Scalable Methods for Long-Range Argumentation in Distributed Discourse

Alina Littek and John Lawrence
University of Dundee
Dundee DD1 4HN, United Kingdom
180016182@dundee.ac.uk; j.lawrence@dundee.ac.uk

Abstract

Argumentative reasoning over long or distributed discourse remains challenging because the number of possible relations between textual units grows quadratically, while true argumentative links remain sparse. As a result, exhaustive relation classification becomes computationally impractical for long documents, multi-document collections, and fragmented discourse. We address this problem with a modular graph-based architecture designed to aggressively reduce the candidate relation space prior to classification. The framework encodes graph-structured discourse representations and applies a two-stage unsupervised filtering pipeline to eliminate unlikely argumentative links before downstream inference. Across experimental settings, the proposed approach removes more than 99% of candidate relations while substantially improving classification performance (up to 6.5× MCC over an unfiltered baseline). These results show that lightweight modular filtering can make large-scale argumentative reasoning possible without sacrificing predictive quality, enabling practical deployment in long-context and distributed-document environments. [1]

1 Introduction

Understanding complex discourse requires the ability to model information in a structured way and draw reliable conclusions from it. Argumentation offers a useful framework for this, breaking down text into components such as claims and premises and linking them through relations like inference, conflict, or rephrase (Van Eemeren et al., 2014). These relational structures reflect how people reason, justify decisions, and build knowledge from evidence. Argument mining seeks to automate this analytical process. It sits at the intersection of computational linguistics and machine learning, with the goal of converting unstructured discourse into structured, interpretable, and auditable representations of reasoning (Lawrence and Reed, 2020). Argument mining focuses on the computational methods used to detect argumentative units in text, determining how they relate to one another, and reconstructing the underlying structure of reasoning. This enables large-scale analysis of discourse that would otherwise be infeasible to carry out manually. At the core of this process is the automatic detection of key argumentative components and their relational links (Hautli-Janisz et al., 2022). While the foundational mechanisms of argument mining are well-established, their application to long and distributed discourse remains largely unexplored. Bridging this gap is essential for navigating the ever-growing volume of data in complex, multi-document domains, such as legal corpora (Habernal et al., 2024), financial reports, scientific articles, and online debates (Lawrence and Reed, 2020).

This gap persists because existing datasets and benchmarks often assume a locality constraint, treating argumentative content as if it occurs only within short windows or adjacent segments. This assumption overlooks a central challenge of modelling real-world argumentative structure. Realistic arguments rarely appear in short, isolated units, but instead they are often contained within extensive and fragmented discourse that may unfold across documents. Capturing these distributed argumentative structures requires reasoning methods that operate over global rather than local contexts. As documents grow, the number of potential relations between argument units expands quadratically, producing a combinatorial search space that challenges scalable inference on large graph structures. This $O(N^2)$ increase represents a fundamental limitation on analysing argument graphs when candidate edges vastly outnumber true edges. Argumentative links are often extremely sparse, especially in large search spaces, yielding highly imbal-

[1] The implementation code for all experiments can be found at: https://github.com/alittek/graph-filter

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2042–2056
November 6-10, 2026 ©2026 Association for Computational Linguistics

anced relational predictions and making it difficult for models to distinguish meaningful inferences from noise. Consequently, approaches based on exhaustive pair wise comparison are not possible in most real-world environments with limited computational resources. The feasibility and efficiency of long-range argumentation therefore depend on reducing the search space without discarding informative connections.

In this paper, we introduce a modular pipeline that treats search space reduction as an integral part of argumentative reasoning in distributed settings. Rather than attempting relation classification over the full quadratic space of candidates, our framework explicitly reasons about which relations are plausible in the first place. It converts the otherwise intractable graph matching problem into a two-stage process, first clustering graph partitions using semantic and structural embeddings to prune irrelevant regions, and then refining candidate links through lightweight node-level similarity checks. We demonstrate the effectiveness of this approach on the argument relation identification task, providing systematic insights into the strengths, limitations, and trade-offs of different filtering strategies. We release a reproducible implementation to support further research, aiming to bridge the gap between current Transformer-based argument mining methods and the complexity of real-world, distributed discourse. Our contributions are as follows: (i) we formalise long-document argument relation identification as a search space reduction problem under extreme sparsity, (ii) we introduce a modular, unsupervised graph-based filtering approach that separates candidate selection from relation classification, and (iii) we demonstrate that filtering can simultaneously improve scalability and downstream predictive performance.

2 Related Work

The task of processing long documents poses well-known challenges arising from the quadratic growth of possible unit pairs and the small number of true relational signals. Simple Transformer models (Vaswani et al., 2017) used to be the dominant approach in language processing, as they offer powerful performance for short-range dependencies. However, due to the quadratic complexity of their self-attention mechanism, they are computationally expensive and poorly suited for structured inference over long argumentative texts. Existing extensions aim to address this bottleneck by approximating attention patterns (Beltagy et al., 2020; Han et al., 2024a), improving positional encoding for long-range dependencies (Press et al., 2022; Chen et al., 2023), or compressing inputs through document-level embeddings (Czinczoll et al., 2024) and hierarchical processing to aggregate information across segmented inputs (Pappagari et al., 2019). These approaches increase the maximum effective context length, but introduce architectural complexity or high memory demands, making them less suitable for the growing search space in long and distributed documents.

Recently, Large Language Models (LLMs) have demonstrated significant improvements across various argument mining tasks, including argument relation identification (Li et al., 2025; Savigny and Yun, 2026; Gorur et al., 2025). However, while these models show promising results on standardised benchmarks, they do not resolve the challenges inherent in long documents and distributed data sources. LLMs remain fundamentally constrained by their available context window, which limits reasoning over extensive datasets. Although recent models claim capacities exceeding one million tokens, Du et al. (2025) demonstrate that predictive performance typically degrades as input length increases and theoretical input limits are rarely reached in practice. Kuratov et al. (2024) demonstrate that models often use only a small fraction of their context window effectively, with performance deteriorating sharply as input length and task complexity grow. This gap between advertised and effective capacity is further highlighted by Paulsen (2026), showing severe accuracy degradation at context lengths far below reported limits, with no evaluated model maintaining reliable performance across its full claimed window. These constraints suggest that expanded context windows alone do not resolve the challenges of long-range argumentation and cannot substitute for explicit search space reduction.

