MSR-RAM: Macro-Structure Representation for Robust Argument Mining

Anonymous ACL submission

Abstract

Argument Mining (AM) involves detecting Argument Relations (ARs) between Argumentative Discourse Units (ADUs) to uncover the argument structure. AR detection techniques often rely on micro-structural features derived from the internal structure of ADUs. However, argument structure is guided by a macro-structure representing the functional interdependence among ADUs of the argument. This macro-structure comprises segments, each segment containing ADUs serving specific functions to maintain coherence within that segment (**local coherence**) and cross-segment coherence (**global coherence**). This paper proposes an approach capturing such macro-structure encoding both local and global coherence for detecting AR. Experimental results on heterogeneous datasets showcase a notable performance enhancement, outperforming state-of-the-art models for both in-dataset and cross-dataset evaluation scenarios. The cross-dataset evaluation result underscores that the macro-structure boosts AR prediction skill transferable to new dataset.

1 Introduction

Argument Mining (AM), a Natural Language Processing (NLP) task, involves identifying and analysing argument structures within text (Persing and Ng, 2016; Stab and Gurevych, 2017; Eger et al., 2017; Potash et al., 2016; Lawrence and Reed, 2020). It comprises several tasks including segmenting arguments into Argumentative Discourse Units (ADUs) (Peldszus and Stede, 2015a), distinguishing argumentative units from non-argumentative ones, classifying ADUs, labelling argument relation (AR) between ADUs, and identifying argument schemes (Eger et al., 2017; Lawrence and Reed, 2020). In this study, we focus on classifying the AR between ADUs into supporting, attacking, and none-relation.

In the literature, AR detection is framed as dependency parsing task (Peldszus and Stede, 2015b), sequence tagging problem (Eger et al., 2017) and sequence classification task, (Reimers et al., 2019; Ruiz-Dolz et al., 2021). Across these configurations, there is a notable emphasis on the features derived from the internal structure of ADUs. This feature resonates with the concept of "logical form" central to deductive logic also referred to as the micro-structure of an argument (Freeman, 2011). Some of these works apply classifiers on the features derived from individual ADUs (Moens et al., 2007), while others exploit the dependencies between pairs of ADUs, using dependency parsing (Muller et al., 2012b), similarity techniques (Lawrence et al., 2014), linguistic indicators (Villalba and Saint-Dizier, 2012), and embeddings from Large Language Models (LLMs) (Reimers et al., 2019; Ruiz-Dolz et al., 2021).

AR, however, is governed by macro-structure that represents an argument as an arrangement of ADUs where these ADUs interact to fulfil functional roles within a discourse (Grosz and Sidner, 1986; Grosz et al., 1995; Accuosto and Saggion, 2020; Boltužić and Šnajder, 2016; Freeman, 2011). Such functional support between ADUs allows the formation of coherent argument structure involving chain of thoughts. At global-level, the macro-structure addresses the overall argument's communicative intent, known as discourse purpose (DP) that is further divided into segments (local-structures), addressing its discourse segment purpose (DSP). The local-structure maintains coherence within segments (local coherence) by tracing the flow of ideas relevant to the DSP, while the global-structure ensures cross-segment coherence (global coherence) (Grosz et al., 1995; Lochbaum, 1994). Accordingly, AR is tied to its role within the structure and contingent upon maintaining such coherence.

The macro-structure of an argument, reflecting

such functional interdependence among ADUs, can be captured by encoding **argument flow** into the representation of argument.

Argument flow (Travis, 1984; Wang et al., 2019; Kazemnejad et al., 2024) can be captured by leveraging **ADU order**, analogous to how positional encoding in Transformers (Vaswani et al., 2017) captures the sequence of tokens. In this context, we use the sequential order of ADUs rather than tokens. This feature aligns with the argumentation framework proposed by Grosz and Sidner (1986), suggesting that ADUs naturally form argument segments, similar to how words form phrases within a sentence. ADUs within a segment serve specific roles, just as words in a phrase do, while discourse segments fulfil functions within the overall discourse, like phrases in a sentence.

Additionally, argument flow can be captured by tracking the transitions between participants of the argument. This argument dynamic can be modelled using theoretical frameworks like Inference Anchoring Theory (IAT) (Budzynska and Reed, 2011), which explores the interplay between dialogue and argument structures, as also explored in philosophy and linguistics through approaches like Dialogue Macrogame Theory (Mackenzie, 1990; Mann, 2002). This interaction between participants, also known as **proponent-opponent interaction** (Freeman, 2011), unfolds as a sequence of dialogue moves such as "Asserting", "Arguing", "Questioning", where each dialogue move is mapped to a structural element within the argument structure (Budzynska and Reed, 2011) (see Figure 3b for an illustrative example).

Moreover, arguments can be deconstructed into **local-structures**, which are crucial for tracing the sequence of ADUs that contribute to the reasoning both leading up to and following an AR. These local-structures facilitate the identification of local coherence and DSPs relevant to the AR, thereby providing a specific relevant context. The example in Figure 3a illustrates the interplay between local-structures, where one local-structure addresses a DSP pertaining to the Scottish National Party's internal divisions and disagreements, while another focuses on conducting disagreements respectfully (see Figure 2 for more examples).

In this study, we model AR prediction as a function of both micro-structure and macro-structure. An argument is modelled by unified representation that integrates both micro-structure and macro-structure. The micro-structure is captured through the representation of ADUs, while the macro-structure is addressed by modelling argument flow, by leveraging ADU order and proponent-opponent transitions. An attention mechanism is employed to attend to these unified representation for capturing the relationships among ADUs within the argument, similar to the attention mechanism in transformers attending to positional and token encoding for representing text (Vaswani et al., 2017). The resulting attention output is used for identifying local-structures relevant to AR while simultaneously detecting ARs through a multi-task learning approach. The local-structures prediction functions as an auxiliary task to supply contextual cues for the AR prediction.

Our contributions are threefold: First, we introduce an approach that explicitly incorporates macro-structures to capture both local and global coherence for AR detection. Second, we demonstrate that leveraging macro-structures enhances AR detection performance. Third, our evaluation shows that integrating macro-structures provides robust AR detection features that are transferable to new datasets. It achieves a new state-of-the-art performance in both cross-dataset and in-dataset evaluation setups.

2 Related Works

Argument Mining. has been explored through various approaches in the literature. One approach frames AM as a dependency parsing task (Peldszus and Stede, 2015b), employing discourse parsing techniques (Muller et al., 2012a). Similarly, Peldszus and Stede (2016) aim to map Rhetorical Structure Theory (RST) trees to argumentation structures (Taboada and Mann, 2006) using sub-graph matching and evidence graph models. Additionally, AM has been framed as a token-based sequence tagging problem (Eger et al., 2017), where tokens are classified into argument components (premise, conclusion) and their respective argument relations (support, attack) using the BIO tagging approach. Gemechu and Reed (2019) decompose propositions into fine-grained components and use classifiers to predict AR based on the relations between these components. Several recent works fine-tuned pre-trained LLM on AM data-set based on sequence pair classification configurations (Reimers et al., 2019; Ruiz-Dolz et al., 2021). These configurations primarily focus on the internal structure of ADUs, akin to the "logical form" central to de-

ductive logic. Freeman (Freeman, 2011) refers to this as the "micro-structure of an argument". However, these setups overlook the overall functioning of whole ADUs in an argument and focus on the micro-structure obtained from pairs of ADUs.

End-to-end AM aims to leverage the interdependence between tasks to enhance performance (Persing and Ng, 2016; Stab and Gurevych, 2017; Eger et al., 2017; Potash et al., 2016; Morio et al., 2022). Persing and Ng (2016) and Stab and Gurevych (2017) employ a pipeline architecture, training independent models for each sub-task and subsequently defining an Integer Linear Programming (ILP) model to encode global constraints. Eger et al. (2017) propose a neural end-to-end paradigm, addressing the problem in a joint multi-task setup and exploiting the inter-dependency between tasks. Additionally, Morio et al. (2022) introduce an end-to-end cross-corpus training approach, while Bao et al. (2022) presents a generative framework for end-to-end AM using a constrained pointer mechanism (CPM) and reconstructed positional encoding (RPE). None of these works explicitly encode macro-structure other than exploiting the inter-dependency between the AM tasks.

