediately pairing it with the leftmost unmatched $\langle m \rangle$ before continuing. This way, the spans are resolved incrementally without crossing boundaries, and the unmatched symbols are removed as erroneous predictions. These actions are performed using both FFN 1 & FFN 2 jointly, as shown in Fig. 6.2.

With the completion of the *Span-Identifying* and *Boundary-Pairing* actions, the *ACI task* is completed.

Type-Labeling Actions: These actions, denoted as Z_n, perform the following: *(i) labelling AC spans into predefined AC types (ACC task)*, and *(ii) jointly establishing the relationships between related ACs and classifying their AR types (ARI & ARC tasks)*. Here, the sequence produced by the *Span-Identifying Actions* is traversed from left to right to find the $\langle /m \rangle$ symbol. Once found, it connects with the previously most related already identified $\langle /m \rangle$ to build the intra-span relationship. Additionally, both the spans corresponding to the $\langle /m \rangle$ symbols are classified into predefined AC types, and the connected link is classified into a predefined AR type. This whole process is performed using both FFN 2 & FFN 3 jointly in Fig. 6.2. Thus, if we define a set L that combines all AC and AR types, then:

$$Z_n = \{p \mid p < q \wedge a_p = \langle /m \rangle\} \times L \tag{3}$$

6.2.2 Conditional Modeling of Actions

Given an input argumentative text W, we transform it into a sequence of tuples denoted by $y = (y_1, y_2, \ldots, y_N)$. This sequence of tuples is modelled as the conditional probability given below:

$$p_\theta(y \mid W) = \prod_{n=1}^{N} p_\theta(y_n \mid y_{<n}, W) \tag{4}$$

Here, each $p_\theta(y_n \mid y_{<n}, W)$ corresponds to the probability of the n-th action given the preceding actions and the input argumentative text W.

Hence, the log-likelihood of the model is then given by:

$$\log p_\theta(y \mid W) = \sum_{n=1}^{N} \log p_\theta(y_n \mid y_{<n}, W) \tag{5}$$

We then calculate the probability value of $p_\theta(y_n \mid y_{<n}, W)$ over softmax of the dynamic set Y_n that changes as a function of the history $y_{<n}$, i.e.,

$$p_\theta(y_n \mid y_{<n}, W) = \frac{\exp s_\theta(y_n)}{\sum_{y'_n \in Y_n} \exp s_\theta(y'_n)} \tag{6}$$

where s_θ is a parameterized score function and is implemented using two independent feed-forward networks (FFN 2 & FFN 3) as:

$$s_\theta(y_n) = \begin{cases} \text{FFN}_{a_n}^{z_n}(p_n) & \text{if } a_n = \langle /m \rangle \\ \text{FFN}_{a_n}(h_n) & \text{otherwise} \end{cases} \tag{7}$$

Here, h_n is the decoder hidden state at step n, represented as a column vector, and $p_n = [h_n^\top; h_{b_n}^\top]^\top$ represents the AC span mention corresponding to y_n. Considering the longer text spans of ACs and the presence of distantly related ACs, we choose to select larger FFNs as compared to the ASP framework to better capture the relational dependencies.

The dynamic vocabulary is used at each time step, and the best-decoded sequence is finalized as y^* using the greedy decoding strategy. At decoding step n, y_n^* is calculated as:

$$y_n^* = \arg\max_{y_n'} p_\theta(y_n' \mid y_{<n}, W) \tag{8}$$

This resultant y_n^* is the best sequence of actions used to construct the argumentative structure for the given W.

6.3 Experimental Setup

6.3.1 Datasets

We evaluate our approach on these three AM benchmarks: **AAE**, **AAE-FG** and **CDCP**.

6.3.2 Implementation Details

We employ the *Flan-T5-Large* [103] as the conditional language model across all experiments. For the `AAE` and `AAE-FG` datasets, we use a batch size of 64, while for `CDCP`, the batch size is set to 16. The maximum output sequence length is capped at 256 for `AAE` and `AAE-FG` and at 1024 for `CDCP`. We utilize the AdamW optimizer with a learning rate of 5×10^{-5}. All experiments are conducted on a single NVIDIA A100 GPU. Each experiment is run for 200 epochs, with checkpoints saved every 200 steps. The best model is selected based on performance on the validation set. Results are averaged over three independent runs. Both the feed-forward neural networks include one hidden layer of size 1500 and are trained with a learning rate of 3×10^{-4}. We use the following libraries: (i) ASP framework[1] [112], and (ii) HuggingFace's Transformers[2].

[1]https://github.com/lyutyuh/ASP
[2]https://github.com/huggingface

6.3.3 Evaluation

Following the prior studies [73], we evaluate all four AM tasks using the *Micro-F1* score. We consider only exact matches of AC spans as correct, disregarding any partial matches to maintain strict alignment with previous works.

6.3.4 Baselines

We select the following SoTA models as baselines, comparing them to the datasets where applicable: **BiPAM** [100]: Dependency parsing for end-to-end AM using biaffine operations and BERT-base [101]. **BiPAM-syn** [100]: Extends BiPAM with syntactic information from the Stanford dependency parser. **BART-B** [113]: A generative model for aspect-based sentiment analysis, adapted for AM by [18]. **GMAM** [18]: Frames AM as sequence generation with positional encoding and pointer mechanisms. **ST** [73]: A dependency parser for AM using Longformer [74] and biaffine attention [114]. **ANL-AM** [76]: Adapts the TANL framework [75] for AM with text-to-text generation. **DENIM** [77]: A sequence generation-based AM framework with discourse-aware multi-task prompting.