Graph structures allow capturing non-local and relational properties of discourse in an intuitive way and have become central to many tasks requiring structured knowledge representation. Early work, such as Cross-Document Structure Theory (CST) (Maziero et al., 2010) attempted to define intertextual relations, while recent projects like InterText (Kuznetsov et al., 2022; Ruan et al., 2024) model entire documents as interconnected graphs to analyse cross-document links. Graph-based rep-

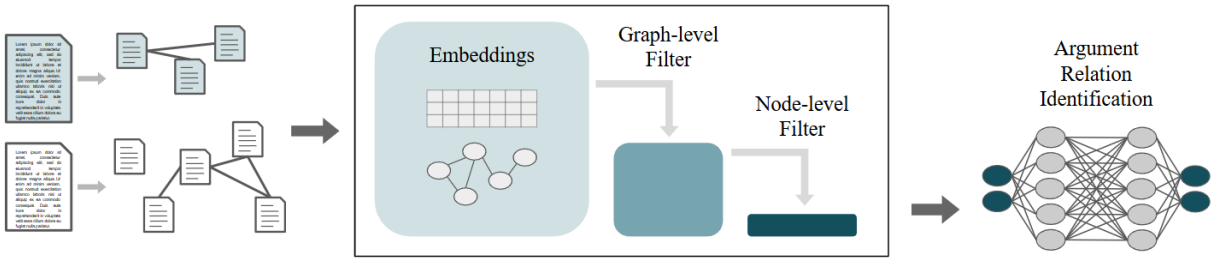


Figure 1: Overview of our framework. Documents are represented as argument graphs and partitioned into subgraphs representing document sections. The graphs are encoded using semantic and structural embeddings to capture the content and the relations of the arguments. The dataset is filtered in two stages, at graph- and at node-level, to enable argumentation in large search spaces. The resulting, reduced dataset is evaluated for argumentative relation detection in long documents.

resentations have been widely applied in multi-document summarisation (Li et al., 2020; Han et al., 2024b) and knowledge-enhanced question answering (Wang et al., 2024) and have proven particularly effective in modelling complex argumentative and semantic structures. They are commonly used to provide a global perspective of argumentative texts and documents (Li et al., 2020). Graph Neural Networks (GNNs) have further strengthened the role of graph-based methods in natural language processing and knowledge representation, offering powerful tools for modelling relational data (Khemani et al., 2024; Wu et al., 2022). GNNs have been shown to improve relation extraction by capturing inter-sentence dependencies (Sahu et al., 2019), and have been applied in argument mining for tasks such as assessing debate polarisation (Alsinet et al., 2022) and identifying argumentative structures (Ding and Ito, 2023). Beyond discourse, global relation graphs support few-shot relation extraction by enabling generalisation across relation types with minimal labelled data (Qu et al., 2020). Yue et al. (2024) employ graph-based techniques and external knowledge sources to manage the complexity of long-document and multi-document settings. Na et al. (2024) and Jain et al. (2024) experiment with reinforcement learning and the integration of domain knowledge and relevance filtering to improve relation extraction, especially for long document settings. These methods commonly focus on large, complex architectures or extensive domain knowledge, approaches that are often computationally prohibitive in real-world applications.

3 Methodology

We address the broader task of relation prediction and identification in long-text settings, where candidate components or pairs may appear far apart within the same document or distributed across multiple texts. Such scenarios substantially expand the search space compared to standard tasks, as the number of possible component pairs grows quadratically with input size, while only a small subset corresponds to meaningful relations. To address the scarcity of annotated data for long-distance relations, we adopt a methodology that operates exclusively on argument graphs. Graph representations capture sparse, non-local dependencies while avoiding reliance on large amounts of labelled text data. By assuming raw text is first converted into graph structures by existing extraction tools, our model performs classification without accessing the original text or metadata, preventing data leakage (e.g., author or source) and ensuring modularity.

Our methodology introduces a general framework for reducing the search space through structured graph representations and multi-stage filtering, preserving meaningful candidates for downstream prediction models. The framework, visualised in Figure 1, consists of three stages:

1. Graph-based representation. Input texts are transformed into graph-structured representations, where nodes denote candidate components (e.g., propositions) and edges capture relations between them. To simulate long-document reasoning, we partition the graphs into subgraphs and introduce controlled cross-partition candidate links to model the challenges of large, distributed data sources.

2. Semantic and structural embeddings. Each subgraph is embedded using both semantic encoders, which capture content-level similarity, and structural encoders, which capture graph topology. The joint embedding space provides a richer representation that supports downstream filtering and

enables the separation of meaningful relational patterns from noise.

3. Two-stage unsupervised filtering. We perform lightweight similarity filtering in two stages. First, a graph-level reduction step prunes unrelated subgraphs by clustering them based on combined semantic and structural evidence. Second, a node-level filtering step applies complementary lexical, semantic, and structural similarity measures to eliminate low-likelihood links. A transformer-based classifier is used for final relation prediction on the reduced set of candidate pairs.

3.1 Task Formalisation

We formulate argument relation identification across distributed discourse as a two-stage search space reduction problem.

Input Graphs and Candidates. We represent discourse as an argument graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $u \in \mathcal{V}$ denote Argumentative Discourse Units (ADUs; e.g., claims or premises) and edges $(u, v) \in \mathcal{E}$ denote argumentative relations. To simulate fragmented or cross-document discourse, $\mathcal{G}$ is partitioned into K disjoint subgraphs $\mathcal{S} = \{G_1, \ldots, G_K\}$. An exhaustive search across subgraphs produces two candidate sets:

$$\mathcal{P}_G = \{(G_i, G_j) \mid 1 \leq i < j \leq K\}, \quad (1)$$

$$\mathcal{P}_V = \{(u, v) \in \mathcal{V}_i \times \mathcal{V}_j \mid (G_i, G_j) \in \mathcal{P}_G\}. \quad (2)$$

While candidate node pairs $|\mathcal{P}_V|$ scale quadratically with $|\mathcal{V}|$, true cross-subgraph relations $\mathcal{E}^* = \mathcal{E} \cap \mathcal{P}_V$ are extremely sparse ($|\mathcal{E}^*| \ll |\mathcal{P}_V|$).

Two-Stage Filtering Pipeline. To prune candidates without discarding true relations, we apply filtering at two granularities:

1. **Graph-Level Filtering (Φ_G):** Selects candidate graph pairs $\widetilde{\mathcal{P}}_G \subseteq \mathcal{P}_G$ likely to contain cross-graph reasoning. The binary target is $y^*_G(G_i, G_j) = 1$ if at least one gold edge connects G_i and G_j, and 0 otherwise.
2. **Node-Level Filtering (Φ_V):** From the candidate node pairs induced by the surviving graphs, denoted $\widetilde{\mathcal{P}}_V$, this step yields a refined pair subset $\widehat{\mathcal{P}}_V \subseteq \widetilde{\mathcal{P}}_V$. The target label is $y^*_V(u, v) = 1$ if (u, v) is a true relation in $\mathcal{E}^*$, and 0 otherwise.