While advances in transfer learning, leveraging LLMs, showcase encouraging performance on in-dataset evaluation, they encounter challenges when applied to different datasets from distinct domains. Studies in the NLI domain reveal their struggle to learn robust features applicable across datasets (McCoy et al., 2019). These shortcomings stem from shortcut learning (Wu et al., 2023; McCoy et al., 2023), leading models to adopt shallow heuristics hindering the capacity to generalise (Naik et al., 2018; Poliak et al., 2018; Nie et al., 2019; McCoy et al., 2019). Notably, McCoy et al. (2019) shows that NLI models learn invalid heuristics that are regularly expressed in the dataset, while Poliak et al. (2018) show that they tend to predict entailment solely based on hypotheses, exploiting dataset-specific artifacts by neglecting discourse-level context—a crucial concept in AR prediction.

Structural Encoding in LLMs. Recent advancements extend Transformers in encoding long text and capturing document structures (He et al., 2024; Cao and Wang, 2022; Liu et al., 2022; Bai et al., 2021; Zaheer et al., 2020; Beltagy et al., 2020). He et al. (2024) and Cao and Wang (2022) use section structure to represent document hierarchy, while Liu et al. (2022) employs hierarchical sparse attention and special tokens for encoding local and global information. Bai et al. (2021) propose the encoding of the positional information of various linguistic segments. Beltagy et al. (2020) present Longformer, an LLM that employs a combination of local windowed attention and global attention, for processing long documents. Similarly, Zaheer et al. (2020) introduce BigBird, a model that leverages a sparse attention mechanism combining global, local, and random attention patterns, to handle longer sequences.

While these approaches attempt to incorporate structural information similarly to our work, they often rely on document-specific elements such as section titles, which are not present in argument structures, while generic approaches typically use token-level attention mechanisms. Our method differs by focusing on argument-specific features (argument flow) and attention mechanisms to model interactions between high-level linguistic units (ADUs) and their functional relationships, while simultaneously capturing logical segments of argument (local-structures).

3 Macro-Structure

An argument is a coherent arrangement of utterances organised in a specific order (Grosz and Sidner, 1986; Toulmin, 1958; Freeman, 2011). Freeman (2011) propose a framework describing how these utterances collectively contribute to natural language argumentation, particularly focusing on their supportive roles and structural patterns, termed as "macro-structure". This framework encompasses techniques such as divergent, convergent, linked, and serial reasoning, which illustrate how reasons combine to support conclusions. It underscores the significance of understanding the entire sequence of ideas within an argument, including claims, challenges, responses, and counter-responses, to establish coherent structure.

Coherence within discourse can be viewed at two levels: **local coherence** and **global coherence**. Local coherence refers to coherence among the utterances in a segment of an argument, while global coherence refers to the coherence spanning segments (Grosz and Sidner, 1986; Grosz et al., 1995). Grosz and Sidner argue that the coherence depends on the intentional structure of discourse addressed via the overall DP and DSP (Grosz and Sidner, 1986; Grosz et al., 1995). These intentions are reflective of the speaker's goals, akin to Gricean conversational implicatures (Grice, 1975). In a

multi-party discourse, the DSP for a given segment aligns with the intention of the conversational participant initiating that segment (Lochbaum, 1994). Freeman (2011) models these interactions as the interplay between the proponents and opponents, showing how proponents assert and address opponents' challenges, forming a chain of reasoning and highlighting the importance of tracing these transitions for understanding the argument.

IAT (Budzynska and Reed, 2011) offers a framework representing how argument structure is linked to the intentional structure and the dynamics within dialogue structure. In essence, IAT offers a macro-structural analysis by representing the intentional structure and illocutionary dynamics within argumentative discourse, by linking dialogical moves to their communicative intentions and illocutionary forces. For example, Figure (1b) illustrates participant interactions alongside argument structures, showcasing diverse dialogue moves such as "Asserting", "Arguing", "Questioning", "Illocuting", and "Restating" (Budzynska and Reed, 2011). Annotated corpora, such as the corpus of US presidential debate 2016 (Visser et al., 2019) annotated following such framework, exemplify how dialogical interactions unfold as a series of moves, each mapped to a structural element within the argument graph. Although these dynamics are common in dialogue, similar conceptualisations apply to monologue, where a speaker delivers multiple utterances to an audience (Grosz et al., 1995).

4 Method

4.1 Data

Heterogeneous datasets encompassing various domains and genres are utilised, including student persuasive essay corpora (AAEC) (Stab and Gurevych, 2017), argumentative micro text (MTC) (Peldszus and Stede, 2013), the US 2016 presidential debate corpus (US2016) (Visser et al., 2019), and a corpus of argument and conflict in broadcast debate (QT30) (Hautli-Janisz et al., 2022). The AAEC and MTC, are monolingual, while the US2026 and QT30 are dialogical. AAEC consists of student essays and MTC is created through a controlled text generation experiment, whereas US2026 and QT30 are derived from real-world discussions.

4.2 Local-Structure Annotation

The four datasets are automatically annotated to identify local-structure for each AR. We traverse argument structure as a graph to identify segments (local-structures) containing each AR, with ADUs and ARs represented as nodes connected by edges. It involves both upward and downward traversals from the AR node. Given an AR, the upward traversal identifies the chains of ADUs leading to the AR, capturing the local coherence that builds up to the current AR. The downward traversal identifies the chain of ADUs following the AR, ensuring the continuity of the argument.

The beginning of a local-structure is marked by a node that lacks inward connections (start ADU), signifying the starting point of the segment and the end is marked by a node without successors (end ADU), indicating the conclusion of the segment. If the start ADU involves multiple downward chains (divergent structure), all such chains are included. Moreover, every sub-graph including serial, divergent, convergent and linked structures between the start and end ADUs are included to ensure a complete and coherent segment (additional information can be found in the Appendix C).

4.3 Data Study

Table 1 shows the summary of the datasets statistics. Accordingly, 73% of the argument structures involve more than one local-structure, where 67% of those involves 2 to 7 local-structures, with the exception of MTC. The lack of local-structures in MTC is expected since the arguments are short and addressing defined argument intentions (an argument has about 5 ADUs and one of the ADUs serves as the central claim). Across the dataset, 64% of AR are between ADUs located 1 to 5 distance away from each other. Distance refers to the difference between the positions of the two ADUs in the argument. Notably, 17% of the AR are between ADUs within a distance of 1.

4.4 Model

An argument is represented with a unified model that encodes both micro and macro-structures (Section 4.4.1). This unified representation is utilised for AR prediction within a joint multi-task learning framework, where local-structure identification serves as an auxiliary task to provide additional macro-structural context (Section 4.4.2).

4.4.1 Argument Representation

The unified argument representation is achieved by combining two types of embeddings: ADU embeddings, which capture the micro-structure of the

Dataset	No_arg	No_ADU	No_RA	No_CA	Loc_struct	Dist_ARs
AAEC	402	6089	4841	497	3.3	2.6
MTC	112	576	272	171	1.2	1.3
US2016	499	8610	2830	942	5.1	3.2
QT30	724	11266	2756	558	7	4.8

Table 1: Summary of dataset showing the number of arguments (No_arg), the average number of ADUs within each argument (No_ADU), the number of support (No_RA), attack (No_CA), the average number of local-structures (Loc_struct), and distance between ADUs involving AR (Dist_ARs).

argument, and argument flow embeddings, which capture the macro-structure.

ADU Embedding. Each ADU is represented by a fixed-size embedding of dimension d. A pre-trained LLM generates contextualised embeddings for the argument, where ADUs are separated by the special token `[SEP]`. Mean pooling is applied to token embeddings from the final output layer to obtain each ADU's representation, with the `[SEP]` token marking ADU boundaries (see Section B.1 for additional details).

Argument Flow Embedding. ADU order and proponent-opponent transitions are leveraged to encode argument flow. Both absolute (Vaswani et al., 2017) and relative Shaw et al. (2018) positional embeddings are explored for this purpose. Absolute positional embedding follows the sine-cosine technique proposed in Transformers (Vaswani et al., 2017), generating an $n \times d$ matrix, where n is the number of ADUs and d is the embedding size. Relative positional embeddings follow the approach introduced by Shaw et al. (2018), capturing the relative distances between ADU positions using learnable embeddings e.

For the ADU order based embedding, positional index of each ADU, reflecting their sequential order within the argument is used. Unlike the standard positional embedding which utilise token positions, our approach uses ADU positions. It is also worth noting that the embedding used to represent ADUs (at the ADU embedding step) is derived from a pre-trained LLM, which already incorporates token-based positional embeddings. For proponent-opponent transition embeddings, each ADU is assigned a unique numerical index representing the participant making the ADU, capturing participants transitions. Each participant is assigned a unique index ranging from 0 to $n-1$, where n is the total number of participants. In multi-participant dialogues, ADUs are tagged with these indices to track shifts between participants while in single-participant monologues, all ADUs are assigned the same index.