6.4 Results and Discussion

6.4.1 Main Results and Comparison with Baselines

Table 6.1 compares the overall performance of AASP compared to recent baselines across all four AM tasks on the three standard datasets. Our method consistently outperforms existing baselines on the `AAE` and `AAE-FG` datasets, achieving average Micro-F1 score improvements of 3.3% and 3.81%, respectively. These results underscore the effectiveness of AASP in handling tree-structured argumentation. Notably, the improvements are most pronounced in the complex relational tasks (ARI and ARC), where AASP surpasses the baselines by 4.17% and 4.18% on `AAE`, and 5.74% and 5.12% on `AAE-FG`, compared to the gains in the component tasks (ACI and ACC): 2.05% and 2.79% on `AAE`, and 1.45% and 2.90% on `AAE-FG`. Specifically, in `AAE` dataset, the most recent generative SoTA baseline DENIM uses an additional discourse structure information through *prefix* to better capture the relational dependency of argumentative structures. In contrast, AASP produces SoTA results without using any additional customized information. On the graph-structured `CDCP` dataset, AASP demonstrates significant gains in the relational tasks, achieving a substantial improvement of 3.83% in ARI and ARC over the current SoTA. However, in component tasks

Dataset	Frameworks	Micro-F1 Scores of Key AM Tasks				AVG
		ACI	ACC	ARI	ARC	
AAE	BiPAM	–	72.90	–	45.90	–
	BiPAM-syn	–	73.50	–	46.40	–
	BART-B	81.71	73.61	49.75	47.93	63.25
	GMAM	84.10	75.94	50.40	50.08	65.13
	ST	85.02	75.43	55.75	55.19	67.84
	ANL-AM	–	76.93	–	58.57	–
	DENIM	85.75	76.50	59.55	58.51	70.08
	AASP (Ours)	**87.80**	**79.29**	**63.72**	**62.69**	**73.38 (+3.3)**
AAE-FG	GMAM*	84.20	57.32	30.81	25.95	49.57
	ST*	85.75	60.06	57.53	57.03	65.09
	AASP (Ours)	**87.20**	**62.96**	**63.27**	**62.15**	**68.90 (+3.81)**
CDCP	BART-B	–	56.15	–	13.76	–
	GMAM	–	57.72	–	16.57	–
	ANL-AM	–	**68.94**	–	28.42	–
	ST	**82.88**	68.90	31.94	31.94	**53.92**
	AASP (Ours)	73.82	62.14	**35.77**	**35.77**	51.88

Table 6.1 Performance comparison on the AAE, AAE-FG and CDCP datasets. The Micro-F1 scores of ACI, ACC, ARI and ARC tasks are reported with AVG, indicating the average score of all four tasks. Apart from AAE-FG, all the reported results are directly taken from the corresponding baselines. For AAE-FG, baselines with an asterisk (*) indicate results obtained using the corresponding open-sourced code with original hyperparameters used for AAE dataset. "–" indicates the results are not available for the corresponding baselines.

TH-4046_216101004

Dataset	Model	ACI	ACC	ARI	ARC	AVG
AAE	AASP	**87.80**	**79.29**	**63.72**	**62.69**	**73.38**
	AASP*(-rhs)*	87.61	78.23	61.49	60.19	71.88 (-1.50)
AAE-FG	AASP	**87.20**	62.96	**63.27**	**62.15**	**68.90**
	AASP*(-rhs)*	86.94	**63.43**	62.10	60.78	68.31 (-0.59)
CDCP	AASP	73.82	62.14	**35.77**	**35.77**	**51.88**
	AASP*(-rhs)*	73.82	**62.85**	33.94	33.94	51.13 (-0.74)

Table 6.2 Results of ablation experiments across datasets. AASP*(-rhs)* refers to the AASP framework with *Reduced Hidden States* of both the Feed-Forward Networks corresponding to the score function s_θ described in Section 6.2.2. Best scores are in **Bold**.

such as ACI and ACC, AASP shows limited performances in the CDCP dataset. While most baselines struggle with complex relational tasks, AASP demonstrates improved performance in both ARI and ARC tasks in both tree-structured and graph-structured argumentation. This highlights AASP's ability to model relational dependencies effectively across diverse datasets.

Given that the ST baseline has been widely applied to most standard datasets, it stands as a robust baseline for comparison. Therefore, in the subsequent sections, we compare all analyses with ST to better compare the performance of AASP.

6.4.2 Ablation Study

We conduct an ablation study to assess the impact of FFNs in the score function s_θ by altering their size. In AASP, the hidden state size is 1500, while the original ASP framework uses 150. To analyze this, we run experiments on all datasets with a hidden state size of 150 instead of 1500. We denote this variant as AASP*(-rhs)*, where *rhs* stands for *reduced hidden states*. The results, as shown in Table 6.2, indicate that AASP*(-rhs)* underperforms in terms of average scores across all four AM tasks compared to AASP. The ACI task performance is dropped in AAE and AAE-FG datasets with smaller FFNs. It means that the *Boundary-Pairing* actions require larger FFNs to finalise the boundaries efficiently. Although for AAE-FG and CDCP, the ACC task performance improves in AASP*(-rhs)*, the performance decline in ARI and ARC tasks is significant across all datasets. It suggests that the size of the FFNs plays a crucial role in effectively modelling complex relational dependencies. Specifically, the score function s_θ is better equipped to handle distantly related ACs when the *Type-Labeling* actions connect the $\langle/m\rangle$ symbols of both the ACs. As a by-product, the *chains of reasoning* is also handled efficiently when larger FFNs are used.

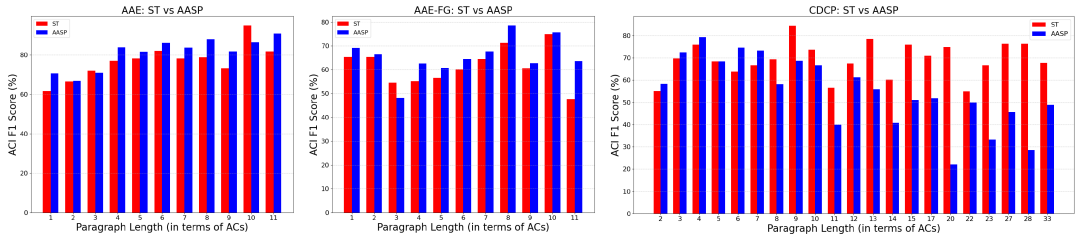

Fig. 6.3 Comparison of ACI Micro-F1 scores (%) between AASP and ST across paragraphs of varying lengths (in terms of the number of ACs present in that paragraph) for the AAE, AAE-FG, and CDCP datasets.