Relation Classification. The downstream classifier f_{ARI} assigns each retained candidate pair $(u, v) \in \widehat{\mathcal{P}}_V$ a label from the relation space:

$$\mathcal{Y}_{\text{ARI}} = \{\text{MA}, \text{RA}, \text{CA}, \text{None}\}, \quad (3)$$

denoting rephrase (MA), inference (RA), conflict (CA), and no relation (None), respectively.

Data	Relations	MA	RA	CA
All	10365	4879	4244	1242
Train	5963	2796	2459	708
Val	2188	1008	913	267
Test	2214	1075	872	267

Table 1: Overview of the size of the dataset and the distribution for the argument relation classes rephrase (MA), inference (RA), and conflict (CA) within the whole dataset and in the subsets for training, validation, and testing.

3.2 Dataset

We use 17 episodes of the QT50 corpus [2], an extension of QT30 (Hautli-Janisz et al., 2022). The dataset is an annotated collection of dialogical argumentation from 30 episodes of BBC's "Question Time". It is one of only a few available datasets featuring large-scale, real-world discussions, annotated with argumentative relations and covering diverse topics and domains, providing a robust dataset for general inferences. Since the classification model we use was trained on a portion of this dataset, we restrict our analysis to the remaining episodes to prevent data leakage and ensure a valid evaluation. An overview of the size of the dataset and number of relations can be found in Table 1. Additional details about the data and the preprocessing can be found in Appendix A. Three types of argument relations are annotated: (i) rephrase (MA) where one proposition restates or reformulates another or links questions and answers, (ii) inference (RA) where one or more propositions provide a reason to accept another proposition, and (iii) conflict (CA) where one proposition provides an incompatible alternative to another proposition. An example of this can be observed in Figure 2.

3.3 Subgraph Generation

To simulate distributed argumentation settings, we partition each episode-level argument graph into multiple subgraphs. This allows us to represent realistic discourse conditions, where related arguments are rarely presented as a single coherent structure. Instead, they are often scattered across different parts of a text or dialogue, requiring reasoning over fragmented segments rather than relying solely on local proximity. Subgraph generation

[2]Dataset available at: https://corpora.aifdb.org/QT50IAAoriginalmaps

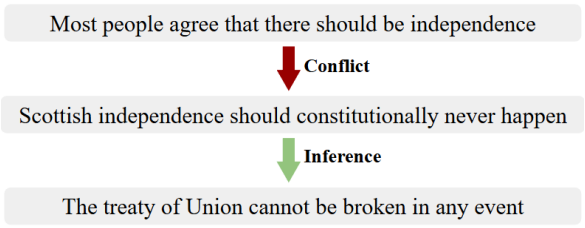


Figure 2: Example of the argumentative units in the data, showing propositions and their relations, corresponding to the nodes and their relations in the argument graphs.

allows us to test models under conditions that more closely approximate real argumentative contexts.

We partition each episode graph using modularity-based community detection to group nodes into cohesive clusters. If the number of detected communities exceeds the desired number of subgraphs, we iteratively merge the smallest clusters. If fewer are detected, we subdivide large ones by removing high-betweenness nodes until they split. We add controlled randomness: each node is assigned a random value in [0,1], and those below a small threshold (the randomisation factor) are reassigned to another subgraph. This increases cross-subgraph connectivity, improves generalisation, and diversifies dataset structure. Each episode graph is split into five subgraphs, producing 85 graphs in total. The final dataset contains three inter-subgraph configurations: (i) unconnected subgraphs from unrelated episodes, (ii) loosely related subgraphs from the same episode, and (iii) directly connected subgraphs with explicit argumentative links at the node level. We label type-(iii) graph pairs as ground truth for the graph-level filtering task and use their node-level links for relation identification.

4 Experiments

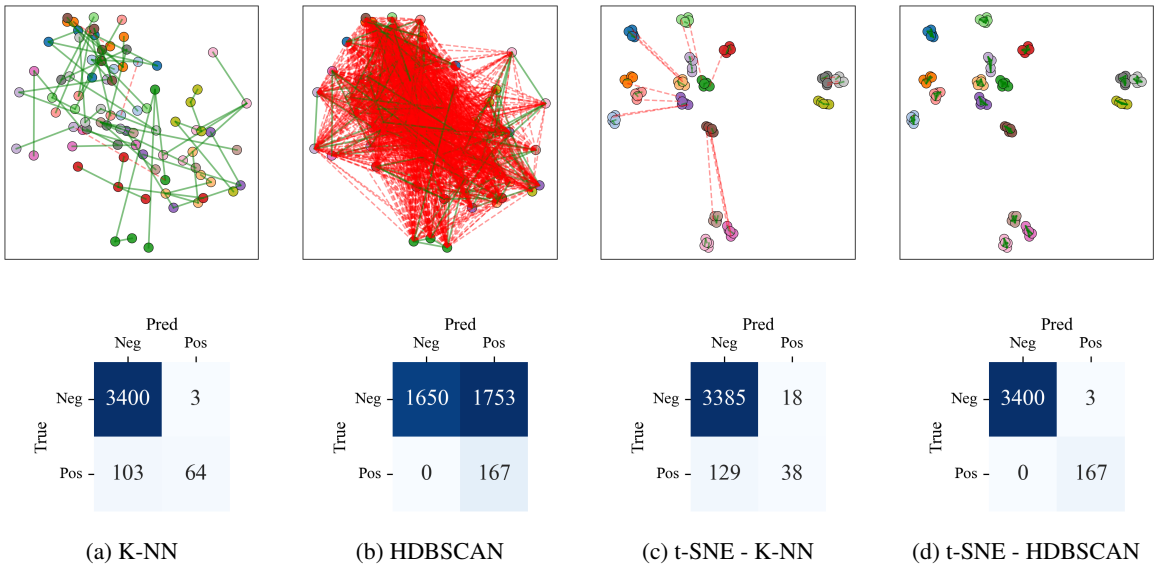
4.1 Graph Embeddings

To enable effective modelling of argumentative discourse, we transform each argument graph into a set of continuous representations that capture both the semantic content of individual argument units and the structure of their relations. The embeddings are created from the cleaned argument graphs only, with no additional document context or metadata included. The semantic content of each node, corresponding to an argument unit such as a claim or premise, is encoded using Sentence-BERT (SBERT) (Reimers and Gurevych, 2019). For each argument node, we obtain a fixed-dimensional embedding that captures the core proposition expressed in the text. This allows the model to group semantically similar arguments, even when they differ in surface form, to recognise paraphrases, shared stances, and thematic similarities across argument graphs. While semantic embeddings capture content, argumentation also depends on the structure of relations between units. To model these, we employ a GNN, which iteratively updates node representations by aggregating information from their neighbours. We explore Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) and Relational Graph Convolutional Networks (RGCNs) (Schlichtkrull et al., 2017). RGCNs extend GCNs by incorporating edge types into the message-passing mechanism. Each relation type has its own transformation matrix, enabling the model to learn how different relations contribute to the meaning of arguments in context.