Unified Argument Representation. Let A represent an argument consisting of n ADUs, denoted as $A = \{a_1, a_2, \ldots, a_n\}$, where a_i denotes the i-th ADU. The unified embedding of each ADU_i integrating argument-flow information is computed as:

$$\mathbf{ADU}' = \mathbf{ADU}_i \oplus \mathbf{O}_i \oplus \mathbf{P}_i \quad (1)$$

where $\mathbf{ADU}_i$ is the ADU embedding, $\mathbf{O}_i$ is the order embedding, $\mathbf{P}_i$ is the proponent-opponent transition embedding and $\oplus$ is the fusion operation integrating the argument flow. This results in an $n \times d$ matrix, where n is the number of ADUs and d is the embedding size.

In the absolute positional embedding, the fusion operation is the summation of the embeddings to obtain $\mathbf{ADU}'_{\text{abs}}$, and a multi-head attention mechanism is applied to capture contextual dependencies between ADUs. In contrast, the relative positional embedding integrates argument flow embeddings dynamically during the attention score ($\mathbf{R}$) computation as follows:

$$\mathbf{R} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{O} + \mathbf{P}\right) \quad (2)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value embeddings of the ADUs, and $\mathbf{O}$ and $\mathbf{P}$ are the order and proponent-opponent transition embeddings, respectively. In both approaches, the multi-head attention computes cross-attention scores between ADUs, since each position in the attention matrix represents an ADU. Further details on both strategies are provided in Section B.2.

4.4.2 Predicting Argument Relations

AR prediction is framed within a joint multi-task learning, where one task identifies the local-structure, and the other predicts the AR. Given $\mathbf{A}'$ representing the unified argument representation, and $\mathbf{ADU}_i$, $\mathbf{ADU}_j$ denoting the pair of ADUs

under consideration, the input to the model can be represented as $\{\mathbf{A}', (\mathbf{ADU}_i, \mathbf{ADU}_j)\}$ (see A.2). The output from the cross-attention mechanisms on $\mathbf{A}'$ is combined with the unified embeddings of $\mathbf{ADU}_i$, $\mathbf{ADU}_j$ and processed through a feed-forward network to predict AR and local-structures (see Appendix B.2 for more details). The outputs for both tasks are given by:

$$\text{Output} = \text{LS-cls}(\mathbf{H}) \oplus \text{AR-cls}(\mathbf{H}) \quad (3)$$

where H represents shared model parameters.

The AR-cls is a classification layer predicting AR and LS-cls is a token classification layer identifying the relevant local-structure. For the LS-cls, the BIO (Beginning, Inside, Outside) labeling scheme (Ramshaw and Marcus, 1999) is used. Each token in the argument is assigned one of the three BIO labels to predict the ADUs constituted by the respective local-structure described in Section 4.2. Both tasks share the same input and model parameters, and the overall loss function is the sum of the losses from both tasks.

Baselines. Two baseline configurations are evaluated: vanilla sequence-pair classification (V-SeqCls), which fine-tunes various LLMs on concatenated ADU pairs, and vanilla argument context (V-ArgC), which uses the entire argument along with ADU pairs.

5 Experiment

5.1 Training setup

The models train for 6 epochs, with a batch size of 16 samples. Adam optimisation (Kingma and Ba, 2014) is used with a learning rate of 2×10^{-5} and categorical cross-entropy loss to minimise the cost function. Results represent the average of three runs using different random seeds. More experimental setup provided in the Appendix A. The code used in this work can be publicly accessed at `https://github.com/ANONYMOUS` (redacted).

5.2 Evaluation Setup

Two evaluation setups are employed to assess the model's robustness in the AR prediction task.

In-Dataset Evaluation: Each dataset is divided into 70% training, 20% test, and 10% validation sets.

Cross-dataset Evaluation: A model is trained on the combination of $n-1$ data sources and evaluated on the remaining data source, repeated n times for each data source. This setup aims to evaluate the model's performance on unseen data source, providing insights into its adaptability and robustness across different datasets.

In both settings, macro precision (P), recall (R), and F-measure (F) are reported for the test dataset. Please note that the local-structure prediction task is not evaluated, as it serves as an auxiliary task.

5.3 Model configurations

Pre-trained LLMs are fine-tuned based on the argument representation presented in Section 4.4.1 using the multi-task setting described in Section 4.4.2. Diverse LLMs are explored to evaluate their efficacy in leveraging the macro-structural features: BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), Big Bird (Zaheer et al., 2020), and DialoGPT (Zhang et al., 2020). BERT, RoBERTa, and Big Bird are pre-trained on generic datasets, while DialoGPT is pre-trained on dialogical datasets for capturing dialogue dynamics. Big Bird (Zaheer et al., 2020) extends Transformers with sparse attention by integrating global, local, and random patterns, enabling it to handle longer sequences and tasks requiring long context. Evaluating these models with and without the MSR-RAM configuration establishes robust baselines, with DialoGPT and Big Bird standing out as particularly strong baselines, given the relevancy of pre-training strategies and datasets (see Appendix A.3 for more details).

5.4 Results

The findings underscore the role of incorporating macro-structural features in enhancing the performance of AR identification. Table 2 shows the performance of each configuration, the baselines and comparison systems across the datasets. As can be seen from the Table, MSR-RAM configurations consistently surpasses the baselines both in in-dataset and cross-dataset evaluation settings. Specifically, MSR-RAM configuration outperforms both V-SeqClas and V-ArgC configurations, with an average performance improvement of 9.2%, 8.5% on in-dataset and 10%, 9.3% on cross-dataset evaluations, respectively.

The cross-dataset evaluation result demonstrate the ability of MSR-RAM configurations in learning transferable skills across datasets. As can be seen in Table 2, in contrast to MSR-RAM, the vanilla configurations struggle to exceed random chance performance in cross-dataset evaluation settings. Note that the cross-dataset evaluation in Table 2 involves training on $n-1$ data sources and test-

LLM	Model	In-dataset					Cross-dataset				
		AAEC	MTC	US16	QT30	AVG	AAEC	MTC	US16	QT30	AVG
Comparison	**Bao et al. (2022)**	50	n/a	n/a	n/a	50	n/a	n/a	n/a	n/a	n/a
	Peldszus and Stede (2016)	n/a	53	n/a	n/a	53	n/a	n/a	n/a	n/a	n/a
	Eger et al. (2017)	51	n/a	n/a	n/a	51	n/a	n/a	n/a	n/a	n/a
	Gemechu and Reed (2019)	77	75	62	n/a	71	n/a	n/a	n/a	n/a	n/a
	Morio et al. (2022)	55	58	n/a	n/a	57	n/a	n/a	n/a	n/a	n/a
	GPT-4	63±2	48±2	55±2	60±2	57±2	n/a	n/a	n/a	n/a	n/a
BERT	**V-SeqCls**	67 ±0.0	61±0.2	66±0.0	67±0.2	65±0.1	43±0.3	30±0.3	33±0.2	36±0.1	36±0.2
	V-ArgC	68±0.1	61±0.1	69±0.0	67±0.1	66±0.0	45±0.3	31±0.2	32±0.1	38±0.3	37±0.1
	MSR-RAM^{abs}	79±0.1	78±0.2	76±0.1	77±0.1	78±0.1	50±0.3	42±0.2	48±0.2	50±0.2	48±0.0
	MSR-RAM^{rel}	79±0.2	79±0.1	78±0.1	78±0.2	79±0.1	51±0.3	43±0.3	40±0.3	53±0.3	50±0.1
Roberta	**V-SeqCls**	75±0.0	63±0.1	74±0.0	75±0.1	72±0.1	45±0.2	37±0.1	46±0.2	46±0.1	44±0.1
	V-ArgC	75±0.1	65±0.2	74±0.1	76±0.1	73±0.1	46±0.2	39±0.1	47±0.3	45±0.1	44±0.2
	MSR-RAM^{abs}	79±0.1	80±0.2	78±0.1	79±0.0	79±0.0	53±0.2	45±0.3	54±0.2	52±0.3	51±0.1
	MSR-RAM^{rel}	80±0.2	81±0.1	79±0.2	80±0.1	80±0.0	55±0.3	46±0.2	55±0.3	52±0.3	52±0.0
DGPT	**V-SeqCls**	76±0.1	63±0.0	73±0.0	74±0.1	72±0	46±0.0	39±0.1	47±0.0	45±0.0	44±0.0
	V-ArgC	75±0.0	67±0.1	75±0.0	74±0.1	73±0.0	47±0.1	40±0.0	48±0.1	46±0.1	45±0.0
	MSR-RAM^{abs}	80±0.0	81±0.1	79±0.0	80±0.0	80±0.0	56±0.1	49±0.2	55±0.1	53±0.2	54±0.1
	MSR-RAM^{rel}	81±0.1	82±0.2	80±0.1	81±0.0	81±0.1	57±0.2	51±0.2	55±0.1	53±0.2	54±0.2
Big Bird	**V-SeqCls**	78±0.1	65±0.0	76±0.0	78±0.1	74±0	48±0.0	41±0.1	49±0.0	46±0.0	46±0.0
	V-ArgC	77±0.0	66±0.1	77±0.0	77±0.1	74±0.0	47±0.1	41±0.0	48±0.1	47±0.1	46±0.0
	MSR-RAM^{abs}	81±0.0	82±0.1	81±0.0	81±0.0	82±0.0	60±0.1	53±0.2	55±0.1	54±0.2	56±0.1
	MSR-RAM^{rel}	83±0.1	82±0.2	81±0.1	82±0.0	82±0.1	59±0.2	51±0.2	56±0.1	55±0.2	55±0.2

