

Structure-Aware Data Augmentation for Argument Mining

Masayuki Kawarada

CyberAgent, Japan
Institute of Science Tokyo
kawarada_masayuki@cyberagent.co.jp

Tsutomu Hirao

Kanazawa University
hirao@se.kanazawa-u.ac.jp

Manabu Okumura

Institute of Science Tokyo
oku@lr.pi.titech.ac.jp

Abstract

This paper introduces a novel data augmentation approach designed for argument mining. Argument mining is a task that automatically identifies argumentative structures within texts, typically represented as trees or directed acyclic graphs (DAGs). A major challenge facing current argument mining systems is the limited structural diversity of existing annotated datasets, as manual annotation inevitably captures only a small subset of the argumentative patterns encountered in practice. To overcome this challenge, we propose an innovative data augmentation method. Our approach begins by generating pseudo-argumentative structures through the exploration of various combinations of argument labels. These structures then guide a large language model (LLM) to produce new texts with annotations directly incorporated into the text. We evaluated our method on the Argument Annotated Essay Corpus (AAEC) and the AbstRCT dataset. Experimental results showed that the proposed method outperformed existing augmentation baselines on Component-F1 and Relation-F1 in most settings. Further analysis indicated that the variety in the generated pseudo-argumentative structures plays a crucial role in achieving these improvements.

1 Introduction

Argument mining (AM) is the task of automatically extracting argumentative structures from texts (Lawrence and Reed, 2019). Unlike traditional sentiment analysis, which simply identifies whether an opinion is positive or negative, AM reveals the reasons behind an opinion and their logical connections. Complex argumentative structures are more faithfully captured by directed acyclic graphs (DAGs) but are often simplified to trees in practice for modeling and annotation. In these structures, nodes correspond to argumentative components, which are spans of text labeled according to their roles, such as “CLAIM” or “PREMISE.”

Edges represent relationships of “SUPPORT” or “ATTACK” among these components. AM has been extensively studied in various fields, including law (Elaraby and Litman, 2022), education (Stab and Gurevych, 2017), and medical research (Mayer et al., 2020). It has potential applications in areas like automatic essay scoring (Wachsmuth et al., 2016; Nguyen and Litman, 2018) and document summarization (Fabbri et al., 2021).

A fundamental challenge in AM, despite its significant applications, is the limited structural diversity of existing annotated datasets. The forms of argumentation vary widely, ranging from simple linear sequences of claims and premises to more complex hierarchical structures with multiple branches of reasoning, and this diversity is further amplified by variations across domains, genres, and writing styles. Manual annotation inevitably reflects the biases of specific corpora and annotators, leaving many valid argumentative structures unrepresented.

This paper proposes a novel data augmentation method for argument mining that specifically addresses the need for document-level structural diversity. Our approach leverages probabilistic context-free grammars (PCFGs) (Booth and Thompson, 1973) to systematically generate diverse argumentative structures, which then guide large language models (LLMs) to produce coherent argumentative texts that conform to these predetermined structural patterns. Unlike existing methods that focus on superficial text variations or rely on error-prone auto-annotation, our approach generates entirely new documents with explicitly controlled, hierarchical structural diversity. This structure-first generation deterministically aligns text with its labels, directly addressing the scarcity of diverse structural patterns in annotated datasets.

To demonstrate the effectiveness of our approach, we conducted experiments on two established AM datasets: the Argument Annotated Essay Corpus (AAEC) and the AbstRCT dataset. In

most settings, our method outperformed existing augmentation methods on both datasets in terms of Component-F1 and Relation-F1 scores, which measure the performance of identifying argumentative components and their relationships, respectively. Further analysis revealed that the performance improvements correlate with the structural diversity characteristics of the datasets, supporting our hypothesis that explicitly generating diverse argumentative structures addresses a critical gap in existing augmentation methods.

The main contributions of this paper are: (1) a novel PCFG-based data augmentation method for generating diverse document-level argumentative structures, (2) empirical demonstration of its effectiveness on benchmark datasets, and (3) analysis revealing the relationship between dataset diversity and augmentation effectiveness.

2 Related Work

2.1 Argument Mining

Early studies in AM developed domain-specific annotated corpora to establish benchmarks for the field. Notably, [Stab and Gurevych \(2017\)](#) created a persuasive essay dataset (AAEC), which annotates argumentative components (MajorClaim, Claim, Premise) and their relationships (Support, Attack, For, Against), and has become a widely used benchmark. Other significant datasets include Microtext corpus ([Peldszus and Stede, 2015](#)), which contains tree-structured arguments; the CDCP corpus ([Park and Cardie, 2018](#)), which annotates propositions with five types (fact, testimony, value, policy, and reference); and the AbstRCT corpus introduced by [Mayer et al. \(2020\)](#), consisting of biomedical abstracts, annotated with MajorClaim, Claim, and Evidence. These corpora demonstrate the versatility of AM across various domains, including persuasive writing ([Stab and Gurevych, 2017](#)), legal documents ([Poudyal et al., 2020](#)), and scientific papers ([Mayer et al., 2020](#)).