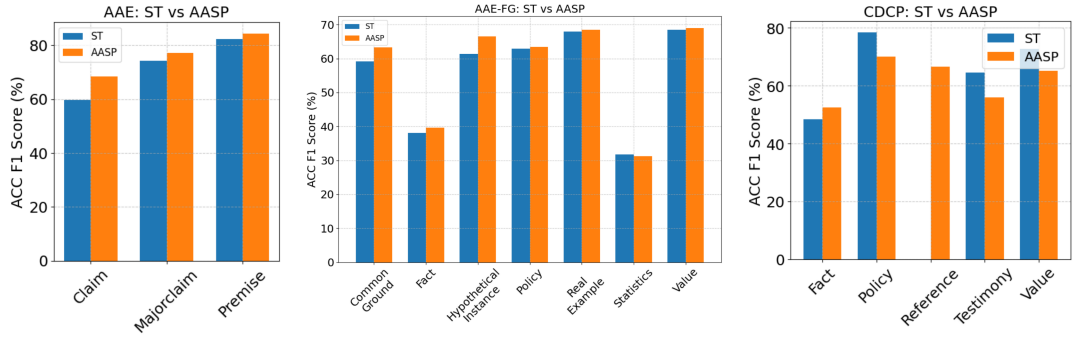

Fig. 6.4 Comparison of ACC Micro-F1 scores (%) for AASP and ST across different AC categories in the AAE, AAE-FG, and CDCP datasets.

6.4.3 Performance of the ACI task on Varied Length Paragraphs

In this section, we compare the performance of AASP with ST for the ACI task across datasets (See Fig. 6.3). In the AAE dataset, the Micro-F1 score for the ACI task improves as paragraph length (in ACs) increases, peaking at 9–11 ACs. The ST baseline starts at around 60% Micro-F1 for single-AC paragraphs and rises to 80% for longer ones. AASP begins slightly higher, at 65%, and consistently outperforms ST, exceeding 80% for the longest paragraphs. For the AAE-FG dataset, a similar trend is observed, but the overall performance is slightly lower than in AAE. Here, both ST and AASP show a more gradual improvement, beginning at around 65% and reaching approximately 70–75% for 8–10 ACs. In the CDCP dataset, the relationship between paragraph length and Micro-F1 score becomes interesting. The ST baseline performs well across a wide range of lengths, achieving high scores (70–80%) for most configurations. However, while AASP remains competitive, it underperforms compared to ST for longer paragraphs, especially beyond 8 ACs. But it surpasses ST by a noticeable margin for paragraphs with fewer than 8 ACs. This indicates that the AASP framework struggles with the ACI task for longer paragraphs in the complex, graph-structured CDCP dataset. However, it achieves superior performance across all three datasets for paragraphs of moderate length.

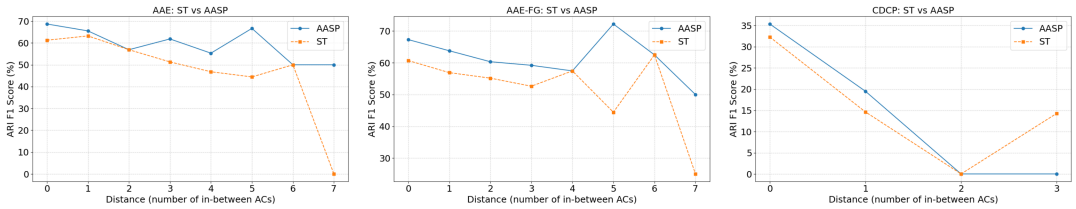

Fig. 6.5 Performance analysis of ARI task Micro-F1 scores (%) for AASP and ST across varying distances (in terms of number of intermediate ACs) in the AAE, AAE-FG, and CDCP datasets.

6.4.4 Analysis of the ACC task for Different AC Categories

In this section, we compare the performance of AASP and ST on different AC categories across datasets (see Fig. 6.4). In the `AAE` dataset, AASP outperforms ST across all AC types. The `PREMISE` category achieves the highest Micro-F1 score, making it the most predictable for both frameworks. Strong performance is also observed for the `MAJORCLAIM` type. Notably, AASP shows a significant improvement of nearly 10% over ST for the `CLAIM` category, making it the better choice for `CLAIM` classification. The `AAE-FG` dataset, with its more fine-grained AC labels, poses a greater challenge for both frameworks. In the `AAE-FG` dataset, the original AC categories from `AAE` are divided into finer-grained classes, reducing the training instances per class. Despite this, both frameworks perform robustly. AASP shows notable gains in categories like `HYPOTHETICAL INSTANCE` and `COMMON GROUND`, demonstrating its strength in handling finer-grained argumentation. The `CDCP` dataset presents mixed results. While ST performs better overall in the AC classification task, AASP outperforms ST in certain categories. For example, AASP achieves better performance in `FACT` classification. Surprisingly, ST fails to detect any instances of the `REFERENCE` category, whereas AASP achieves over 65% Micro-F1 score in this category. These empirical results emphasize the effectiveness of AASP for ACC task, particularly for fine-grained or underrepresented categories.

6.4.5 Analysis of ARI Task for Long-Range Relations

In argumentative paragraphs, the relationships between related ACs exhibit a fascinating variety of patterns. Some related ACs are directly linked with no intermediate ACs, forming straightforward connections, while others are moderately far apart, involving a few intermediate ACs. The most challenging cases arise when ACs are distantly related, with one appearing near the beginning of the paragraph and the other near the end. Identifying such varied-distant relations is critical for understanding complex argumentative structures. To assess this capability, we analyze the performance of both ST and AASP across varying distantly related ACs, as shown in Fig. 6.5. In

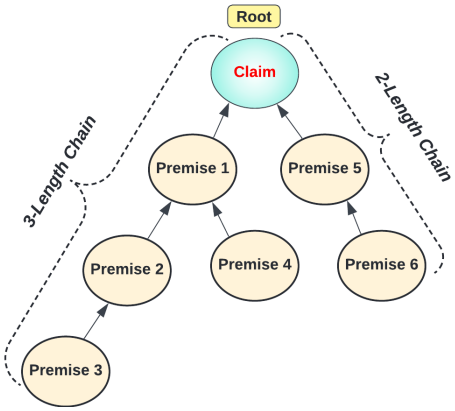

Fig. 6.6 Example of chains present in an argumentative structure

Dataset	Framework	Lengthwise Distributions of Chains			Accuracy (%)	
		Ground Truth	Predicted	Correct	2-length	3-length
AAE	ST	{2: 130, 3: 15, 4: 2}	{2: 109, 3: 5}	{2: 39}	30.00	0.00
	AASP	{2: 130, 3: 15, 4: 2}	{2: 124, 3: 16, 4: 1}	{2: 46, 3: 2}	**35.38**	**13.33**
AAE-FG	ST	{2: 130, 3: 15, 4: 2}	{2: 122, 3: 9, 4: 1}	{2: 37, 3: 2}	28.46	13.33
	AASP	{2: 130, 3: 15, 4: 2}	{2: 125, 3: 10}	{2: 45, 3: 3}	**34.62**	**20.00**
CDCP	ST	{2: 23, 3: 9, 4: 1}	{2: 18, 3: 10}	{2: 1}	4.35	0.00
	AASP	{2: 23, 3: 9, 4: 1}	{2: 13, 3: 1}	{2: 4}	**17.39**	0.00