Table 2: In-dataset and cross-dataset evaluation performance of MSR-RAM, baselines and the comparison systems.

ing on the remaining one. Detailed cross-dataset performance, where models are trained on individual datasets and evaluated on others, is provided in Table 5 in the Appendix. As shown in Table 5, the MSR-RAM configurations demonstrate competitive performance in cross-dataset evaluation, comparable to state-of-the-art results reported in in-dataset evaluation. Notably, Big Bird based MSR-RAM, trained on QT30 but evaluated on AAEC, MTC achieve comparable result to the SOTA models trained and evaluated on AAEC and MTC.

The MSR-RAM configuration is compared against related works including (Eger et al., 2017), (Peldszus and Stede, 2016), (Gemechu and Reed, 2019), (Morio et al., 2022), (Bao et al., 2022) and OpenAI's GPT-4 (OpenAI, 2023). Eger et al. (2017) investigate a multi-task setup to exploit the dependency between component identification and AR prediction, achieving an F1-score of 51 for AR identification on AAEC dataset. Peldszus and Stede (2016) aim to map RST trees to argumentation structures (Taboada and Mann, 2006) using sub-graph matching and an evidence graph model and achieve an overall F-measure of 53 in identifying ARs on MTC dataset. Morio et al. (2022) introduces a multi-task architecture built on Longformer, integrating task-specific classifiers based on biaffine model to identify and classify argument spans and determine AR between them. Bao et al. (2022) presents a framework for end-to-end AM that uses CPM to define ADU boundaries and categories, and RPE to correct order biases in autoregressive models. To evaluate GPT-4 on the AR prediction task, a few-shot prompting approach is used. Details of the experimental setup and the prompt template are provided in Section B.4.

As can be seen in Table 2, our approach outperforms all the comparison systems. Please note that direct comparisons between our approach and the comparison systems must be interpreted in context due to differences in task setup; for example, Eger et al. (2017), (Morio et al., 2022), and (Bao et al., 2022) combine argument segmentation with AR identification, while the work of Gemechu and Reed (2019) and MSR-RAM focus solely on AR identification based on correctly segmented ADUs.

LLMs show consistent performance improvements across architectures when using MSR-RAM configurations. DialoGPT based MSR-RAM surpass the standard LLMs, with the exception of Big Bird. This improvement is expected due to its pre-training on dialogical data, demonstrating the advantages of more relevant task-specific pre-training compared to generic datasets. Configurations utilising Big Bird outperform all other LLMs, suggesting its ability to capture both local and global context as claimed by the authors. It is worth noting that Big Bird with the MSR-RAM configuration shows a significant performance boost, averaging 7.4% and 9.5% in in-dataset and cross-dataset evaluations, respectively, compared to Big Bird without this configuration.

5.5 Error Analysis

We analyse the error types observed in MSR-RAM versus the baseline. Analysis of 50 argument maps showed that 53% of the baseline's miss-classification occur within the same local-structure, compared to only 12% for MSR-RAM. This indicates a 77.4% reduction in miss-classifications within the local-structure, reflecting a better adherence to argument flow. Unlike NLI, AM requires coherence constraints that prevent evaluating ARs in isolation. Thus, while ADU pairs might involve ARs, the coherence can invalidate ARs that disrupt the overall argumentative flow. A common error, termed **jump to conclusion Error**, occurs when an AR is formed by skipping necessary intermediate ARs. This happens when an ADU A is incorrectly linked directly to ADU C, despite A and C being connected through an intermediary ADU B. For instance, as can be seen from the output of the baseline result in Appendix 4, while AR (19) might be considered as valid AR, the coherence enforced by AR (2) role in the argument could invalidate it. Errors of this type account for 14% in MSR-RAM compared to 56% in the baseline.

5.6 Ablation study

Config	Monologue	Dialogue	Average
Full (Abs)	80	79	80
Full (Rel)	81	80	81
P^- (Abs)	70	75	73
P^+ (Abs)	74	77	76
P^- (Rel)	70	76	73
P^+ (Rel)	74	78	76
O^- (Abs)	75	76	76
O^+ (Abs)	79	79	79
O^- (Rel)	77	78	78
O^+ (Rel)	81	79	80

Table 3: In-dataset F-1 scores for configurations with absolute (Abs) and relative (Rel) positional embeddings on monological and dialogical datasets.

Config	Monologue	Dialogue	Average
L^{ID}	74 (69)	76 (73)	75 (71)
L^{CD}	47 (41)	48 (44)	48 (43)

Table 4: F1-scores for configurations without local-structure and (baseline) on monological and dialogical datasets, in in-dataset (ID) and cross-dataset (CD) evaluations. Baseline results are shown in parentheses.

The impact of each macro-structural feature on the performance of the models is analysed.

Argument flow: As can be seen from Table 3, both order embedding (O) and proponent-opponent transition embedding (P) prove to be effective, with O surpassing P, while their fusion achieve a new SOTA performance. On average, this fusion attains 6.1%, 11.6% performance increase in dialogical and monological datasets on in-dataset evaluation settings. The performance gain is calculated as the difference between the average F1-scores of the positional embeddings (representing SOTA) and the average F1-scores of V-SeqCls and V-ArgC (representing the baseline), across the configurations. To isolate their individual effects, we assess O and P independently, with (+) and without (-) the inclusion of local-structures, for both absolute (Abs) and relative (Rel) positional embeddings. O contributes to an average 8.4% and 5.2% enhancement with and without local-structures, respectively, while P contributes to a 4.7% and 2% improvement with and without local-structures. Notably, P, prevalent in dialog settings, provide only marginal improved performance in monologue datasets.

Local-structure prediction: The findings underscore the significance of incorporating local-structure prediction in the multi-task setting. Table 4 shows that omitting the local-structure prediction sub-task results in a 4.5% decrease in overall performance. We observe that only 16% of errors originating from cross local-structures, when local-structure prediction is incorporated. This highlights the effectiveness of local-structure in preserving coherence both at local and global levels.

6 Conclusion

This study introduces MSR-RAM, a method to incorporate macro-structural features for robustness AM. By explicitly capturing both local and global-structural information, MSR-RAM outperforms existing methods across diverse datasets, achieving new SOTA result. MSR-RAM's result in cross-dataset evaluations highlights its effectiveness in learning generic, transferable features, marking an advancement in overcoming the challenge of domain adaptation in AM. Future research could delve into modelling more nuanced dialogical features to encode argument flow, including the incorporation of specific dialog moves such as "asserting", "arguing", and "questioning".

Limitations

Despite its merits, the MSR-RAM approach has the following limitations:

Limited Applicability to Other NLP Tasks: The proponent-opponent features and local-structure encoding are specifically designed for argumentation tasks. As such, their applicability to other NLP tasks that do not involve argumentative structures is limited.

Pre-Training Objectives Not Addressed: Although the evaluation focuses on fine-tuning for leveraging macro-structural features, it does not address the training objectives that could be employed during the pre-training phase of LLMs to better integrate these features.

Interpretability and Explainability: The explanations for the model's performance are based on empirical results, ablation studies, and error analysis. While these analyses are valuable, additional techniques such as attention mechanism analysis could provide a more comprehensive understanding of model behavior.