Methodologically, AM has transitioned from rule-based and traditional machine learning approaches toward advanced neural models ([Persing and Ng, 2016](#); [Stab and Gurevych, 2017](#); [Eger et al., 2017](#); [Bao et al., 2022](#); [Ye and Teufel, 2021](#); [Morio et al., 2022](#)). Early work typically employed feature-based classifiers for identifying argumentative components, followed by structured inference methods like Integer Linear Programming to construct coherent argumentative trees. As datasets

expanded and deep learning techniques matured, researchers increasingly adopted neural methods. For instance, [Eger et al. \(2017\)](#) approached AM as dependency parsing or sequence-to-sequence tasks, using neural networks to predict argumentative graphs. More recently, transformer-based models such as BERT ([Devlin et al., 2019](#)) have become the standard, achieving state-of-the-art results in component classification and relation identification through fine-tuning on annotated corpora. Recent studies have demonstrated that pre-trained generative models, such as T5 ([Raffel et al., 2020](#)) and FLAN-T5 ([Chung et al., 2022](#)), can achieve state-of-the-art performance in AM ([Kawarada et al., 2024](#)). This study treats AM as a text-generation task that generates text structured in the Translation between Augmented Natural Languages (TANL) format ([Paolini et al., 2021](#)), from which argumentative structures can be explicitly extracted. Despite these methodological advancements, existing datasets cover only a limited subset of argumentative structures, causing models to overfit to specific patterns and hindering their ability to generalize to the full spectrum of argumentation encountered in practice.

2.2 Data Augmentation

To address data limitations in NLP, data augmentation artificially generates training examples to supplement manually annotated datasets. Existing methods primarily focus on either lexical or syntactic diversity. Lexical approaches, such as Easy Data Augmentation (EDA) ([Wei and Zou, 2019](#)) and back-translation ([Sennrich et al., 2016](#)), are effective for tasks like text classification ([Xie et al., 2020](#)) but introduce only superficial changes without altering the underlying structural arrangements. Conversely, structure-aware methods manipulate syntactic configurations, utilizing PCFGs ([Booth and Thompson, 1973](#)) or LLMs ([Zhang et al., 2025](#)) to generate varied syntactic subtrees. However, because these methods operate locally at the sentence or subtree level, they are insufficient for AM, which requires capturing complex argumentative structures spanning entire documents.

To address this document-level gap in AM, [Bao et al. \(2025\)](#) proposed two complementary approaches: a quality-oriented synthesis that rewrites existing documents to improve lexical quality while preserving the original structure, and a diversity-oriented synthesis that prompts an LLM to generate new argumentative texts with modified component

sequences. To balance quality and diversity, they also explored mixing the outputs of both methods. However, since the LLM freely generates text without explicit structural constraints, there is no guarantee that the resulting documents exhibit valid or systematically controlled structural diversity. In addition, extracting argumentative structures from the generated texts relies on post-hoc auto-annotation, which risks propagating annotation errors into the augmented corpus.

Our approach addresses both issues by explicitly generating a valid argumentative structure before text generation, which ensures structural diversity and deterministic label alignment without relying on post-hoc annotation.

3 Argument Mining as a Text-to-Text Generation Task

In this section, we first formalize the argument mining task and then describe the text generation framework that our method builds upon. We employ the approach of Kawarada et al. (2024), which regards argument mining as a text generation task, exploiting the TANL format, as our base model. We chose this framework for two reasons: (1) it achieves state-of-the-art performance on standard benchmark datasets, and (2) its text generation paradigm offers a simpler implementation than traditional pipeline-based approaches, which require separate models for span identification, component classification, and relation classification.

3.1 Task Formulation

Argument mining comprises three primary sub-tasks: span identification, component classification, and relation classification. Given an input document, represented as a sequence of n tokens $\mathcal{X} = (x_1, x_2, \dots, x_n)$, span identification involves extracting contiguous subsequences of tokens that constitute argumentative units. Extracted spans are represented as pairs of indices $S = \{(\text{start}_i, \text{end}_i)\}_{i=1}^{|S|}$, where $1 \leq \text{start}_i \leq \text{end}_i \leq n$ and each span corresponds to the token sequence $(x_{\text{start}_i}, x_{\text{start}_i+1}, \dots, x_{\text{end}_i})$. Component classification assigns a label to each identified span, categorizing it according to its argumentative role. Given the predefined set C of argumentative component types (e.g., claim, premise), this step produces a set of labeled argumentative components $L = \{(\text{start}_i, \text{end}_i, c_i)\}_{i=1}^{|L|}$, where each tuple associates span boundaries with its assigned compo-

nent type $c_i \in C$. Relation classification identifies how argumentative components interact with each other through directed, labeled relations. For each related pair of components, we determine the relation type from the predefined set R (e.g., support, attack). Relations are represented as 5-tuples that specify: (1) the source component’s span boundaries $(\text{start}^{\text{src}}, \text{end}^{\text{src}})$, (2) the target component’s span boundaries $(\text{start}^{\text{tgt}}, \text{end}^{\text{tgt}})$, and (3) the relation type $r \in R$. The complete set of relations is $E = \{(\text{start}_i^{\text{src}}, \text{end}_i^{\text{src}}, \text{start}_i^{\text{tgt}}, \text{end}_i^{\text{tgt}}, r_i)\}_{i=1}^{|E|}$.