Table 6.3 Performance Comparison of ST and AASP across datasets on Relation Chain Detection for Varied Length Chains.

the AAE dataset, AASP consistently outperforms ST across all distances. For direct relations (no intermediate ACs), AASP achieves a Micro-F1 score of nearly 70%, while ST lags behind at 60%. Although performance declines with increasing distance, AASP proves more robust. It maintains a Micro-F1 score above 50% even at the maximum distance of 7 intermediate ACs, where ST experiences a sharp drop. In the AAE-FG dataset, a similar trend is observed. The CDCP dataset presents a more challenging scenario due to the complex, graph-structured nature of its argumentative relations. Both frameworks show similar patterns in this dataset up to 2 intermediate ACs. However, for the maximum distance (3 intermediate ACs), AASP's performance drops sharply to 0%, highlighting the challenges of identifying long-range relations in this dataset.

6.4.6 Performance Analysis of Relational Chains

Relational chains (e.g. a series of premises/claims as shown in Fig. 6.6) are the building blocks of the *Chain of Reasoning* present in any argumentative discourse. Currently, none of the existing works has focused on this aspect of analysis. As shown in Fig. 6.6, an argumentative paragraph comprising seven ACs: $\{Claim, Premise_1, Premise_2, \ldots, Premise_6\}$. With these ACs, a 3-length sequential chain is formed as: $Premise_3 \rightarrow$

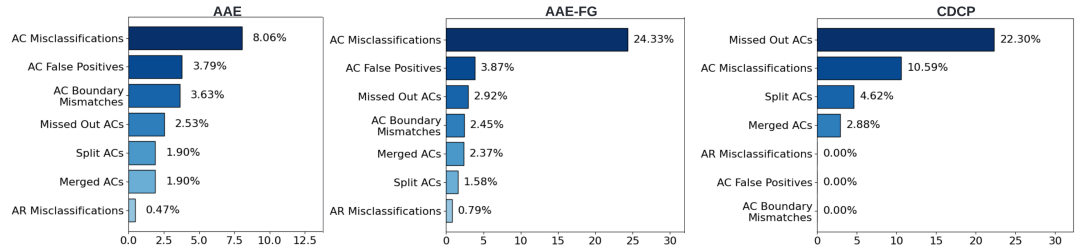

Fig. 6.7 Error distribution across three datasets with AASP framework, highlighting the percentage of various error categories.

$Premise_2 \rightarrow Premise_1 \rightarrow Claim$, with *Claim* as the root AC. Here, each AC is directly connected to the preceding one. In such configurations, identifying relations between adjacent ACs poses a significant challenge. This difficulty arises because all ACs in the chain are (in-)directly connected through the root AC, increasing the likelihood of misidentifying relations. For example, one might incorrectly predict $Premise_3 \rightarrow Premise_1$ instead of the correct $Premise_3 \rightarrow Premise_2$. In order to assess the performance in detecting these relational chains, we present an in-depth analysis in Table 6.3, where we compare ST and AASP frameworks across datasets. AASP achieves 35.38% accuracy for 2-length chains in `AAE` and 34.62% in `AAE-FG`, compared to ST's 30.00% and 28.46%, respectively. However, both frameworks struggle to accurately identify longer chains (3-length and above). The CDCP dataset, with its complex graph argumentative structures, poses challenges in both frameworks. However, AASP still surpasses ST with 17.39% accuracy for 2-length chains compared to ST's 4.35%. This comparative analysis highlights the effectiveness of AASP in detecting chains of ACs over existing baselines. Moreover, it also emphasizes the need for further advancements in modelling complex relational structures to capture the chain of reasoning more accurately.

6.4.7 Error Analysis

We conducted error analysis across all three datasets and identified the following errors (see Fig. 6.7): *(i) AC Boundary Mismatches*, where predicted ACs differ in length from the ground truth; *(ii) Missed ACs*, where valid ACs are not detected; *(iii) False Positive ACs*, where non-existent ACs are predicted; *(iv) AC Misclassifications*, where ACs are assigned incorrect types; *(v) AR Misclassifications*, where ARs are incorrectly labeled; *(vi) Split ACs*, where true ACs are divided into smaller segments; and *(vii) Merged ACs*, where multiple ACs are wrongly combined into one.

Analyzing the errors across the datasets, *AC Misclassifications* emerge as the most significant error type overall, particularly in AAE-FG (24.33%) and CDCP (10.59%), indicating a consistent challenge in correctly classifying ACs, where the number of AC

TH-4046_216101004

types is more. *Missed Out ACs*, another major source of error for the CDCP dataset with 22.30% error, suggests that the AASP struggles more with identifying all potential ACs in this complex graph-structured dataset than other datasets. *AC False Positives* and *AC Boundary Mismatches* show relatively similar proportions in AAE (3.79% and 3.63%, respectively) and AAE-FG (3.87% and 2.45%, respectively). Surprisingly, the CDCP dataset has no such errors due to the fact that in CDCP, each AC spans over a full sentence, unlike AAE or AAE-FG, where the ACs are further segmented inside sentences. Detecting full-sentence ACs is relatively easier than the segmented ACs. Upon manually analyzing the predictions of both versions of AAE, we find spans such as *"I believe that"* or *"People might argue that"* are included inside the AC spans, alleviating the incorrect boundaries. *Split ACs* and *Merged ACs*, which reflect structural prediction errors, remain minor across datasets but still noticeable, with AAE and AAE-FG showing similar patterns (1.9%-2.4% for both error types). However, the CDCP dataset shows a slightly higher occurrence of Split ACs (4.62%) compared to Merged ACs (2.88%) due to its ACs spanning over a full sentence. Whenever the sentence is longer, the split sometimes happens when connectives such as *"and"* or *"because"* are present inside sentences. Surprisingly, *AR Misclassifications* are the least frequent errors across all datasets, staying below 1%, indicating that the AASP performs relatively well in recognizing argumentative relations effectively.