References

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

He Bai, Peng Shi, Jimmy Lin, Yuqing Xie, Luchen Tan, Kun Xiong, Wen Gao, and Ming Li. 2021. Segatron: Segment-aware transformer for language modeling and understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12526–12534.

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

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

Filip Boltužić and Jan Šnajder. 2016. Fill the gap! analyzing implicit premises between claims from online debates. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 124–133.

Katarzyna Budzynska and Chris Reed. 2011. Whence inference. *University of Dundee Technical Report*.

Shuyang Cao and Lu Wang. 2022. Hibrids: Attention with hierarchical biases for structure-aware long document summarization. *arXiv preprint arXiv:2203.10741*.

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

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

James B Freeman. 2011. *Dialectics and the macrostructure of arguments: A theory of argument structure*, volume 10. Walter de Gruyter.

Debela Gemechu and Chris Reed. 2019. Decompositional argument mining: A general purpose approach for argument graph construction. In *Proceedings of the 57st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351.

HP Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:43–58.

Barbara J Grosz, Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. 2022. Qt30: A corpus of argument and conflict in broadcast debate. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 3291–3300. European Language Resources Association (ELRA).

Haoyu He, Markus Flicke, Jan Buchmann, Iryna Gurevych, and Andreas Geiger. 2024. Hdt: Hierarchical document transformer. *arXiv preprint arXiv:2407.08330*.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2024. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

John Lawrence, Chris Reed, Colin Allen, Simon McAlister, and Andrew Ravenscroft. 2014. Mining arguments from 19th century philosophical texts using topic based modelling. In *Proceedings of the First Workshop on Argumentation Mining*, pages 79–87.

Yang Liu, Jiaxiang Liu, Li Chen, Yuxiang Lu, Shikun Feng, Zhida Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2022. Ernie-sparse: Learning hierarchical efficient transformer through regularized self-attention. *arXiv preprint arXiv:2203.12276*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Karen Elizabeth Lochbaum. 1994. *Using collaborative plans to model the intentional structure of discourse*. Harvard University.

Jim Mackenzie. 1990. Four dialogue systems. *Studia logica*, 49:567–583.

William C Mann. 2002. Dialogue macrogame theory. In *Proceedings of The Third SIGdial Workshop on Discourse and Dialogue*, pages 129–141.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

Marie-Francine Moens, Eric Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM.

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

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012a. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING 2012*, pages 1883–1900.

Philippe Muller, Stergos D. Afantenos, Pascal Denis, and Nicholas Asher. 2012b. Constrained decoding for text-level discourse parsing. *Proceedings of COLING 2012*, pages 1883–1900.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

Andreas Peldszus and Manfred Stede. 2013. Ranking the annotators: An agreement study on argumentation structure. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 196–204.

Andreas Peldszus and Manfred Stede. 2015a. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948.

Andreas Peldszus and Manfred Stede. 2015b. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948.

Andreas Peldszus and Manfred Stede. 2016. Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the Third Workshop on Argument Mining*, pages 103–112.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2016. Here's my point: Joint pointer architecture for argument mining. *arXiv preprint arXiv:1612.08994*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. *arXiv preprint arXiv:1906.09821*.

Ramon Ruiz-Dolz, Jose Alemany, Stella M Heras Barberá, and Ana García-Fornes. 2021. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intelligent Systems*, 36(6):62–70.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

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

Maite Taboada and William Mann. 2006. Rhetorical structure theory: Looking back and moving ahead. *Discourse studies*, 8(3):423–459.

Stephen Toulmin. 1958. The uses of argument, cambridge univ.

Lisa deMena Travis. 1984. *Parameters and effects of word order variation*. Ph.D. thesis, Massachusetts Institute of Technology.

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

Maria P.G. Villalba and Patrick Saint-Dizier. 2012. Some facets of argument mining for opinion analysis. *COMMA*, 245:23–34.

Jacky Visser, Barbara Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. 2019. Argumentation in the 2016 us presidential elections: annotated corpora of television debates and social media reaction. *Language Resources and Evaluation*, pages 1–32.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2019. Encoding word order in complex embeddings. *arXiv preprint arXiv:1912.12333*.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

A Experiment Setup

A.1 Training Procedure

Hyper-parameters: We employ Adam optimisation (Kingma and Ba, 2014) to minimise the cost function, using a learning rate of 2×10^{-5} and categorical cross-entropy loss and a batch size of 16.

Gradient Clipping: To prevent exploding gradients during training, we applied gradient clipping. We used a maximum gradient norm (`max_grad_norm`) parameter to determine the threshold for gradient clipping.

Warm-up and Learning Rate Schedule: We employ a linear warm-up strategy for the learning rate. The number of warm-up steps is set to 10% of the total training steps. Following the warm-up phase, the learning rate schedule is determined by a lambda function. This function linearly increases the learning rate during the warm-up phase and decreases it linearly thereafter.

A.2 Input Setup

Except the V-SeqClas configurations, the entire argument along with the pair of ADUs is provided to the model.

The **Input Format:**"{Argument} [EG] {premise} [SEP] {conclusion}", where Argument = {ADU1 [SEP] ADU2 [SEP] ... ADUn}, with n representing the number of ADUs in the argument.

Extracting Relevant Argument: For configurations requiring the entire argument as input, if the input length exceeds the maximum sequence length of the underlying LLM, we extract a relevant span of the argument pertaining to the premise and conclusion. The process is detailed as follows:

1. **Tokenisation:** Tokenise the argument, premise and conclusion using a tokeniser.

2. **Total Length Calculation:**

   - Compute the total token length, including special tokens.
   - Sum the lengths of premise tokens, conclusion tokens, argument tokens, and special tokens (`[CLS]` and `[SEP]`).

3. **Span Selection:**

   - If the total length is within the maximum sequence length of the LLM , concatenate the entire argument with the premise and conclusion.
   - If exceeding the maximum length:
     - Locate the positions of the premise and conclusion within the argument.

- Select a span that includes both the premise and conclusion with additional surrounding context, ensuring the total length remains within the limit.
- If including the span involving both the premise and conclusion exceed the maximum limit, start with the premise, expand the span towards the conclusion until the size constraint is met, and append the conclusion to the argument span.

4. **Concatenation:** Construct the final input text by concatenating the selected argument span with the premise and conclusion.

Maximum Number of ADUs in an Argument: We set the maximum number of ADUs to 128 for computational efficiency. This limit is sufficient, as no argument in the dataset exceeds this number of ADUs.

A.3 Selecting Pre-Trained LLMs

We experimented with various LLM types with varying architecture, and nature of data used during pre-training to investigate their effectiveness in leveraging macro features. Accordingly, we categorise the models based on the following factors. (A) Context size: local context, and global context, (B) pre-training data-set genre: generic data-set, dialogical data-set. We then select BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), Big Bird (Zaheer et al., 2020), and DialoGPT (Zhang et al., 2020). BERT, RoBERTa and Big Bird are pre-trained on generic datasets while DialoGPT is pre-trained on dialogical datasets. Big Bird (Zaheer et al., 2020) is designed to handle long input sequences by incorporating both local and global attention mechanisms. Unlike standard LLMs that process shorter texts, it is optimised for documents with higher linguistic structures. DialoGPT and Big Bird establish strong baselines for this task. DialoGPT is pre-trained on a dialogical dataset, which is particularly relevant for argumentation tasks, as opposed to generic datasets. Big Bird's effective local and global attention mechanisms are also highly pertinent, given that ARs are influenced by both local and macro structural features.

Both for the baselines and the MSR-RAM configurations, we utilise the HuggingFace implementation of BERT[1], RoBERTa[2], DialoGPT [3] and Big Bird [4]. In the baseline setup (both with and without argument context), we fine-tune the models based on the output of the [CLS] token from the final layer.

B MSR-RAM Architecture

Our proposed model represents an argument based on the embeddings of the ADUs and argument flow (via order embedding and opponent-proponent transition embedding), building upon the principles of the Transformer architecture (Vaswani et al., 2017). It extends the standard Transformers block with additional layers designed to learn the dependencies among ADUs via ADU level attention mechanisms. The architecture consists of three main components: ADU embedding, argument flow embedding, and multi-head attention mechanism. Each component is described in detail below.