Thus, the full argumentative structure for a document is represented as $\mathcal{A} = \langle L, E \rangle$, where L contains all labeled argumentative components and E contains all relations between them. An argument mining dataset $\mathcal{D}$ consists of documents paired with their corresponding argumentative structures, formally defined as $\mathcal{D} = \{(\mathcal{X}_j, \mathcal{A}_j)\}_{j=1}^{|\mathcal{D}|}$, where $\mathcal{X}_j$ represents the j -th document and $\mathcal{A}_j$ represents its annotated argumentative structure.

3.2 TANL Format

Kawarada et al. (2024) proposed a novel approach to argument mining by applying the TANL framework. Their method regards argument mining as a text-to-text generation task, where encoder-decoder models generate structured text that directly embeds argumentative structures. In the TANL format, argumentative components and their relations are explicitly encoded as annotated spans within the generated text, enabling the extraction of complete argumentative structures from the model’s textual output. The TANL framework represents the argumentative structures using bracketed annotations, as in the following syntax.

For components with relations: $[s^{\text{src}} \mid c \mid r = s^{\text{tgt}}]$, where s^{src} is the source component’s text span, $c \in C$ is its component type, $r \in R$ is the relation type, and s^{tgt} is the target component’s text span. For stand-alone components without outgoing relations: $[s^{\text{src}} \mid c]$, where only the component text and its type are specified.

The following example illustrates the TANL format, where argumentative components and their relations are embedded directly within the text:

Input: For this reason, many marine lives have been endangered , in the extremes part of the reef become uninhabitable for these marine species . Thus , it is apparent that tourism has threatened the nature environments .

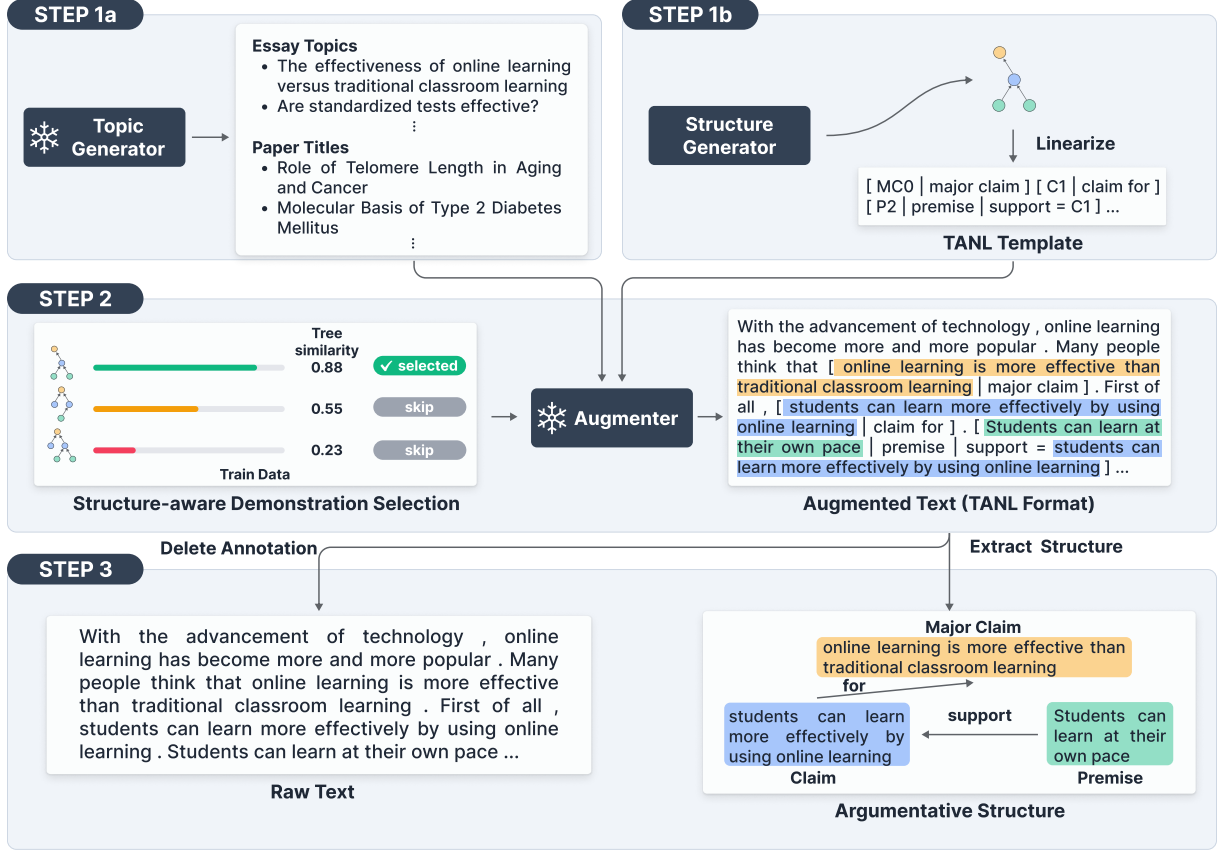


Figure 1: Overview of our proposed structure-aware data augmentation pipeline. We generate novel topics (STEP 1a) and use a PCFG with beam search to generate argumentative trees, which are then linearized into TANL templates (STEP 1b). The Augmenter fills each template with text using demonstrations retrieved based on tree edit distance (STEP 2), and the output is split into an input text and its argumentative structure (STEP 3).