6.5 Conclusion

In this chapter, we introduced **AASP**, an end-to-end framework designed to address all four key tasks of AM jointly. By extending the ASP framework, we effectively tackled the ACI task through span identification and boundary-pairing actions, while ACC, ARI, and ARC tasks were addressed using type-labeling actions. Our approach leverages larger FFNs to enhance the execution of these actions, improving the model's ability to process longer AC spans and distantly related ACs. This contributes to overall performance gains, as verified through our ablation study. To gain deeper insights into the relational performance, we conduct an in-depth relational chain analysis. The results highlight the effectiveness of AASP in outperforming competitive baselines. Out of the three benchmarks, AASP achieved SoTA results across all four AM tasks in two benchmarks while demonstrating SoTA performance for relational tasks in the remaining benchmark.

TH-4046_216101004

TH-4046_216101004

Chapter 7

Conclusions and Future Work

This thesis has embarked on a systematic exploration of advanced computational models for AM, charting a deliberate progression from generative, text-based formulations to direct, action-based structure prediction. The central narrative of this work has been the pursuit of more effective, efficient, and semantically aware methods for capturing the complex, often non-linear, nature of human argumentation. By identifying and addressing the limitations of each successive approach, this research has not only produced a series of SoTA models but also contributed to a deeper understanding of the methodological challenges inherent in the field. The journey began by enhancing text-to-text models with explicit linguistic cues, evolved to incorporate deeper semantic knowledge, expanded into the biomedical domain, and culminated in a paradigm shift away from text-flattening towards a more direct and intuitive method of structure construction.

7.1 Scope of the Thesis

The contributions of this thesis are presented across five distinct but interconnected stages:

Chapter 3: The initial investigation began with the `argTANL` framework, which established the viability of a text-to-text generative approach for end-to-end AM. Later, this chapter demonstrated that surface-level linguistic signals, specifically argumentative and discourse markers, could be explicitly integrated into the generative process through fine-tuning and direct output enhancement (`ME-argTANL`). This work affirmed that even without deep semantic understanding, guiding models with explicit structural cues significantly improves their ability to identify argumentative structures.

TH-4046_216101004

Chapter 4: Building upon the generative paradigm, the second chapter addressed a key limitation of the marker-based approach, namely its reliance on surface-level indicators. This work shifted focus inward to the semantic content of the arguments themselves, introducing the concept of "Related Key Phrases" as the conceptual bridges between related components. By using an LLM to extract these semantic links and embedding them within the generative output, this framework demonstrated that enriching the model's input with fine-grained semantic knowledge leads to a more robust and accurate identification of argumentative relations, establishing a new SoTA.

Chapter 5: This chapter extends the exploration of AM into the biomedical domain, introducing `Ent-BAM`, a generative framework for Biomedical Argument Mining enriched with entity-level information. Leveraging domain-specific cues such as entity co-reference and co-occurrence, this work integrates biomedical semantics within the generative AM paradigm. The proposed approach significantly enhances the interpretability and precision of biomedical argumentative relation modeling, establishing a strong foundation for domain-adapted generative AM.

Chapter 6: The final chapter addresses a fundamental inefficiency shared by previous generative models, which is the necessity of flattening hierarchical argument structures into linear text sequences. To overcome this, the thesis introduces the Autoregressive Argumentative Structure Prediction (`AASP`) framework, which reframes AM not as text generation but as direct structure construction through a sequence of discrete, learnable actions. By modeling the task in this way, `AASP` provides a more efficient and direct method for capturing complex dependencies, excelling at modeling long-range relations and chains of reasoning without the redundancy of text-flattening.

Collectively, these chapters represent a comprehensive inquiry into how different forms of knowledge, from surface-level markers to deep semantic links, domain-specific biomedical entities, and explicit structural actions, can be leveraged to parse argumentative discourse with increasing semantic depth and structural fidelity.

7.2 Potential Future Directions

The findings and methodologies developed in this thesis open up several promising avenues for future research. Each of the frameworks presented provides unique insights and techniques that, when combined or extended, can push the boundaries of computational AM even further.

Hybrid and Integrated Models: Future work could explore hybrid models that combine the strengths of each chapter. For instance, the action-based `AASP` framework from Chapter 6 could be enhanced with the semantic knowledge of key phrases

TH-4046_216101004

introduced in Chapter 4, potentially by using key phrases to inform or constrain the type-labeling actions. Similarly, the argumentative and discourse markers proposed in Chapter 3 could serve as auxiliary features to guide the span-identifying actions in AASP, creating a unified model that leverages surface, semantic, and structural cues in tandem.

Advanced Integration of Large Language Models: While Chapter 4 utilized an LLM as a pre-processing component for key phrase extraction, deeper integration remains largely unexplored. Future research could investigate the use of LLMs in few-shot or zero-shot settings for full end-to-end AM, or employ them to generate natural language explanations for the structures predicted by the AASP framework. Such integration could enhance interpretability and adaptability, enabling the system to reason about argumentative structures in a more human-like manner.

Biomedical Argument Mining Extensions: The domain-specific framework introduced in Chapter 5, Ent-BAM, opens a new direction for domain-adapted AM. Future work could extend this framework by incorporating biomedical knowledge graphs or contextualized entity representations to enhance relational inference. Another promising avenue involves exploring cross-domain transfer, where models trained on biomedical argumentative structures can inform or regularize AM in other technical or evidence-based domains such as legal or policy texts.

Cross-Domain and Multilingual Argument Mining: The experiments presented in this thesis primarily focused on English-language datasets. A critical next step involves evaluating the generalizability of these frameworks, particularly the action-based AASP model from Chapter 6, across different languages and discourse conventions. Applying these approaches to domains such as legal reasoning, scientific literature, or social discourse can reveal how cultural and linguistic diversity influences argumentative structure and reasoning patterns.