B.1 ADU Embedding

We utilise pre-trained LLMs (Devlin et al., 2018; Liu et al., 2019; Radford et al., 2019; Zhang et al., 2020) to obtain contextualised token embeddings $\mathbf{H} \in R^{n \times d}$ for the entire input where n is the input length and d is the hidden size of the model. ADUs are identified within the sequence using the special separator token (`[SEP]`). To obtain embeddings for each ADU, we apply mean pooling over the token embeddings within each ADU. Let $\mathbf{H}_i \in R^{l_i \times d}$ represent the token embeddings for the i-th ADU, where l_i is the length of the i-th ADU. The ADU embedding $\mathbf{ADU}_i \in R^d$ is computed as:

$$\mathbf{ADU}_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \mathbf{H}_{i,j}$$

The resulting set of ADU embeddings forms a matrix $\mathbf{A} \in R^{m \times d}$, where m is the number of ADUs.

B.2 Argument Flow Embedding and Multi-Head Attention

To capture the structural relationships between ADUs, we introduce a custom attention mechanism

[1] https://huggingface.co/docs/transformers/en/model_doc/bert

[2] https://huggingface.co/docs/transformers/en/model_doc/roberta

[3] https://huggingface.co/docs/transformers/en/model_doc/dialogpt

[4] https://huggingface.co/docs/transformers/en/model_doc/big_bird

that incorporates argument flow embedding. We experiment with both fixed and relative positional embeddings. For absolute positional embeddings, we employ the sinusoidal position signal, following the approach introduced by the Transformer model (Vaswani et al., 2017). For relative positional embeddings, we adopt the method proposed by Shaw et al. (2018), which encodes the relative distances between ADU in the argument, $a_{ij} = e_{j-i}$, where e represents the learnable embeddings and $j - i$ indicates the relative distance between ADU j and ADU i. We leverage dual positional embeddings to incorporate the two types of positional information: the index representing the order of each ADUs within the argument (ADU order embedding) and the proponent-opponent transition embedding. Both approaches are further explained below.

Absolute Positional Encoding. As illustrated in Figure 1, the embedding of an ADU, denoted as $\mathbf{ADU}_i$, is enhanced with absolute positional information by incorporating both order embeddings and transition embeddings. This process involves the following steps:

1. **Sinusoidal Function for Embeddings:** Consistent with the approach used in standard Transformers, sinusoidal functions are employed to generate embeddings for argument flow ($\mathbf{T}_i$) based on both ADU order ($\mathbf{O}_i$) and proponent-opponent transitions ($\mathbf{P}_i$):

$$\mathrm{T}_{(index,2i)} = \sin\left(\frac{index}{10000^{2i/d_{\text{model}}}}\right)$$

$$\mathrm{T}_{(index,2i+1)} = \cos\left(\frac{index}{10000^{2i/d_{\text{model}}}}\right)$$

where $index$ denotes the position of the ADU and d_{model} is the dimensionality of the model. This method applies to both ADU order and proponent-opponent transition embeddings, providing a unified approach for incorporating positional information.

2. **Unified Representation of ADUs:** Each ADU is represented by fusing its ADU embedding ($\mathbf{ADU}_i$), order embedding ($\mathbf{O}_i$), and proponent-opponent transition embedding ($\mathbf{P}_i$) to form a unified representation of ADU ($\mathbf{ADU}'_{\text{abs}}$) . A matrix $\mathbf{A}_{\text{abs}}$ of size $n \times d$ is formed, where n is the number of ADUs in the argument and d is the embedding dimension:

$$\mathbf{A}_{\text{abs}} = \text{ADU}_i + \mathbf{O}_i + \mathbf{P}_i$$

3. **Multi-Head Attention Computation:** Multi-head attention is applied to the unified embedding matrix $\mathbf{A}_{\text{abs}}$ to capture dependencies and relationships among ADUs within the argument:

$$\mathbf{Q} = \mathbf{W}_Q\mathbf{A}_{\text{abs}}, \quad \mathbf{K} = \mathbf{W}_K\mathbf{A}_{\text{abs}}, \quad \mathbf{V} = \mathbf{W}_V\mathbf{A}_{\text{abs}}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable weight matrices.

The attention scores $\mathbf{A}'_{\text{abs}}$, which incorporate both the ADU embeddings and the argument flow information, are calculated as:

$$\mathbf{A}_{\text{abs}'} = \frac{\mathbf{Q} \cdot \mathbf{K}^\top}{\sqrt{d_k}}$$

Softmax is applied to the attention scores $\mathbf{A}'_{\text{abs}}$ to obtain attention probabilities $\mathbf{P}_{\text{abs}}$:

$$\mathbf{P}_{\text{abs}} = \text{softmax}(\mathbf{A}'_{\text{abs}})$$

The final output $\mathbf{Out}_{\text{abs}}$ of the attention mechanism is computed by weighting the values $\mathbf{V}$ with the attention probabilities $\mathbf{P}_{\text{abs}}$:

$$\mathbf{Out}_{\text{abs}} = \mathbf{P}_{\text{abs}}\mathbf{V}$$

Relative Positional Encoding. The attention mechanism adjusts the attention scores $\mathbf{A}'_{i,j}$ to integrate relative distances on the fly:

$$\mathbf{A}'_{i,j} = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_j^\top}{\sqrt{d_k}} + \mathbf{R}^{\text{O}}_{i,j} + \mathbf{R}^{\text{P}}_{i,j}\right)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the query, key, and value matrices, respectively, derived from the ADUs embeddings.

$\mathbf{R}^{\text{O}}_{i,j}$ represents the embeddings of the relative order information and is given by,

$\mathrm{R}^{\text{O}}_{i,j} = \mathbf{W}^{\text{O}}(\text{pos}_i - \text{pos}_j)$

where $\mathbf{W}^{\text{O}}$ is the learnable weight matrix for ADU positions, and O_i and O_j are the index reflecting the order of the ADUs i and j within the argument.

$\mathbf{R}^{\text{P}}_{i,j}$ represents the relative embeddings for proponent-opponent transitions and is given by,

$\mathrm{R}^{\text{P}}_{i,j} = \mathbf{W}^{\text{P}}(\mathrm{P}_i - \mathrm{P}_j)$

where $\mathbf{W}^{\text{P}}$ is the learnable weight matrix for turn number, and P_i and P_j are the turn numbers of ADUs i and j within the argument.

The attention output is then computed as:

$$\mathbf{Out} = \sum_j \mathbf{A}'_{i,j}\mathbf{V}_j$$

where $\mathbf{Out}$ is the output of the attention layer.

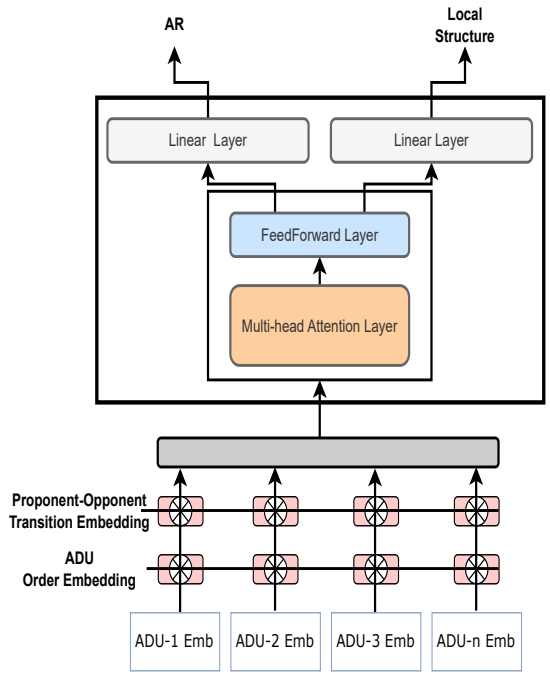


Figure 1: Architecture for capturing both micro and macro argument structures using absolute positional embeddings.

B.3 Feedforward Layers

The output of the attention mechanism $\mathbf{Out}$ is concatenated with the CLS token embeddings of the pair of ADUs (premise, conclusion), forming the combined representation:

$$\mathbf{C} = [\mathbf{Out}, \mathbf{CLS}]$$

The combined representation $\mathbf{C}$ is then fed into a feedforward neural network, which consists of two linear layers. The final layer outputs the classification logits $\mathbf{z}$:

$$\mathbf{z} = \text{FFN}(\mathbf{C})$$

where $\mathbf{z} \in R^k$ and k is the number of AR for AR detection (in sequence-classification fashion) and local-structure prediction (in token-level classification task fashion).

B.4 GPT for AR Prediction

B.4.1 Experimental Settings

We utilise the chat completion configuration of ChatGPT-4 for predicting AR.

1. **Configurations:** We use GPT-4 based on `gpt-3.5-turbo-instruct`. We set a maximum token limit of 2048, a temperature of 0.7, a top-p probability of 0.9.