Output: For this reason , [*many marine lives have been endangered, in the extremes part of the reef become uninhabitable for these marine species* | **premise** | **support** = *tourism has threatened the nature environments*] . Thus , it is apparent that [*tourism has threatened the nature environments* | **claim for**] .

The TANL format thus captures both the argumentative components and their logical relationships within a single text representation, which we denote by $\mathcal{X}^{\text{tanl}}$ for a document–structure pair $(\mathcal{X}, \mathcal{A})$.

4 Proposed Method

Our goal is to artificially generate a high-diversity augmented corpus, denoted as $\mathcal{D}^{\text{aug}}$, from the original argument mining corpus $\mathcal{D}^{\text{org}}$.

Figure 1 depicts the overall methodology, which consists of three main components: a Topic Generator, a Structure Generator, and an Augmenter. In this section, we describe each of these components in detail.

4.1 Topic Generator

The Topic Generator employs LLMs with zero-shot prompting to produce diverse topics for essays or titles for academic paper abstracts. Given a prompt p^{topic} , the Topic Generator outputs a set of M diverse topics or titles by utilizing the LLM, $\text{LLM}^{\text{topic}}$. This is formally expressed as:

$$\{\mathcal{S}_i^{\text{aug}}\}_{i=1}^M = \text{LLM}^{\text{topic}}(p^{\text{topic}}), \quad (1)$$

where prompt p^{topic} is specifically formatted, such as “Generate essay topics for discussion:” when creating essay topics or “Generate research paper titles:” when producing academic paper titles.

4.2 Structure Generator

The Structure Generator is designed to produce a variety of argumentative trees, which represent the topology of an argument independent of any specific text. This is done by learning a PCFG from the structures in the training dataset of the original corpus $\mathcal{D}^{\text{org}}$ and then using this grammar to generate new, structurally unique trees.

PCFG Training. First, we extract argumentative structures from the training dataset of the original corpus and abstract them into structural templates by discarding specific text content while preserving the hierarchical relationships between argumentative components. Each argumentative structure $\mathcal{A} = \langle L, E \rangle$ is transformed into an argumentative tree $\mathcal{T}^{\text{org}}$, where nodes represent component types and directed edges represent relation types.

From this collection of structural templates, we estimate a PCFG that captures the distribution of argumentative patterns in the training data. In our PCFG formulation, each production rule represents how a parent component can be expanded into its children, where children are characterized by both their component type and their relationship to the parent.

Formally, a production rule takes the form: $\alpha \rightarrow \beta$, where $\alpha \in C$ is the parent component type and $\beta = \{(r_1, c_1), (r_2, c_2), \dots, (r_k, c_k)\}$ represents a set of children, with each (r_i, c_i) denoting a child of component type $c_i \in C$ connected to the parent via relation $r_i \in R$. For example, the rule $\text{claim} \rightarrow \{(\text{support}, \text{premise}), (\text{attack}, \text{premise})\}$ indicates that a claim can have one supporting and one attacking premise.

The probability of each production rule $P(\alpha \rightarrow \beta)$ is estimated by maximum likelihood, i.e., as the relative frequency of $\alpha \rightarrow \beta$ among all observed expansions of α in the training data. To handle terminal nodes (components with no children), we introduce a special symbol `None` as a possible child for each component type; the probability of the terminal rule $\alpha \rightarrow \text{None}$ is likewise estimated as the fraction of nodes of type α that are leaf nodes in the training data.

Tree Generation with Beam Search. During inference, we use the learned PCFG to enumerate diverse argumentative tree skeletons via beam search. Starting from a `ROOT` symbol, we recursively expand non-terminal nodes: for a parent node with component type α , every right-hand side β with non-zero probability $P(\alpha \rightarrow \beta)$ constitutes a candidate expansion. The expansion of a node terminates when the `None` symbol is applied, whose probability reflects how often that component type appears as a leaf node in the training data; a tree is complete once all of its nodes have terminated.

To prevent the generation of extremely small, large, or repetitive trees, we impose structural constraints on the search: we enforce a minimum num-

ber of nodes (excluding `None` terminals) and a maximum tree depth so that generated structures are neither trivially simple nor excessively complex. At each step, the beam search retains up to a fixed number of the most probable partial trees, with each tree scored by the product of the probabilities of all applied production rules:

$$\log P(\mathcal{T}_j^{\text{aug}}) = \sum_{(\alpha \rightarrow \beta) \in \mathcal{T}_j^{\text{aug}}} \log P(\alpha \rightarrow \beta). \quad (2)$$

Completed trees that satisfy these constraints are moved to a candidate pool, and the search continues until the beam is exhausted or the maximum depth is reached, yielding a large pool of candidate trees (see Appendix B.3).

Finally, to ensure structural diversity, we remove trees with identical topologies from the pool and select the top K unique, highest-probability structures as our final set $\{\mathcal{T}_j^{\text{aug}}\}_{j=1}^K$. These structures serve as the argumentative skeletons for subsequent text generation.