To combine semantic and structural information, we concatenate the SBERT-derived semantic embeddings with the GNN-based structural embeddings for each node. We evaluate the performance of the different embeddings by testing how well they allow us to find subgraphs belonging to the same QT episode. We find that the combined embeddings achieve perfect performance on this task compared to slightly lower results using semantics or structure only, see Appendix B. By encoding what is said and how it connects to other statements, the model gains a deeper understanding of argumentative content, including both the meaning of an argument and its role within the discourse.

4.2 Graph-level Search Space Reduction

In large-scale graph-based reasoning tasks, a central challenge is efficiently identifying which graphs are likely to contain meaningful relations. Without filtering, the number of graph-to-graph comparisons grows quadratically with dataset size, introducing high computational demands and many irrelevant comparisons. Because each graph pair entails numerous node-level comparisons, reducing the number of graph pairs substantially lowers overall complexity. To address this challenge, we introduce a graph-level filtering mechanism that narrows the search space before fine-grained node-level analysis. All pairs of directly connected subgraphs, as described in Section 3.3, are annotated as ground truth for this task. The goal is to retain the graph pairs that are likely to present argumentative relations at the node-level while discarding



(a) K-NN (b) HDBSCAN (c) t-SNE - K-NN (d) t-SNE - HDBSCAN

Figure 3: Comparison of the graph pair reduction task, showing the embedding space (top row) and confusion matrices (bottom row) for each approach. Each data point represents a subgraph. Colours indicate graphs from the same QT episode. Green edges indicate correctly predicted relations between graphs and red edges false positives.

those that are unrelated. This trade-off must be handled carefully. High recall is critical to ensure that no graph pairs with true node-level connections are removed prematurely, as doing so could lead to the loss of essential relational information in downstream tasks. Our solution presents unsupervised clustering as a means of pruning the search space. The central intuition is that graphs expressing similar argumentative structure or topical focus should occupy the same regions in a learned embedding space. The aim is to restrict relational inference to intra-cluster graph pairs, dramatically reducing the number of comparisons.

We explore two strategies for clustering: (i) direct clustering of the graph embeddings in their original high-dimensional space, and (ii) clustering following a nonlinear dimensionality reduction using t-distributed Stochastic Neighbour Embedding (t-SNE) (van der Maaten and Hinton, 2008). While t-SNE is primarily designed for visualisation, our use exploits its preservation of local neighbourhood structure rather than global geometry. This aligns with our use case, as the goal is high-recall pruning rather than precise cluster boundaries. In both cases, we experiment with a simple nearest neighbour algorithm (K-NN) and the density-based method HDBSCAN (Campello et al., 2013), which is particularly well-suited for discovering clusters of varying shapes and densities without requiring a fixed number. The method does not account for graphs within the same episode without node-level relations, as clustering is based solely on regions in the embedding space. Figure 3 visualises the embedding space of the four different combinations with their corresponding confusion matrices. The graph-level evaluation results can be observed in Table 2. Among the approaches tested, the best results were obtained by applying density-based clustering to the t-SNE-reduced embeddings. t-SNE preserves local neighbourhood structure, making it easier to reveal graph groupings in two or three dimensions, while density-based clustering naturally identifies densely populated regions corresponding to semantically or topically related graphs.

	Clustering	Acc	Prec	Recall	F1
Orig.	K-NN	97.03	95.52	38.32	54.70
	HDBSCAN	50.90	8.70	100.00	16.00
t-SNE	K-NN	95.88	67.86	22.75	34.08
	HDBSCAN	**99.92**	**98.24**	**100.00**	**99.11**

Table 2: Performance comparison for the graph-level search space reduction task with different settings, identifying graph pairs with node-level relations.

4.3 Node-level Filtering

Once relevant graph pairs were identified, the specific argumentative relations between nodes across these graphs needed to be detected and identified. Although graph-level filtering narrows the search space, it still leaves a potentially large number of candidate node pairs, as every node in each graph can be compared to every node in the others. To ad-

Anchor Node u: "the manifesto (...) made a very, very specific commitment to ringfencing a pot of money for social care"		
Retained Pair (u, v)	**Gold:** Inference (RA)	"social care being sorted has been the government's cast iron guaranteed commitment"
Pruned Pair 1 (u, v_1')	**Gold:** No Relation	"we have to deal with the real politic of this"
Pruned Pair 2 (u, v_2')	**Gold:** No Relation	"the administration centres processing currently about 6,000 applications a day and it is done very swiftly"

Table 3: Qualitative example from the test set illustrating node-level filtering for a node u. Filtering retains candidate pair (u, v), preserving the true inference relation (RA). Candidate pairs (u, v_1') and (u, v_2') lack topical and structural relevance and are pruned, preventing non-relational distractors from entering downstream relation classification.

dress this, we introduce node-level filtering strategies, balancing efficiency with the risk of losing relevant connections. As illustrated in Table 3, filtering isolates pairs with shared argumentative focus while pruning non-relational noise prior to classification. To evaluate candidates, we compare three approaches:

1. Lexical Overlap Filtering (TF-IDF). Node texts are compared using TF-IDF vectorisation (up to 100 features, 1-2 grams) with cosine similarity. This high-recall approach includes any pair with positive similarity, prioritising coverage over precision and relying on surface-level lexical overlap rather than deeper meaning.

2. Semantic Filtering (SBERT). Node texts are encoded with a SentenceTransformer to capture semantic similarity. Filtering uses multiple criteria. Pairs must exceed both 75th percentile thresholds and a global threshold (0.2), appear in the top-50 candidates, or achieve high confidence (cosine similarity ≥ 0.3), to allow a pair to pass without meeting all conditions. Unlike TF-IDF, this method emphasises semantic precision, matching texts with similar meaning even when word overlap is low.

3. Structure-Aware Similarity (EMBED). L2-normalised embeddings are compared via dot-product similarity. Filtering uses 75th percentile thresholds and top-20 selection, with a global threshold set at the 90th percentile of all similarities. This ensures that filtering is applied consistently across embeddings with different similarity distributions. Unlike the text-based methods, this approach also captures structural content.