Application-Oriented Evaluation: Finally, future work should aim to evaluate these models in real-world applications. Integrating these frameworks into systems for automated essay assessment, debate summarization, or policy argument analysis would demonstrate their practical potential while uncovering new challenges related to domain noise, interpretability, and user interaction. Addressing these challenges will pave the way for the next generation of AM systems that are not only accurate but also context-aware and socially relevant.

TH-4046_216101004

TH-4046_216101004

List of Publications

Manuscripts Published

1. **Nilmadhab Das**, Vishal Choudhary, V. Vijaya Saradhi, and Ashish Anand, "Exploration of Marker-Based Approaches in Argument Mining through Augmented Natural Language," in *Proceedings of the International Joint Conference on Neural Networks (**IJCNN**)*, Italy, 2025.

2. **Nilmadhab Das**, V. Vijaya Saradhi, and Ashish Anand, "On the Role of Key Phrases in Argument Mining," in *Findings of the 2025 Conference of the Nations of Americas Chapter of the Association for Computational Linguistics (**NAACL**)*, New Mexico, 2025.

3. **Nilmadhab Das**, Yash S. Choudhary, V. Vijaya Saradhi, and Ashish Anand, "Ent-BAM: A Generative Framework for Biomedical Argument Mining Enriched with Entity Information," In *IEEE Transactions on Artificial Intelligence (**TAI**)*, 2025.

4. **Nilmadhab Das**, Vishal Vaibhav, Yash Sunil Choudhary, V. Vijaya Saradhi, and Ashish Anand, "End-to-End Argument Mining through Autoregressive Argumentative Structure Prediction," in *Proceedings of the International Joint Conference on Neural Networks (**IJCNN**)*, Italy, 2025.

Manuscripts Under Review

1. **Nilmadhab Das**, V. Vijaya Saradhi, and Ashish Anand, "Span-Constrained Oracle Replay Mechanism for Argument Mining," *Submitted to **ACL** 2026 through ARR January 2026 cycle*.

2. **Nilmadhab Das**, Sayan Pal, V. Vijaya Saradhi, and Ashish Anand, "CIARAM: Class Imbalance Aware Generative Framework for Relational Argument Mining," *Submitted to 15th edition of the Language Resources and Evaluation Conference (**LREC**)*, Spain, 2026.

3. **Nilmadhab Das**, Sanjana Siri Kolisetty, Aasneh Prasad, Ayush Kumar, Yogesh Mishra, Akshay Shome, Mithilesh Kumar Jha and Ashish Anand. "The Argumentative Landscape of Administrative Reports: A Corpus Study." *Submitted to 15th edition of the Language Resources and Evaluation Conference (**LREC**)*, Spain, 2026.

TH-4046_216101004

TH-4046_216101004

References

[1] Y. Ajjour, W.-F. Chen, J. Kiesel, H. Wachsmuth, and B. Stein, "Unit Segmentation of Argumentative Texts," in *Proceedings of the 4th Workshop on Argument Mining*, 2017.

[2] H. Wang, Z. Huang, Y. Dou, and Y. Hong, "Argumentation Mining on Essays at Multi Scales," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[3] J.-F. Lin, K. Y. Huang, H.-H. Huang, and H.-H. Chen, "Lexicon Guided Attentive Neural Network Model for Argument Mining," in *Proceedings of the 6th Workshop on Argument Mining*, 2019.

[4] S. Dutta, J. Juneja, D. Das, and T. Chakraborty, "Can Unsupervised Knowledge Transfer from Social Discussions Help Argument Mining?" in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[5] E. Cabrio and S. Villata, "Five Years of Argument Mining: a Data-driven Analysis," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[6] D. Gemechu and C. Reed, "Decompositional Argument Mining: A General Purpose Approach for Argument Graph Construction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[7] J. Lawrence and C. Reed, "Argument mining: A survey," *Computational Linguistics*, 2019.

[8] T. Mayer, E. Cabrio, and S. Villata, "Transformer-based argument mining for healthcare applications," in *ECAI*, 2020.

[9] Z. Ke, W. Carlile, N. Gurrapadi, and V. Ng, "Learning to give feedback: Modeling attributes affecting argument persuasiveness in student essays," in *IJCAI*, 2018.

[10] V. R. Walker, D. Foerster, J. M. Ponce, and M. Rosen, "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*, 2018.

[11] V. Niculae, J. Park, and C. Cardie, "Argument Mining with Structured SVMs and RNNs," in *ACL*, 2017.

[12] C. Stab and I. Gurevych, "Parsing Argumentation Structures in Persuasive Essays," *Computational Linguistics*, 2017.

TH-4046_216101004

[13] T. Chakrabarty, C. Hidey, S. Muresan, K. McKeown, and A. Hwang, "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)*, 2019.

[14] T. Kuribayashi, H. Ouchi, N. Inoue, P. Reisert, T. Miyoshi, J. Suzuki, and K. Inui, "An Empirical Study of Span Representations in Argumentation Structure Parsing," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[15] N. Kotonya and F. Toni, "Explainable Automated Fact-Checking: A Survey," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[16] J. Bao, B. Liang, J. Sun, Y. Zhang, M. Yang, and R. Xu, "Argument Pair Extraction with Mutual Guidance and Inter-sentence Relation Graph," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[17] G. Morio, H. Ozaki, T. Morishita, and K. Yanai, "End-to-end Argument Mining with Cross-corpora Multi-task Learning," *Transactions of the Association for Computational Linguistics*, May 2022.

[18] J. Bao, Y. He, Y. Sun, B. Liang, J. Du, B. Qin, M. Yang, and R. Xu, "A Generative Model for End-to-End Argument Mining with Reconstructed Positional Encoding and Constrained Pointer Mechanism," in *EMNLP*, 2022.

[19] R. Schaefer, R. Knaebel, and M. Stede, "Towards fine-grained argumentation strategy analysis in persuasive essays," in *Proceedings of the 10th Workshop on Argument Mining*, 2023.

[20] T. Mayer, E. Cabrio, and S. Villata, "Transformer-Based Argument Mining for Healthcare Applications," in *ECAI 2020*, 2020.

[21] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, 1995.

[22] C. Cayrol and M.-C. Lagasquie-Schiex, "On the acceptability of arguments in bipolar argumentation frameworks," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 2005.