Tree Linearization. The resulting structures are then linearized into a sequence of structural placeholders to facilitate text-based processing by the Augmenter. We traverse the generated argumentative tree to create a flat template where each node is represented by a bracketed placeholder. This placeholder indicates the component type, a unique identifier, and its relation to other components. For example, a generated structure is linearized into a template such as:

```
[ <MC0> | major claim ] [ <C1> | claim
for ] [ <P2> | premise | support = <C1>
]
```

In this format, `<MC0>`, `<C1>`, and `<P2>` act as variable identifiers for the text spans to be generated. The remaining elements explicitly define the argumentative roles and their structural dependencies. This sequence serves as a structural blueprint for the LLM during the text generation phase.

4.3 Augmenter

The final step involves generating augmented argumentative texts that conform to both the topical guidance and structural constraints. The Augmenter takes a topic statement $\mathcal{S}^{\text{aug}}$ from the Topic Generator and an argument tree structure $\mathcal{T}^{\text{aug}}$ from the Structure Generator to synthesize coherent argumentative text in the TANL format. To ensure high-quality text generation without extensive

parameter updates, we employ a large language model, LLM^{aug} , using few-shot prompting.

Few-Shot Prompt Construction. We employ structure-aware demonstration selection (SDS) to construct the few-shot context, using the training set $\mathcal{D}^{\text{org}}$ as the demonstration pool. To provide the most relevant demonstrations for a given generated structure $\mathcal{T}^{\text{aug}}$, we measure the tree similarity between $\mathcal{T}^{\text{aug}}$ and every argumentative tree $\mathcal{T}_m^{\text{org}}$ in the pool, defined as $\text{sim}(\mathcal{T}^{\text{aug}}, \mathcal{T}_m^{\text{org}}) = 1 - \text{TED}(\mathcal{T}^{\text{aug}}, \mathcal{T}_m^{\text{org}})$ for $m = 1, \dots, |\mathcal{D}^{\text{org}}|$, where $\text{TED}(\cdot, \cdot)$ denotes the normalized tree edit distance. We then select the top- k instances with the maximum tree similarity as in-context demonstrations. For each selected instance $(\mathcal{X}_l, \mathcal{A}_l)$, $l = 1, \dots, k$, we extract its argumentative structure $\mathcal{T}_l^{\text{org}}$ and a representative topic $\mathcal{S}_l^{\text{org}}$. If the explicit topic cannot be extracted, we generate it using $\text{LLM}^{\text{topic}}$ with a prompt $\hat{p}^{\text{topic}}$:

$$\mathcal{S}_l^{\text{org}} = \text{LLM}^{\text{topic}}(\mathcal{X}_l, \hat{p}^{\text{topic}}). \quad (3)$$

The structure $\mathcal{T}_l^{\text{org}}$ is represented as a sequence of placeholders, identical to the format used by the Structure Generator. These k triplets of topic, structure, and TANL format text form the few-shot context $\mathcal{C}_{\text{few}} = \{(\mathcal{S}_l^{\text{org}}, \mathcal{T}_l^{\text{org}}, \mathcal{X}_l^{\text{tanl}})\}_{l=1}^k$, guiding the model to generate the target TANL sequence.

Generation. In the generation phase, we systematically combine each topic $\mathcal{S}_i^{\text{aug}}$ with each structural placeholder template $\mathcal{T}_j^{\text{aug}}$. We prompt LLM^{aug} alongside the dynamically retrieved few-shot context $\mathcal{C}_{\text{few}}$ to generate a new argumentative text:

$$\mathcal{X}_{i,j}^{\text{tanl}} = \text{LLM}^{\text{aug}}(\mathcal{S}_i^{\text{aug}}, \mathcal{T}_j^{\text{aug}}, \mathcal{C}_{\text{few}}, p^{\text{aug}}). \quad (4)$$

When $\mathcal{X}_{i,j}^{\text{tanl}}$ fails to parse or violates the intended structure, we re-prompt LLM^{aug} with the faulty output and rule-based feedback describing the detected violation to obtain a revised $\mathcal{X}_{i,j}^{\text{tanl}}$. Finally, splitting each $\mathcal{X}_{i,j}^{\text{tanl}}$ into its raw text $\mathcal{X}_{i,j}^{\text{aug}}$ and argumentative structure $\mathcal{A}_{i,j}^{\text{aug}}$ yields a large-scale augmented corpus $\mathcal{D}^{\text{aug}} = \{(\mathcal{X}_{i,j}^{\text{aug}}, \mathcal{A}_{i,j}^{\text{aug}})\}_{i=1, j=1}^{M, K}$.

5 Experiments

5.1 Datasets

We evaluated our method on two established argument mining datasets: the Argument Annotated Essay Corpus (AAEC) and the AbstrCT dataset.

The AAEC dataset consists of 402 student persuasive essays, split into 286 training, 36 development, and 80 test documents. We used the essay-level annotation setting to represent a realistic document-level argument mining scenario. For AbstrCT, we used the neoplasm subset, which comprises 500 abstracts of randomized controlled trials indexed in MEDLINE, divided into 350 training, 50 development, and 100 test documents. Detailed descriptions of the datasets, including their specific component and relation label schemes as well as our preprocessing steps, are provided in Appendix A.