To balance recall and efficiency in node-level filtering, we explored a series of strategies for combining the outputs of the three node-level filtering methods to make use of their complementary strengths. High-precision approaches include the intersection method, which only retains pairs agreed upon by all three methods, ensuring high confidence. A more dynamic high-precision strategy is progressive filtering, which begins with this strict intersection core and then incrementally expands the set through majority voting until a target coverage level is reached. In contrast, recall-oriented strategies such as majority voting and weighted union aim to maximise robustness. We also employ adaptive coverage, which selects the highest-ranked pairs to fit within a predefined target. Further complexity is introduced through methods like rank aggregation, which merges the relative rankings produced by each method. The hybrid strategy starts with a precise intersection set and selectively adds high-confidence pairs identified through the voting strategy. More detail about each combination of filtering strategies can be found in Appendix C. The different approaches provide flexible control over the precision-recall trade-off while mitigating the limitations inherent in relying on any single filtering measure.

4.4 Argument Relation Identification (ARI)

To evaluate the direct impact of candidate reduction, we benchmark relation identification across two progressively filtered subsets: (i) cross-graph node pairs retained after graph-level reduction, and (ii) candidate pairs preserved after subsequent node-level filtering. We compare both against classification over the complete, unfiltered search space to measure efficiency gains and predictive performance. For the downstream classifier, we employ an established Sentence-BERT architecture (Ruiz-Dolz et al., 2021) fine-tuned specifically for argument relation identification. It predicts the three relation types: rephrase (MA), inference (RA), and conflict (CA) alongside a no relation class (None). We intentionally adopt this standardised, lightweight baseline rather than larger language models to isolate the effect of search space filtering. While larger or more specialised archi-

Filter	Combination	Data Retained (%)	Relations Retained (%)			
			All	MA	RA	CA
TF-IDF	-	47.4	71.0	77.9	63.6	70.4
SBERT	-	13.0	79.0	88.5	68.3	80.0
EMBED	-	32.7	73.0	81.2	65.1	68.7
TF-IDF + SBERT	intersection	7.6	60.1	71.4	47.5	60.9
	rank aggregation	52.8	89.8	94.9	84.4	89.6
	weighted union	9.9	74.3	84.9	62.3	75.7
	balanced	13.2	79.0	88.5	68.3	80.0
	adaptive coverage	20.0	81.2	89.8	71.3	82.6
TF-IDF + EMBED	intersection	12.7	55.1	65.6	44.3	52.2
	rank aggregation	60.0	88.9	93.5	84.4	87.0
	weighted union	20.0	69.7	79.1	60.4	66.1
	balanced	15.0	65.5	75.9	55.7	59.1
	adaptive coverage	20.0	68.7	77.9	60.2	62.6
SBERT + EMBED	intersection	6.5	65.2	76.3	54.2	60.9
	rank aggregation	31.8	86.7	93.3	79.2	87.8
	weighted union	20.0	78.2	87.1	69.4	74.8
	balanced	7.9	70.2	81.4	58.2	68.7
	adaptive coverage	20.0	78.2	87.1	69.4	74.8
TF-IDF + SBERT + EMBED	intersection	3.9	51.1	62.8	38.8	48.7
	voting	19.0	78.3	87.7	68.5	76.5
	rank aggregation	62.8	93.6	97.1	89.7	93.9
	progressive	15.0	75.0	85.5	64.0	73.0
	weighted union	19.4	78.3	87.3	69.0	76.5
	balanced	15.7	76.9	86.7	66.4	75.7
	adaptive coverage	20.0	77.5	86.1	68.5	75.7
	hybrid	7.8	69.4	80.6	58.0	66.1

Table 4: Performance comparison for the node-level filtering using lexical overlap (TF-IDF), semantic filtering (SBERT), and structure-aware similarity (EMBED). For pairs of filters, the methods 'voting', 'progressive', and 'hybrid' produce identical outputs as other methods and were excluded.

tectures could be seamlessly integrated, our modular framework is explicitly designed to remain classifier-independent. Our evaluation examines three criteria: candidate-pair reduction rate, end-to-end processing time, and downstream predictive quality. Performance is reported per class in a one-vs-rest configuration using the Matthews Correlation Coefficient (MCC) (Matthews, 1975) to account for severe class imbalance. All classification experiments were conducted on a single NVIDIA L40S GPU with peak memory consumption below 1.4 GB, underscoring the feasibility of our pipeline in resource-constrained settings.

5 Filtering Results & Discussion

The graph-level reduction step substantially decreased the search space, removing noise irrelevant to the task, while fully preserving the relevant argumentative relations. Out of a total of 3,570 possible graph pairs in the dataset, our unsupervised clustering approach selected only 170 pairs, 4.76% of the total. Despite this significant reduction, 100% of the 1,093 gold-standard argumentative node pairs were retained within the predicted graph pair subset, demonstrating perfect recall at the graph level. To highlight the effectiveness of our filtering strategy, we compared it to a baseline in which 170 graph pairs were selected randomly. This baseline retained on average only 3% of the true node-level argumentative pairs (34 out of 1,093). In terms of node-level complexity, this reduction translated into a decrease from over 167 million possible inter-graph node pairs to approximately 7.99 million, a 95.23% reduction of comparisons required for downstream processing.

At the node level, our evaluation highlights the trade-off between candidate node-pair reduction and recall of the ground truth relations across filtering strategies. Table 4 reports our results regarding two metrics: (i) the percentage of node pairs retained after filtering (based on the graph-filtered subset), and (ii) the percentage of ground truth cross-graph relations preserved. This evaluation quantifies the balance between reducing the search space and maintaining essential semantic connections. Among the three individually tested methods, Sentence-BERT similarity achieved the highest coverage of annotated relations, outperforming

Approach	Node Pairs	Reduction	Run Time (hh:mm)	MCC - per class None	MA	RA	CA
Original Data	167,401,606	-	34:34	0.0044	0.0040	0.0029	0.0022
Graph Reduction	7,985,230	95.23%	01:00	0.0174	0.0165	0.0113	0.0089
Graph & Node Reduction	**1,597,046**	**99.10%**	**00:15**	**0.0253**	**0.0272**	**0.0157**	**0.0175**

Table 5: Summary of argument relation classification performance, including dataset size, reduction rate, runtime, and per-class MCC for the three relation types (MA, RA, CA) and the no-relation class (None). Results are shown for the original data, the graph-level reduced subset, and the node-level filtered subset. Classification is based on an SBERT model pretrained on the ARI task. Node-level filtering employs the TF-IDF+SBERT method with 'adaptive coverage' combination.

both TF-IDF and graph-based node embeddings. It retained more positive ground truth pairs while leading to the largest reduction of the candidate space. Combining Sentence-BERT with TF-IDF led to a small improvement in recall but increased the search space. Overall, these results suggest that pretrained semantic embeddings are more effective than surface-level lexical similarity for preserving argumentative links. None of the methods achieved complete coverage and retained all ground truth relations, as pairs with zero semantic similarity are inherently excluded by the similarity-based filtering. The inference (RA) relation proved especially difficult to retain using all methods, suggesting some relations rely on implicit cues not captured by text or embedding similarity alone. Advanced combinations of the filtering methods did not improve recall. However, they offer more flexibility to adapt the filtering process, aiming for a higher percentage of retained ground truth relations or a larger reduction of the search space.