[23] C. Stab and I. Gurevych, "Identifying Argumentative Discourse Structures in Persuasive Essays," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[24] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th international conference on Artificial intelligence and law*, 2007.

[25] L. Lugini and D. Litman, "Contextual Argument Component Classification for Class Discussions," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[26] T. Goudas, C. Louizos, G. Petasis, and V. Karkaletsis, "Argument Extraction from News, Blogs, and the Social Web," *International Journal on Artificial Intelligence Tools*, 2015.

[27] J. A. Rodrigues and A. Branco, "Transferring confluent knowledge to argument mining," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022.

[28] M. Lippi and P. Torroni, "Argumentation mining: State of the art and emerging trends," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, pp. 1–25, 2016.

[29] M. Stede, J. Schneider, and G. Hirst, *Argumentation mining*. Springer, 2019.

[30] C. Reed, "Preliminary results from an argument corpus," *Linguistics in the twenty-first century*, 2006.

[31] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis, "Argument Extraction from News," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015.

[32] N. Madnani, M. Heilman, J. Tetreault, and M. Chodorow, "Identifying High-Level Organizational Elements in Argumentative Discourse," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012.

[33] K. Al-Khatib, H. Wachsmuth, J. Kiesel, M. Hagen, and B. Stein, "A News Editorial Corpus for Mining Argumentation Strategies," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

[34] I. Habernal and I. Gurevych, "Argumentation Mining in User-Generated Web Discourse," *Computational Linguistics*, 2017.

[35] M. Spliethöver, J. Klaff, and H. Heuer, "Is It Worth the Attention? A Comparative Evaluation of Attention Layers for Argument Unit Segmentation," in *Proceedings of the 6th Workshop on Argument Mining*, 2019.

[36] G. Petasis, "Segmentation of Argumentative Texts with Contextualised Word Representations," in *Proceedings of the 6th Workshop on Argument Mining*, 2019.

[37] D. Trautmann, J. Daxenberger, C. Stab, H. Schütze, and I. Gurevych, "Fine-Grained Argument Unit Recognition and Classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems*, 2013.

[39] J. Lawrence, C. Reed, C. Allen, S. McAlister, and A. Ravenscroft, "Mining Arguments From 19th Century Philosophical Texts Using Topic Based Modelling," in *Proceedings of the First Workshop on Argumentation Mining*, 2014.

[40] P. Saint-Dizier, "Knowledge-driven argument mining based on the qualia structure," *Argument & Computation*, 2017.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017.

TH-4046_216101004

[42] C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych, "Cross-topic Argument Mining from Heterogeneous Sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[43] R. Egawa, G. Morio, and K. Fujita, "Annotating and Analyzing Semantic Role of Elementary Units and Relations in Online Persuasive Arguments," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019.

[44] R. M. Palau and M.-F. Moens, "Argumentation mining: the detection, classification and structure of arguments in text," in *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, 2009.

[45] V. W. Feng and G. Hirst, "Classifying arguments by scheme," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[46] J. Park and C. Cardie, "Identifying Appropriate Support for Propositions in Online User Comments," in *Proceedings of the First Workshop on Argumentation Mining*, 2014.

[47] J. Lawrence and C. Reed, "Combining Argument Mining Techniques," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015.

[48] C. Schulz, S. Eger, J. Daxenberger, T. Kahse, and I. Gurevych, "Multi-Task Learning for Argumentation Mining in Low-Resource Settings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

[49] J. Park and C. Cardie, "A Corpus of eRulemaking User Comments for Measuring Evaluability of Arguments," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[50] C. Stab and I. Gurevych, "Annotating Argument Components and Relations in Persuasive Essays," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.

[51] S. Dutta, J. Juneja, D. Das, and T. Chakraborty, "Can unsupervised knowledge transfer from social discussions help argument mining?" in *ACL*, 2022.

[52] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[53] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer Networks," in *Advances in Neural Information Processing Systems*, 2015.

[54] L. Carstens and F. Toni, "Towards relation based Argumentation Mining," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015.

[55] A. Peldszus and M. Stede, "Towards Detecting Counter-considerations in Text," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015.

[56] H. Nguyen and D. Litman, "Context-aware Argumentative Relation Mining," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[57] O. Cocarascu and F. Toni, "Identifying attack and support argumentative relations using deep learning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[58] A. Peldszus and M. Stede, "Ranking the annotators: An agreement study on argumentation structure," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[59] J. Visser, B. Konat, R. Duthie, M. Koszowy, K. Budzynska, and C. Reed, "Argumentation in the 2016 US presidential elections: annotated corpora of television debates and social media reaction," *Language Resources and Evaluation*, 2020.

[60] J. P. Chang, C. Chiam, L. Fu, A. Wang, J. Zhang, and C. Danescu-Niculescu-Mizil, "ConvoKit: A Toolkit for the Analysis of Conversations," in *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2020.

[61] R. Ruiz-Dolz, J. Alemany, S. M. H. Barbera, and A. Garcia-Fornes, "Transformer-based models for automatic identification of argument relations: A cross-domain evaluation," *IEEE Intelligent Systems*, 2021.

[62] N. Tran and D. Litman, "Multi-task learning in argument mining for persuasive online discussions," in *ArgMining Workshop*, 2021.

[63] B. Liu, V. Schlegel, R. Batista-Navarro, and S. Ananiadou, "Argument mining as a multi-hop generative machine reading comprehension task," in *Findings of EMNLP*, 2023.

[64] D. Paul, J. Opitz, M. Becker, J. Kobbe, G. Hirst, and A. Frank, "Argumentative relation classification with background knowledge," in *Comma*, 2020.

[65] A. Saadat-Yazdi, J. Z. Pan, and N. Kokciyan, "Uncovering implicit inferences for improved relational argument mining," in *Proceedings of EACL*, 2023.

[66] Y. Sun, B. Liang, J. Bao, M. Yang, and R. Xu, "Probing structural knowledge from pre-trained language model for argumentation relation classification," in *Findings of EMNLP*, 2022.

[67] M. L. Contalbo, F. Guerra, and M. Paganelli, "Argument relation classification through discourse markers and adversarial training," in *Proceedings of EMNLP*, 2024.

[68] T. Kuribayashi, H. Ouchi, N. Inoue, P. Reisert, T. Miyoshi, J. Suzuki, and K. Inui, "An empirical study of span representations in argumentation structure parsing," in *ACL*, 2019.