5.2 Compared Methods

We evaluated our structure-aware data augmentation method by comparing it with current state-of-the-art method (Kawarada et al., 2024) and the following existing augmentation techniques, all integrated with Kawarada et al. (2024)’s framework as the base argument mining approach. We adopted Qwen3-30B-Instruct (Yang et al., 2025)¹ as the backbone model for methods employing LLMs in our comparisons. To demonstrate the generality of our method, we conducted additional experiments using other backbone models. Results are presented in Appendix D.3.

w/o Augmentation (Kawarada et al., 2024). This baseline trains the argument mining model solely on the original corpus, without using any augmented data.

Quality-Oriented Synthesis (QOS) (Bao et al., 2025). This method uses an LLM to generate paraphrastic variants of the training documents. It rewrites the text while strictly preserving the original component inventory and label assignments. We filter the generated outputs to ensure the original argumentative structure remains completely unchanged, thereby increasing lexical diversity without altering the structural topology. As QOS modifies only lexical surface forms while preserving structures, it also serves as a representative of lexical augmentation methods such as paraphrasing and back-translation (Sennrich et al., 2016).

Diversity-Oriented Synthesis (DOS) (Bao et al., 2025). This approach increases topical and structural diversity by prompting an LLM to generate new unannotated argumentative texts with modified structural patterns. For a fair comparison, we

¹We used Qwen/Qwen3-30B-A3B-Instruct-2507-FP8 for our experiments.

	AAEC		AbstRCT	
	Comp.	Rel.	Comp.	Rel.
<i>FLAN-T5 Base</i>				
w/o Augmentation	76.02	53.59	70.35	40.08
QOS	77.10	55.64	71.59	40.22
DOS	76.63	54.80	71.12	40.08
QOS + DOS	77.02	55.94	71.62	40.19
Ours	77.48	56.36	72.00	40.30
<i>FLAN-T5 Large</i>				
w/o Augmentation	77.09	56.91	72.45	44.11
QOS	77.47	57.02	73.62	44.85
DOS	77.84	57.11	74.55	45.24
QOS + DOS	77.33	56.99	74.43	45.88
Ours	78.97	57.19	75.73	45.78
<i>FLAN-T5 XL</i>				
w/o Augmentation	78.20	57.77	73.00	44.32
QOS	79.36	58.72	75.02	46.32
DOS	78.99	57.15	75.74	45.73
QOS + DOS	79.28	58.73	75.26	45.99
Ours	79.92	58.88	75.20	46.11
<i>FLAN-T5 XXL</i>				
w/o Augmentation	80.96	61.73	75.10	47.37
QOS	81.22	62.15	75.70	47.84
DOS	81.10	61.98	75.42	47.63
QOS + DOS	81.71	61.86	75.55	47.82
Ours	82.35	62.49	76.05	48.09

Table 1: Performance comparison on the AAEC and AbstRCT datasets. “Comp.” and “Rel.” stand for Component and Relation-F1 scores, respectively.

supply this method with the same novel topics produced by our Topic Generator. We then assign pseudo-labels to these texts using an Annotator trained exclusively on the original gold labels.

Mixed Synthesis (QOS + DOS) (Bao et al., 2025). This baseline combines the high label fidelity of QOS with the topical and structural novelty of DOS. Following the best-performing configuration in Bao et al. (2025), we randomly sample the instances generated by the two methods to form a blend of 25% QOS and 75% DOS, while keeping the total amount of synthetic data identical to the other augmentation methods, and apply the same two-stage training procedure.

Proposed Method. Our approach combines a Topic Generator, a Structure Generator (PCFG-based), and an Augmenter. This generates documents with both topical and structural diversity, as described in Section 4.

5.3 Experimental Settings

Following Kawarada et al. (2024), we used FLAN-T5 Base, Large, XL and XXL as our argument mining model, fine-tuned with QLoRA (Dettmers

et al., 2023). All augmentation methods followed a two-stage training process: (1) fine-tuning on augmented data (silver data) for 10,000 steps, then (2) fine-tuning again on the original data (gold data) for 10,000 steps, with a batch size of 8. Since the number of training steps is fixed across all settings, the first stage consumes at most 80,000 silver instances regardless of the size of the augmented data. We generated 200,000 augmented samples for AAEC and 100,000 for AbstRCT, with the final model selected based on the highest average Component-F1 and Relation-F1 scores on the development dataset. For our method, the Topic Generator created 100 topics/titles per dataset, while the Structure Generator produced 2,000 unique argumentative structures for AAEC and 1,000 for AbstRCT. We filtered these structures to contain at least 6 nodes with a maximum depth of 6 for AAEC, and at least 1 node with a maximum depth of 4 for AbstRCT. The Augmenter generated each text with one-shot prompting, i.e., with a single demonstration selected by SDS ($k = 1$). When the output failed to parse or violated the intended structure, we re-prompted the Augmenter up to two times and discarded the instance if all attempts failed. All experiments were conducted with three different random seeds and we report their average results. Detailed hyperparameters and training configurations are provided in Appendix B.

5.4 Evaluation Metrics

We evaluated our approach using Component-F1 and Relation-F1 scores, the standard metrics for argument mining. Component-F1 measures performance in identifying and classifying argumentative spans, while Relation-F1 evaluates the identification of relationships between the components. Both metrics require exact matching, providing stringent evaluation of argument mining performance.