2. **Prompts Strategy:** We explored two strategies: zero-shot and few-shot prompts. In the zero-shot setting, only instruction based prompts without examples are used. We also try few-shot setup, where specific examples are provided as part of the instruction. Interestingly, our analysis revealed that the example-based experiment achieved a higher score compared to the zero-shot prompt in the AR prediction. As a result, our experiment is based on example-based prompting. We create prompt templates that include instructions and two examples randomly selected from a list of examples.

Prompt Design for Zero-Shot AR Prediction: We prompt GPT-4 to classify the relationship between the ADUs as supporting, contradicting, or having no clear AR using the following prompt template.

```
You are a 3-class classifier model tasked with
assigning a label to the argument
relation between two argument units
(argument 1 and argument 2).
Classify the following  pair of arguments,
argument 1: {ADU_1}
argument 2: {ADU_2},
into:
"support" (if argument 1 supports
argument 2),
"contradict" (if argument 1 attacks
argument 2),
and "None" (if no argument relation exists
between argument 1 and argument 2).
Please enter:
1 - for support,
2 - for contradict,
0 - for None relation.
Examples from each argument
relation types are provided below:
Example 1: the argument relation between
the argument "people feel, when they have
been voicing opinions on different matters,
that they have been not listened to", and
the argument "people
feel that they have been treated
disrespectfully on all sides of the
different arguments and disputes going on"
is support, and  hence prediction label is 1.
```

```
Example 2: The argument relation between
"there would be no non-tariff barriers
with the deal  done with the EU" and
the argument "there are lots of
non-tariff barriers
with the deal done with the EU"
is contradiction, and
hence prediction label is 2.
```

Note: We use the actual examples to show support and contradiction relations, which should be a placeholder variable in the final prompt template.

C Local Structures Extraction from Argument Map

Local-structures are segments of the argument map that represent coherent chains of ADUs leading to and following an AR. We present Algorithm 1 to outline the procedure for extracting local-structures from a global argument map. The algorithm takes as input the argument map represented as nodes and edges, where each node represents ADUs and the ARs. The relations between ADUs are presented based on the edges between the ADU and AR nodes.

The algorithm generates a comprehensive list of local-structures that are pertinent to the respective ARs within the overarching argument map. Each of these local-structures is identified and cataloged according to their relevance to specific AR in the argument map. For illustrative purposes, Figure 2 presents several examples showcasing argument maps that feature multiple local-structures. In these examples, the local-structures are annotated with numerical labels. Each number used for annotation corresponds to a distinct local-structure. ARs that share the same numerical label are part of the same local-structure.

The algorithm iterates through each ADU node in the argument map. It performs an upward traversal to identify the chain of ADUs leading to the AR node and a downward traversal to identify the chain of ADUs following the AR node. The algorithm marks the end of each local-structure in the upward traversal by identifying nodes without inward connections and in the downward traversal by identifying nodes without successors. It includes all chains of ADUs that end at the same node to form the local-structure.

D Cross-Dataset Evaluation

Cross-dataset evaluation involves training a model on one dataset and assessing its performance on the remaining three datasets. This approach is crucial for evaluating a model's ability to generalise and transfer its skills across different datasets. In argument mining, achieving consistent performance improvements through cross-dataset evaluation is particularly noteworthy. This is because models that perform well in in-dataset evaluations often show diminished performance when tested on different datasets, sometimes achieving results comparable to random chance.

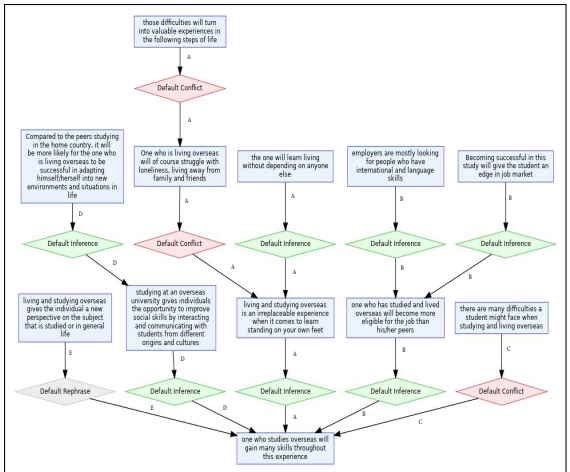
Table 5 provides a comprehensive overview of the cross-dataset evaluation results. It clearly demonstrates that the MSR-RAM configurations consistently outperform the baseline models by a significant margin. Notably, in some cases, the MSR-RAM configurations approach or even match the performance of state-of-the-art (SOTA) models when evaluated in a standard in-dataset setup. This indicates that MSR-RAM not only improves performance within the same dataset but also shows substantial effectiveness in transferring learned skills across diverse datasets.

Model	Data	AAEC	US16	QT30	MTC
BERT	AAEC	-	48	52	40
	US16	53	-	60	45
	QT30	52	57	-	44
	MTC	49	45	47	-
RoBERTa	AAEC	-	51	50	44
	US16	57	-	52	49
	QT30	58	63	-	47
	MTC	52	51	55	-
DialogPT	AAEC	-	53	50	47
	US16	60	-	61	54
	QT30	60	61	-	53
	MTC	56	48	47	-
BIG-BIRD	AAEC	-	55	52	49
	US16	63	-	64	52
	QT30	62	61	-	54
	MTC	52	53	49	-

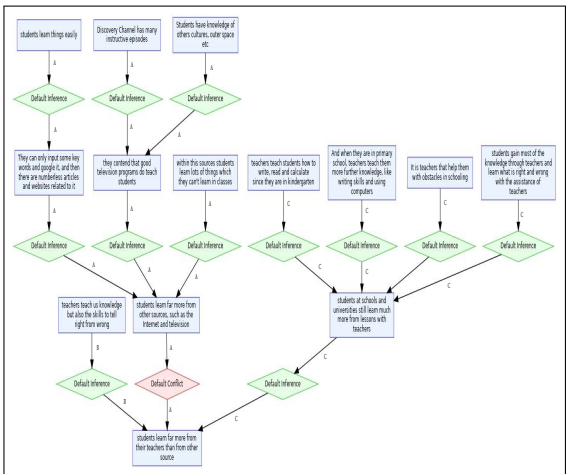
Table 5: Cross-dataset evaluation performance of **MSR-RAM^{rel}** trained on one dataset and evaluated on the other. The result presents the evaluation performance of MSR-RAM configuration based on relative positional embedding.

E Macro-structure

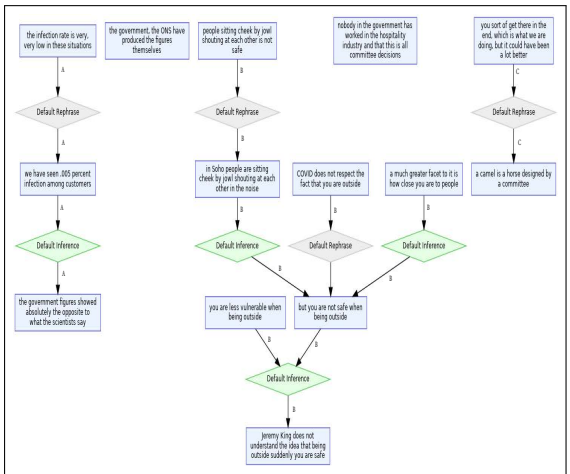
Example of argument annotated based on AIT showcasing argument dynamics along with the argument inferential structure is presented in Figure 3b. The map annotates each dialogue move with



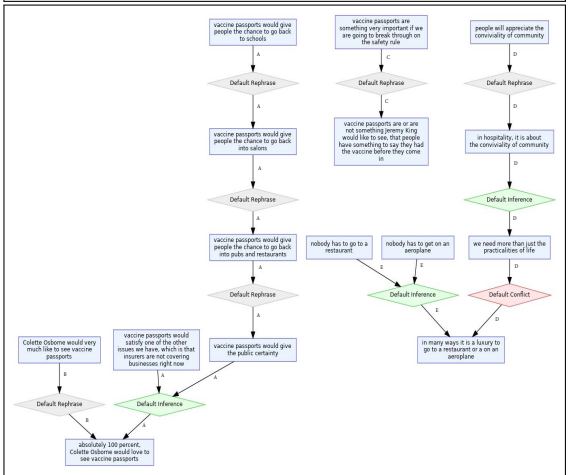
(a) An argument structure with 5 segments (A,B,C,D and E)



(b) An argument structure with 3 segments (A,B and C)



(c) An argument structure with 3 segments (A,B and C



(d) An argument structure with 5 segments (A,B,C,D and E)

Figure 2: An example of argument structures involving multiple segments. ADUs are logically interconnected via AR to form coherent argument structure. Figure (a) and (b) are taken from AAEC, while (c) and (d) are taken from QT30. As can be seen from the figure, (a) and (b) forms one complete graph while (c) and (d) are scattered into multiple disconnected graphs forming islands of argument segments.