The combination of graph-level clustering and node-level filtering reduced the candidate search space by more than 99%, as can be observed in Table 5. This leads to lower computational costs and shorter processing time, transforming a task involving millions of node comparisons into a manageable workflow that preserves most of the relevant argumentative content. At the class level, MCC scores on the filtered data are five to eight times higher than those obtained without filtering. This substantial increase indicates that reducing the search space improves not only computational efficiency but also the performance of relation identification by limiting noisy candidate pairs. Although the absolute MCC values appear low (0.016 - 0.027), this is expected given the extreme class imbalance inherent in unconstrained discourse relation identification. In our setting, true cross-subgraph argumentative links account for less than 0.07% of the candidate space even after a 99.1% reduction of the original dataset. Rather than an inherent limitation of our framework, these scores illustrate the fundamental difficulty of distributed argument mining. Together, these results demonstrate that integrating graph embeddings with graph- and node-level filtering provides an efficient, scalable, and adaptable foundation for real-world argument mining in distributed discourse.

6 Conclusion

In this work, we addressed relation identification in long-document settings, where the quadratic growth of candidate pairs makes direct processing computationally prohibitive. We proposed a lightweight, modular framework that combines graph-level clustering with node-level filtering, using semantic and structural embeddings to capture both local content and global structure. Our findings highlight three key advantages of the proposed approach: (i) it enables scalable graph-based reasoning in long- and cross-document discourse, (ii) it effectively reduces the search space through unsupervised pre-filtering while retaining relevant candidate links for downstream inference, and (iii) it provides a flexible, modular architecture that separates candidate selection from classification and can be integrated with existing models.

We show that simple, well-designed filtering strategies lead to improved efficiency without loss in performance, in fact improving accuracy. Our two-stage pipeline reduces candidate pairs by over 99% and improves relation classification accuracy by a factor of 6.5 compared to the unfiltered baseline. These results demonstrate that scalable, unsupervised filtering prevents computational bottlenecks and enhances downstream learning by reducing noise. More broadly, our results suggest that lightweight pre-filtering offers a practical path to scalable reasoning over long or distributed texts.

Limitations

While our framework offers an efficient approach to reducing the argumentative search space, some limitations remain that open directions for future work.

First, the dataset used in our experiments, though suitable for an initial exploration of the problem, was not designed to capture long-range argumentative relations. The development of dedicated resources with explicit annotations for such relations would allow for more precise evaluation and stronger empirical grounding. Creating such datasets is costly and labour-intensive, but represents a crucial step for advancing this area.

Second, pretrained sentence embeddings can exhibit representation anisotropy, potentially skewing cosine similarity distributions. While our empirical results show that filtering reliably retains gold relations, our framework does not apply explicit geometric corrections to counter this effect.

Third, we observe variation in how well different relation types are preserved during filtering, with inference relations (RA) proving more difficult to retain. This suggests that future work could benefit from more targeted filtering strategies that adapt thresholds or selection criteria to the specific characteristics of each relation type. Although our methods substantially reduce the number of candidate graph and node pairs, the remaining search space is still large. This challenge highlights the inherent difficulty of a vast search space and the need for further optimisation of downstream models to effectively handle the residual complexity.

Acknowledgments

This work benefited greatly from insightful discussions with Ramon Ruiz-Dolz. This work was supported by Health Data Research UK (HDR-23002), which is funded by the Medical Research Council (UKRI), the National Institute for Health Research, the British Heart Foundation, Cancer Research UK, the Economic and Social Research Council (UKRI), the Engineering and Physical Sciences Research Council (UKRI), Health and Care Research Wales, Chief Scientist Office of the Scottish Government Health and Social Care Directorates, and Health and Social Care Research and Development Division (Public Health Agency, Northern Ireland). This work has been supported in part by: the 'CLARUS' project, funded by the EU Horizon Europe Framework Programme (HORIZON) under grant agreement 101121182; and, the 'AI4Deliberation' project, funded by the European Union under grant agreement 101178806. The authors would like to acknowledge the contribution of all consortium partners to formulate the project. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Teresa Alsinet, Josep Argelich, Ramón Béjar, Daniel Gibert, and Jordi Planes. 2022. Argumentation Reasoning with Graph Isomorphism Networks for Reddit Conversation Analysis. *International Journal of Computational Intelligence Systems*, 15(1):86.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. https://arxiv.org/abs/2004.05150v2.

Katarzyna Budzynska, Mathilde Janier, Juyeon Kang, Chris Reed, Patrick Saint-Dizier, Manfred Stede, and Olena Yaskorska. 2014. Towards Argument Mining from Dialogue. In *Computational Models of Argument*, Frontiers in Artificial Intelligence and Applications, pages 185–196. IOS Press.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending Context Window of Large Language Models via Positional Interpolation. *Preprint*, arXiv:2306.15595.

Carlos Chesñevar, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, and Steven Willmott. 2006. Towards an argument interchange format. *The Knowledge Engineering Review*, 21(4):293–316.

Tamara Czinczoll, Christoph Hönes, Maximilian Schall, and Gerard De Melo. 2024. NextLevelBERT: Masked Language Modeling with Higher-Level Representations for Long Documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4656–4666, Bangkok, Thailand. Association for Computational Linguistics.

Shiyao Ding and Takayuki Ito. 2023. Graph Convolutional Networks for Link Prediction in Argument Structure Extraction. *IIAI Letters on Informatics and Interdisciplinary Research*, 3.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025. Context Length Alone Hurts LLM Performance Despite Perfect Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23281–23298, Suzhou, China. Association for Computational Linguistics.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. Can Large Language Models perform Relation-based Argument Mining? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8518–8534, Abu Dhabi, UAE. Association for Computational Linguistics.