[69] P. Potash, A. Romanov, and A. Rumshisky, "Here's my point: Joint pointer architecture for argument mining," in *EMNLP*, 2017.

[70] A. Galassi, M. Lippi, and P. Torroni, "Argumentative link prediction using residual networks and multi-objective learning," in *Proceedings of the 5th Workshop on Argument Mining*, 2018.

[71] G. Morio, H. Ozaki, T. Morishita, Y. Koreeda, and K. Yanai, "Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization," in *ACL*, Jul. 2020.

[72] J. Bao, C. Fan, J. Wu, Y. Dang, J. Du, and R. Xu, "A neural transition-based model for argumentation mining," in *ACL-IJCNLP*, 2021.

[73] G. Morio, H. Ozaki, T. Morishita, and K. Yanai, "End-to-end argument mining with cross-corpora multi-task learning," *TACL*, 2022.

[74] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *ArXiv*, 2020.

[75] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, and S. Soatto, "Structured prediction as translation between augmented natural languages," *ArXiv*, 2021.

[76] M. Kawarada, T. Hirao, W. Uchida, and M. Nagata, "Argument mining as a text-to-text generation task," in *Proceedings of EACL*, 2024.

[77] Y. Sun, G. Chen, C. Yang, J. Bao, B. Liang, X. Zeng, M. Yang, and R. Xu, "Discourse structure-aware prefix for generation-based end-to-end argumentation mining," in *ACL*, 2024.

[78] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020.

[79] T. Mayer, E. Cabrio, M. Lippi, P. Torroni, and S. Villata, "Argument mining on clinical trials," in *Computational Models of Argument*, 2018.

[80] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[81] A. Galassi, M. Lippi, and P. Torroni, "Multi-task attentive residual networks for argument mining," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[82] J. Si, L. Sun, D. Zhou, J. Ren, and L. Li, "Biomedical argument mining based on sequential multi-task learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

[83] B. Liu, V. Schlegel, R. Batista-navarro, and S. Ananiadou, "Entity coreference and co-occurrence aware argument mining from biomedical literature," in *CODI*, 2023.

[84] J. Lawrence, J. Park, K. Budzynska, C. Cardie, B. Konat, and C. Reed, "Using argumentative structure to interpret debates in online deliberative democracy and erulemaking," *ACM Transactions on Internet Technology*, 2017.

[85] H. V. Nguyen and D. J. Litman, "Argument mining for improving the automated scoring of persuasive essays," in *AAAI*, 2018.

[86] Z. Chen, D. Verdi do Amarante, J. Donaldson, Y. Jo, and J. Park, "Argument mining for review helpfulness prediction," in *EMNLP*, 2022.

[87] J. Daxenberger, S. Eger, I. Habernal, C. Stab, and I. Gurevych, "What is the essence of a claim? cross-domain claim identification," in *EMNLP*, 2017.

[88] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various NER subtasks," in *ACL-IJCNLP*, 2021.

[89] U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois, "An autoregressive text-to-graph framework for joint entity and relation extraction," in *AAAI*, 2024.

[90] J. Quan, D. Xiong, B. Webber, and C. Hu, "GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue," in *EMNLP-IJCNLP*, 2019.

[91] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, and S. Soatto, "Structured prediction as translation between augmented natural languages," in *ICLR*, 2021.

[92] Y. Gao, N. Gu, J. Lam, and R. H. Hahnloser, "Do Discourse Indicators Reflect the Main Arguments in Scientific Papers?" in *Proceedings of the 9th Workshop on Argument Mining*, 2022.

[93] D. Sileo, T. Van de Cruys, C. Pradel, and P. Muller, "DiscSense: Automated semantic analysis of discourse markers," in *LREC*, 2020.

[94] A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann, "Building a web-scale dependency-parsed corpus from CommonCrawl," in *LREC*, 2018.

[95] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, 2020.

[96] I. Persing and V. Ng, "End-to-End Argumentation Mining in Student Essays," in *NAACL-HLT*, 2016.

[97] S. Eger, J. Daxenberger, and I. Gurevych, "Neural End-to-End Learning for Computational Argumentation Mining," in *ACL*, 2017.

[98] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *ACL*, 2016.

[99] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," in *ACL-IJCNLP*, 2015.

[100] Y. Ye and S. Teufel, "End-to-end argument mining as biaffine dependency parsing," in *EACL*, 2021.

[101] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019.

[102] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An empirical study of catastrophic forgetting in large language models during continual fine-tuning," *ArXiv*, 2023.

[103] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *JMLR*, 2024.

[104] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[105] Y. Sun, M. Wang, J. Bao, B. Liang, X. Zhao, C. Yang, M. Yang, and R. Xu, "PITA: Prompting task interaction for argumentation mining," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 5036–5049. [Online]. Available: https://aclanthology.org/2024.acl-long.275/

[106] T. Mayer, S. Marro, E. Cabrio, and S. Villata, "Enhancing evidence-based medicine with natural language argumentative analysis of clinical trials," *Artif. Intell. Med.*, 2021.

[107] D. Zheng, M. Lapata, and J. Z. Pan, "Large language models as reliable knowledge bases?" *arXiv preprint arXiv:2407.13578*, 2024.

[108] S. Wadhwa, S. Amir, and B. Wallace, "Revisiting relation extraction in the era of large language models," in *ACL*, 2023.

[109] G. Jiang, Z. Luo, Y. Shi, D. Wang, J. Liang, and D. Yang, "Toner: Type-oriented named entity recognition with generative language model," in *LREC*, 2024.

[110] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: efficient finetuning of quantized llms (2023)," *arXiv preprint arXiv:2305.14314*, 2023.

[111] N. Stylianou and I. Vlahavas, "Transformed: End-to-end transformers for evidence-based medicine and argument mining in medical literature," *Journal of Biomedical Informatics*, 2021.

[112] T. Liu, Y. Jiang, N. Monath, R. Cotterell, and M. Sachan, "Autoregressive structured prediction with language models," in *EMNLP*, 2022.

[113] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," in *ACL-IJCNLP*, 2021.

[114] T. Dozat and C. D. Manning, "Simpler but more accurate semantic dependency parsing," in *ACL*, 2018.