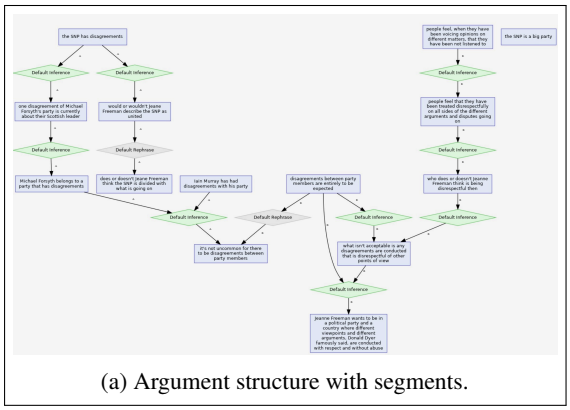
Algorithm 1 Extract Local-Structures from Argument Map

Require: Argument map represented as nodes and edges, with each node categorised as ADU, and AR
Ensure: List of local-structures
Initialise an empty list to store local-structures: $local\_structures$
Identify nodes corresponding to AR Nodes in the argument map
for each ADU Node in the argument map **do**
Perform an upward traversal to identify the chain of ADUs leading to the AR
Perform a downward traversal to identify the chain of ADUs following the AR Node
Mark the start of each local-structure in the upward traversal by identifying nodes without inward connections
Mark the end of each local-structure in the downward traversal by identifying nodes without successors
Include all chains of ADUs between the start and end node
Add the identified local-structure to $local\_structures$
end for
return $local\_structures$

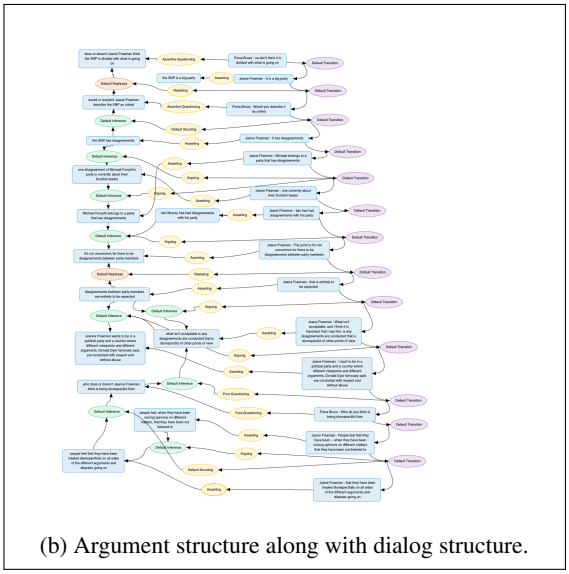
its corresponding illocutionary force (e.g., Assertions, Questions, Transitions, Illocuting and Arguing) and illustrates how these moves influence the inferential structure. Assertions are declarative statements made within the dialogue that contribute to the overall argument structure. In the IAT framework, assertions are mapped to inferential structures where they serve as propositions that provide content for the argument. Questions are interrogative acts that challenge or probe the content of assertions by prompting responses that substantiate or refute the initial assertions. Transitions in dialogue represent the movement from one communicative act to another and are essential for understanding how arguments develop dynamically. Arguing is an illocutionary act that involves presenting and defending an argument and mapped as a type of illocutionary scheme that connects propositions through logical relations. In this paper, given that our aim is to capture argument flow at high level, we employ the transitions between the participants only instead of leveraging the specific dialog moves used by the participants.

F Error Analysis

Figure 4 presents an example of an argument map generated by the baseline model. In this map, argument relations are labeled with numbers, and incorrect AR predictions are highlighted with an (x) symbol. The figure provides a visual representation of the errors made by the baseline model, allowing for a clearer understanding of the error types in AR predictions.



(a) Argument structure with segments.



(b) Argument structure along with dialog structure.

Figure 3: Argument structures annotated based on IAT (Budzynska and Reed, 2011). ADUs are logically interconnected via AR to form coherent argument structure. The left figure shows the interplay between local-structures addressing specific DSPs: one on the Scottish National Party's internal divisions and another on respectful disagreement. The right figure illustrates the interaction between participants alongside the argument structures. It demonstrates how transitions between different dialogue moves are linked to changes in the inferential structure, capturing the dynamic nature of argumentation.

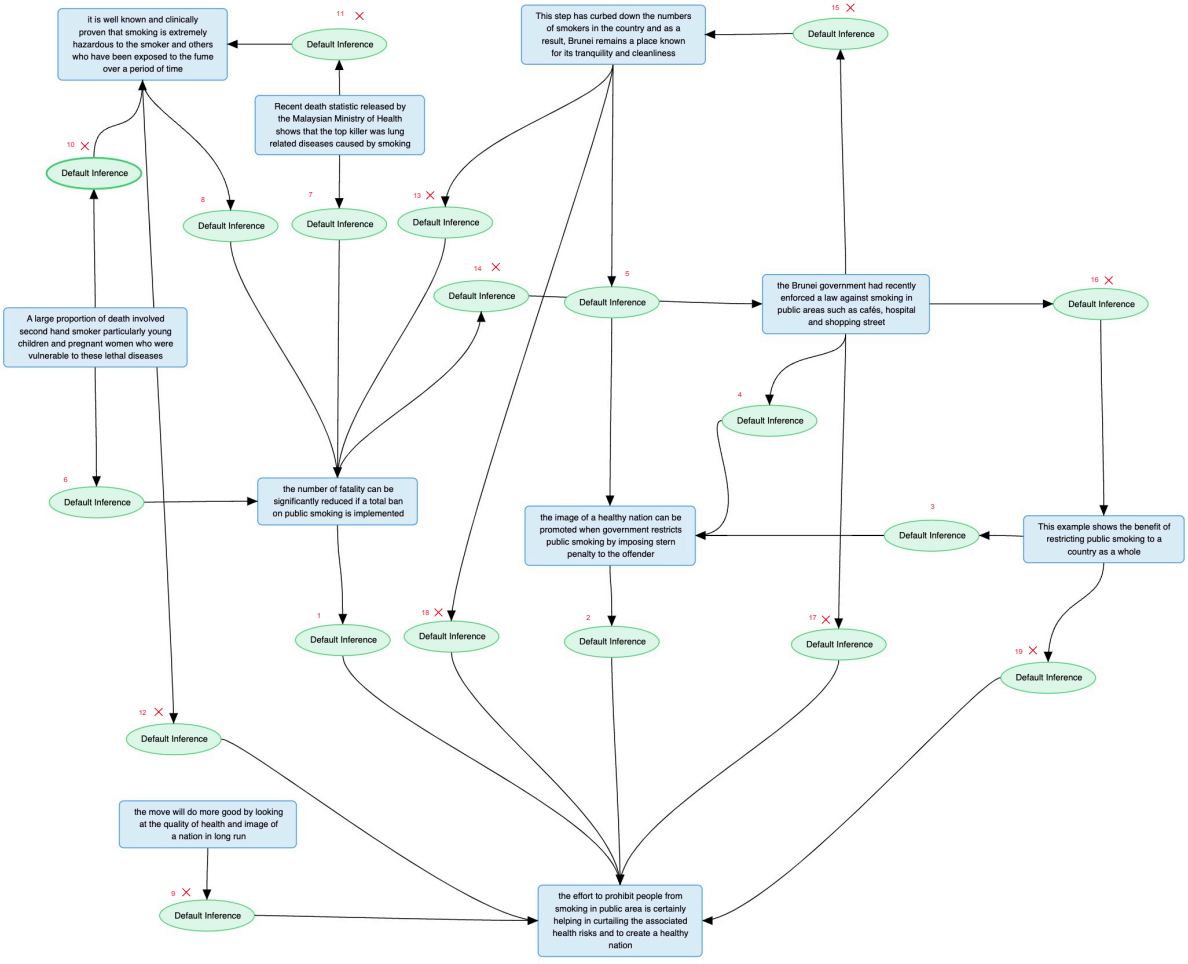


Figure 4: Example of error analysis. The argument map displays relations with arbitrary numbering, where incorrect predictions are marked with an (x) symbol.