Ivan Habernal, Daniel Faber, Nicola Recchia, Sebastian Bretthauer, Iryna Gurevych, Indra Spiecker Genannt Döhmann, and Christoph Burchard. 2024. Mining legal arguments in court decisions. *Artificial Intelligence and Law*, 32(3):1–38.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David Woodruff, and Amir Zandieh. 2024a. HyperAttention: Long-context Attention in Near-Linear Time. In *The Twelfth International Conference on Learning Representations*.

Mengling Han, Zhongqing Wang, Hongling Wang, Xiaoyi Bao, and Kaixin Niu. 2024b. Abstractive Multi-document Summarization with Cross-Documents Discourse Relations. In *Neural Information Processing*, pages 470–481, Singapore. Springer Nature Singapore.

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 Thirteenth Language Resources and Evaluation Conference*, pages 3291–3300, Marseille, France. European Language Resources Association.

Monika Jain, Raghava Mutharaju, Kuldeep Singh, and Ramakanth Kavuluru. 2024. Knowledge-Driven Cross-Document Relation Extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3787–3797, Bangkok, Thailand. Association for Computational Linguistics.

Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. 2024. A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 11(1).

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *Preprint*, arXiv:1609.02907.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and Resubmit: An Intertextual Model of Text-based Collaboration in Peer Review. *Computational Linguistics*, 48(4):949–986.

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

Hao Li, Viktor Schlegel, Yizheng Sun, Riza Batista-Navarro, and Goran Nenadic. 2025. Large Language Models in Argument Mining: A Survey. *Preprint*, arXiv:2506.16383.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging Graph to Improve Abstractive Multi-Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

B. W. Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Erick Galani Maziero, Maria Lucía Castro Jorge, and Thiago Alexandre Salgueiro Pardo. 2010. Identifying Multidocument Relations. In *Proceedings of the 7th International Workshop on Natural Language Processing and Cognitive Science*, pages 60–69, Funchal, Madeira, Portugal. SciTePress - Science and and Technology Publications.

Byeonghu Na, Suhyeon Jo, Yeongmin Kim, and Il-chul Moon. 2024. Reward-based Input Construction for Cross-document Relation Extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9254–9270, Bangkok, Thailand. Association for Computational Linguistics.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.

Norman Paulsen. 2026. Context Is What You Need: The Maximum Effective Context Window for Real World Limits of LLMs. *Advances in Artificial Intelligence and Machine Learning*, 06(01):01–26.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

Meng Qu, Tianyu Gao, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 7867–7876. JMLR.org.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024. Re3: A Holistic Framework and Dataset for Modeling Collaborative Document Revision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4635–4655, Bangkok, Thailand. Association for Computational Linguistics.

Ramon Ruiz-Dolz, Jose Alemany, Stella M. Heras Barbera, and Ana Garcia-Fornes. 2021. Transformer-Based Models for Automatic Identification of Argument Relations: A Cross-Domain Evaluation. *IEEE Intelligent Systems*, 36(6):62–70.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.

Henri Savigny and Bruno Yun. 2026. AMELIA: A family of multi-task end-to-end language models for argumentation. *Argument & Computation*, 17(1):79–103.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling Relational Data with Graph Convolutional Networks. *Preprint*, arXiv:1703.06103.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Frans H. Van Eemeren, Bart Garssen, Erik C. W. Krabbe, A. Francisca Snoeck Henkemans, Bart Verheij, and Jean H. M. Wagemans. 2014. *Handbook of Argumentation Theory*. Springer Netherlands, Dordrecht.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge Graph Prompting for Multi-Document Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19206–19214.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2022. Graph Neural Networks for Natural Language Processing: A Survey. *Preprint*, arXiv:2106.06090.

Hao Yue, Shaopeng Lai, Chengyi Yang, Liang Zhang, Junfeng Yao, and Jinsong Su. 2024. Towards Better Graph-based Cross-document Relation Extraction via Non-bridge Entity Enhancement and Prediction Debiasing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 680–691, Bangkok, Thailand. Association for Computational Linguistics.

A Data

A.1 QT Dataset

The QT30 dataset is an extensive annotated collection of dialogical argumentation from 30 episodes of BBC's "Question Time". The dataset, comprising 19,842 utterances and approximately 280,000 words, represents one of the largest corpora of analysed broadcast political debate. QT30 is being expanded with 20 additional episodes annotated in the same style, increasing the size to form QT50. The dataset captures the dynamic interactions between public and political figures, featuring questions from the audience on current political issues and responses from a panel of UK political and societal figures. The QT dataset is annotated using Inference Anchoring Theory (Budzynska et al., 2014), a framework in manual argument analysis that describes the structure of arguments and relations between them. In this project, data from individual episodes of the show in the Argument Interchange Format (AIF) (Chesñevar et al., 2006) are used. The AIF is designed for argument construction, representation, visualisation, and exchange of data in a standardised notation. AIF is using a graph-based structure where arguments represented as networks of text sections that are connected through relations such as inference or conflict. Each episode's data is divided into multiple sections, with a separate Inter-Map Communication (IMC) map connecting arguments between sections.

A.2 Data Preprocessing

During preprocessing, all sections of an episode and their corresponding IMC annotations are integrated into a single AIF representation, producing one combined argument graph per episode. Each graph merges the node maps that define the argumentative units and their roles with the IMC map, capturing interrelations across sections, resulting in a unified structure for downstream analysis. As part of graph construction, non-essential node types are removed to streamline the representation. Specifically, locution (L), illocution (YA), and transition (TA) nodes are excluded. In contrast, nodes denoting argumentative relations such as rephrase (MA), inference (RA), and conflict (CA) are converted into labelled, typed edges between relevant text nodes, preserving relationships in a simplified form. The dataset features diverse argument structures, including convergent and linked arguments. For the purposes of this analysis, we abstract away from internal argument composition and focus solely on pairwise relationships between content nodes. This reduction to labelled edges simplifies the graph structure while retaining key relational information. Further preprocessing ensures consistency and integrity across the data. Inconsistencies between node maps and IMC annotations are resolved, and the individual node identifiers are aligned across all maps. Additionally, duplicate nodes and redundant or contradictory edges are identified and removed. The resulting graphs are clean, structurally coherent, and ready for use in argument mining tasks.