6 Results and Discussion

6.1 Main Results

Table 1 shows the results for the AAEC and AbstRCT datasets. Our method improves over the no-augmentation baseline on both datasets across all model sizes. In the FLAN-T5 XXL setting, it achieves the best overall performance among the methods compared in Table 1.

Specifically, on the AAEC dataset using the XXL model, our approach achieved Component-F1 and Relation-F1 scores of 82.35 and 62.49, respectively.

This surpasses the strongest augmentation baseline in each metric, namely QOS + DOS in Component-F1 (81.71) and QOS in Relation-F1 (62.15), by 0.64 and 0.34 points, and yields gains of 1.39 and 0.76 points over the no-augmentation baseline. The same pattern holds across the Base, Large, and XL models, where our method obtains the best score in both metrics, confirming that our augmentation strategy is robust regardless of the base model’s capacity.

On AbstRCT, our XXL model achieved 76.05 Component-F1 and 48.09 Relation-F1, bringing improvements of 0.95 and 0.72 points over the no-augmentation baseline and of 0.35 and 0.25 points over the best baseline, QOS. Here the advantage over the baselines is less uniform. Our method is the best in both metrics at Base and XXL, but at Large it falls 0.10 points below QOS + DOS in Relation-F1, and at XL it falls 0.54 points below DOS in Component-F1 and 0.21 points below QOS in Relation-F1.

The difference in the size of these improvements can be explained by the structural diversity of the two datasets. We measured the diversity of the argumentative structures in each test set, treating two structures as identical only when their tree shapes, component labels, and relation labels all match. On AAEC, all 80 test documents have a distinct argumentative structure, and only 5.00% of the test structures appear in the training data. In contrast, on AbstRCT, 71 out of 100 test documents have distinct structures, and 57.00% of the test structures overlap with the training data. Our structure-aware augmentation is therefore particularly effective on AAEC, where generating varied argumentative patterns addresses a genuine data scarcity problem.

6.2 Ablation Study

To evaluate the contribution of each component in our pipeline, we conducted an ablation study on the AAEC dataset using FLAN-T5 Large (Table 2).

Removing structure-aware demonstration selection (“– w/o SDS”) has only a minor impact on performance, resulting in a slight drop in both metrics. This suggests that retrieving structurally similar examples via tree edit distance provides a small but consistent benefit in guiding the generation. Bypassing the PCFG-based Structure Generator (“– w/o Structure Generator (PCFG)”) and reusing existing trees from the training data also diminished both scores, yielding the lowest Component-F1 among the ablations and confirming our hypoth-

	Comp.	Rel.
Ours (Full Method)	78.97	57.19
– w/o Structure Generator (PCFG)	77.80	56.85
– w/o Topic Generator	77.85	56.12
– w/o SDS	78.45	57.07
Baseline (w/o Augmentation)	77.09	56.91

Table 2: Ablation study on the AAEC dataset using FLAN-T5 Large. We evaluated the impact of removing key components from our augmentation pipeline. “Comp.” and “Rel.” stand for Component and Relation-F1 scores, respectively.

esis that injecting structural diversity is crucial to prevent the model from overfitting to limited structural patterns. Omitting the Topic Generator (“– w/o Topic Generator”) causes the most severe drop in Relation-F1, with a Component-F1 drop comparable to that of removing the Structure Generator, as the lack of dynamically generated topics likely leads to semantic homogeneity and repetitive vocabulary in the augmented texts.

Overall, the results demonstrate that topical diversity, structural diversity, and SDS are complementary, and their integration is essential for maximizing the effectiveness of the augmented data.

6.3 Analysis of Structural Coverage

To evaluate how well our augmentation method captures unseen argumentative structures, we analyzed how many of the test structures on AAEC are covered by each training configuration, under the same matching criterion as above (see Figure 2). While the original training data covers only 5.00% of the argumentative structures in the test set, the baseline Diversity-Oriented Synthesis (DOS only) method reaches 10.00%, and our structure-first generation (Ours only) reaches 21.25%, more than twice the coverage of DOS and over four times that of the training data alone.

Furthermore, we evaluated the combined coverage to represent the union of the structures found in the original training data and the augmented data.

The union of the DOS-generated data and the training data (DOS + train) reaches 12.50% coverage, whereas the union of our generated structures and the training data (Ours + train) yields the highest overall coverage at 23.75%, an absolute gain of 18.75 points over train only. Notably, Ours + train adds only 2.50 points over Ours only, indicating that our generated structures already subsume half of the test structures that the training data covers,

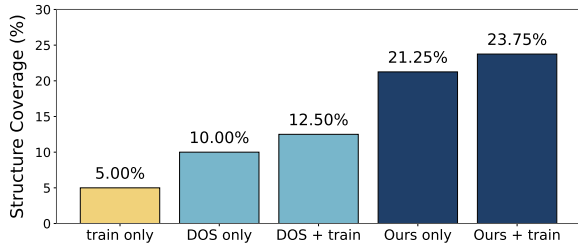


Figure 2: Coverage of argumentative structures in the test dataset by different training configurations on AAEC. A structure is counted as covered only when tree shape, component labels, and relation labels all match. “DOS + train” and “Ours + train” represent the union of the structures found in the original training dataset and the respective augmented datasets. The augmented data from QOS is structurally equivalent to the original training data, thus sharing exactly the same coverage as “train only”.

and that the remaining gain is almost entirely novel. Since QOS preserves the original structures by construction, its augmented data contributes nothing beyond the coverage of train only. The ability to expose the model to nearly five times as many of the test set’s argumentative structures during training is consistent with the robust performance improvements observed in our main results on the AAEC dataset.