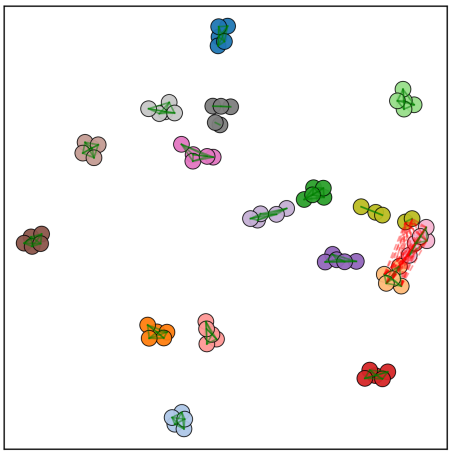
Data	Edges	MA	RA	CA
Dataset	1093	511	467	115
Train	580	281	239	60
Val	235	104	105	26
Test	278	126	123	29

Table 6: Overview of the cross-graph edges in the dataset overall and in the three splits. Showing the total number of edges and the class distribution for the three classes rephrase (MA), inference (RA), and conflict (CA).

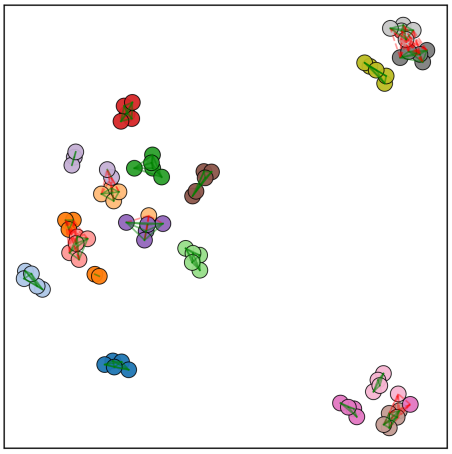
A.3 Data splits

The dataset is split into training, validation, and testing subset before any training to ensure an even distribution of all argument relation classes within the sets. Due to only small numbers of cross-graph relations in the data and an uneven distribution of the three classes MA, RA, CA, the data subset are created with a split close to the class distribution in the overall dataset. To achieve this, a class-balanced splitting strategy is used to assign graphs to each subset. This method prioritises maintaining proportional representation of each class across the splits by iteratively allocating graphs based on their contribution to the class totals, helping to reduce imbalance and supporting reliable evaluation. Table 6 shows an overview of the number of cross-graph relations in the dataset overall and the different data splits.

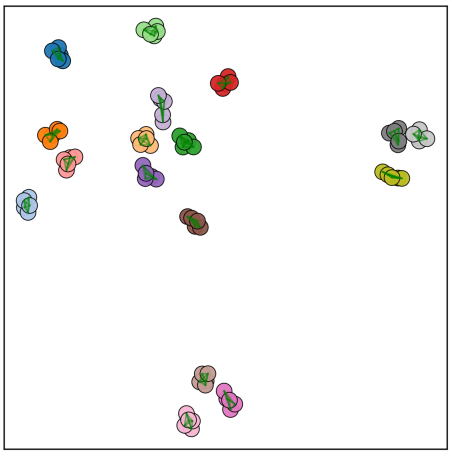
B Structural and Semantic Embeddings



(a) Text-based embeddings



(b) RGCN-based embeddings



(c) Text + RGCN embeddings

Figure 4: Embedding space of the semantic and structural graph embeddings using a) the embeddings of the node text, b) GNN training for structural embeddings, and c) a combination of both. Each data point represents a graph; colours indicate subgraphs from the same QT episode. Green edges indicate correctly predicted relations between graphs and red edges false positives.

Embedding	Acc	Prec	Recall	F1
SBERT	98.40	77.83	92.94	84.72
RGCN	97.54	69.52	85.88	76.84
Combined	**100.00**	**100.00**	**100.00**	**100.00**

Table 7: Performance of semantic, structural, and combined graph embeddings in identifying related subgraphs of the QT episodes.

To evaluate the different embedding types, we test how well they allow us to find subgraphs belonging to the same QT episode. We perform basic clustering of the SBERT and the RGCN embeddings individually and combined, to compare the performance. Table 7 shows that the combination of the structural and semantic embeddings achieves perfect performance on this task. Figure 4 visualises the clustering results of the three different embedding types.

C Filter Combinations

A brief overview of the eight different approaches that were employed and evaluated for effectively combining the node-level filtering methods.

Intersection This strict strategy includes only those node pairs that are identified by all selected filtering methods. It favours high precision and is well-suited when agreement between methods is a strong indicator of relevance. However, it often results in lower recall, as it excludes any pairs missing from even a single method.

Voting In this more lenient strategy, a node pair is retained if it is found by the majority of the methods. This balances between strict filtering and inclusivity, offering improved recall over intersection while still requiring some agreement between methods.

Rank Aggregation Instead of relying on absolute similarity scores, this strategy aggregates rankings of node pairs from each method, as different similarity scales may lead to one method dominating integration. We explicitly discard raw similarity values and instead rely only on the relative orderings (ranks) within each method. This ensures that all methods contribute equally, since what matters is whether a candidate pair is consistently ranked near the top across methods, not the specific score. The average rank normalises across heterogeneous scales by design, making the integration robust to the problem of different similarity scales.

Progressive Filtering This strategy combines the outputs of the selected filtering methods in a stepwise manner, starting with a strict intersection of node pairs that all methods agree on and then gradually relaxing the selection criteria. Initially, only pairs present across all methods are retained, which ensures high precision but often results in low pair coverage. To address this, the method introduces additional pairs based on majority voting, where pairs supported by more than half of the methods are incrementally added back, prioritised by their average similarity scores. This process continues until a predefined target proportion of pairs, relative to the total possible pairs, is reached or a maximum pair ratio threshold is exceeded. The progressive approach is designed to balance precision (from strict intersection) and recall (via controlled voting-based expansion), rather than relying on voting alone.

Confidence-Weighted Union This approach includes all node pairs found by any method but applies a weighted scoring function. It is implemented as a linear combination of three confidence signals: the average similarity across active methods (40% weight), the maximum similarity across methods (30%), and the fraction of methods in agreement (30%). This produces a single confidence score that prioritises both strong individual matches and consistent cross-method support. A configurable confidence threshold controls the strictness of inclusion, offering a soft union mechanism with prioritisation.

Balanced Scoring This method explicitly optimises for a trade-off between reduction (efficiency) and recall. It computes a composite score for each node pair based on similarity quality and inter-method agreement, weighted according to a tunable balance parameter. It is particularly useful for adaptive tuning in task-specific settings.

Adaptive Coverage This strategy aims to retain a sufficient number of high-scoring pairs to meet a predefined coverage target, using similarity and agreement scores. The top-ranked node pairs are selected up to a maximum number, adjusting thresholds as needed to ensure the target is achieved while keeping the candidate pool within resource constraints.

Hybrid Filtering This strategy combines the strengths of multiple other strategies. It begins with the high-precision intersection set, then selectively adds high-confidence node pairs from the voting strategy that were not already included. This two-stage process allows tight control over pair selection while extending coverage where needed.