6.4 Effect of the Amount of Augmented Data

We varied the amount of augmented data from 10k to 250k samples, scaling the number of generated structures proportionally from 100 to 2.5k, and evaluated on AAEC with FLAN-T5 Large (Figure 3). For each setting, we changed only K , the number of structures retained by the Structure Generator; the beam size and all other generation settings were kept fixed. The candidate pool produced by beam search contained enough unique structures to cover even the largest setting of 2,500 structures. Given the fixed training budget, points beyond 80k differ not in the amount of data seen but in the structural diversity of the pool from which the silver data is drawn. Component-F1 improves monotonically but with clear diminishing returns: 10k samples already yield 78.02, and the gain from 200k to 250k is only 0.02 points. Relation-F1 behaves differently. It falls slightly below the no-augmentation baseline at 10k (56.83 vs. 56.91), does not exceed it until 50k, and peaks at 200k (57.19) before dropping at 250k (57.10). This contrast indicates that identifying components benefits from lexical and topical variety, which even a small amount of aug-

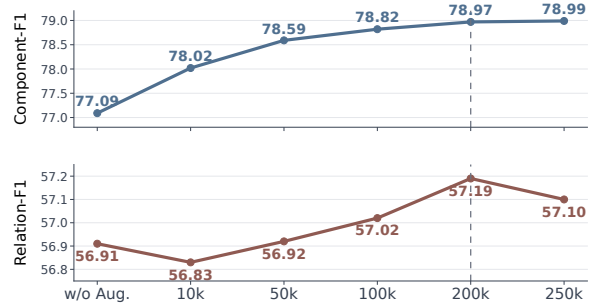


Figure 3: Performance on AAEC with FLAN-T5 Large as the amount of augmented data and the number of generated structures increase together.

mented data provides, whereas relation classification requires exposure to many distinct argumentative structures before the augmented data becomes useful. The decline at 250k is consistent with the long-tail behavior of the PCFG. As the number of required structures grows, the Structure Generator must reach further down the probability-ranked candidate pool and admit increasingly low-probability trees, which deviate from the argumentative patterns observed in human-written texts and act as noise rather than a useful training signal, harming relation classification in particular.

7 Conclusion

In this paper, we introduced a novel, structure-aware data augmentation technique tailored for argument mining tasks, addressing the persistent issue of limited structural diversity in existing annotated datasets. By leveraging a PCFG to create diverse pseudo-argumentative structures, we effectively guided an LLM to generate new, coherent argumentative texts enriched with accurate annotations. Our experimental evaluations using the AAEC and AbstrCT datasets showed that our method outperforms traditional augmentation techniques, particularly in structurally diverse settings. In most settings, the proposed approach delivered notable improvements in both Component-F1 and Relation-F1 scores, underscoring the critical importance of structural diversity in augmentation processes, as opposed to mere lexical variation. Further analyses supported this finding, confirming that the increased variety of argumentative structures significantly contributes to the overall enhancement in argument mining performance.

Limitations

While our method demonstrates improvements over existing augmentation baselines, several limitations should be acknowledged.

Dependence on Training Data Distribution.

Since our PCFG is estimated from the training corpus, the diversity of generated structures is inherently bounded by the patterns already observed. In particular, maximum likelihood estimation assigns low probabilities to rare structural patterns, so they are ranked low in the probability-driven beam search and rarely survive into the top- K selected structures.

Evaluation Scope. Our experiments cover two tree-structured English datasets, leaving DAG-structured corpora such as CDCP (Park and Cardie, 2018) outside the current scope. Extending our PCFG formulation to DAGs is non-trivial, as the tree grammar framework does not accommodate components with multiple outgoing relations. Generalizability to other languages and annotation schemes also remains an open question.

Computational Cost. Generating 200,000 augmented samples for AAEC and 100,000 augmented samples for AbstrCT requires substantial LLM inference, which may limit applicability in resource-constrained settings.

Potential Risks. Our method generates large volumes of synthetic argumentative text with an LLM. Such text may contain factually incorrect or one-sided claims, since the Augmenter is prompted to produce persuasive content rather than verified statements, and it may inherit social biases present in the backbone LLM and in the source corpora. The augmented data is intended solely as training material for argument mining models and should not be treated as a source of factual or medical information; this is particularly relevant for AbstrCT, whose generated texts imitate the style of clinical trial abstracts.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number JP25K03191.

References

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianzhu Bao, Yuqi Huang, Yang Sun, Wenya Wang, Yice Zhang, Bojun Jin, and Ruifeng Xu. 2025. [Exploring quality and diversity in synthetic data generation for argument mining](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26581–26604, Suzhou, China. Association for Computational Linguistics.

Taylor L. Booth and Richard A. Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22(5):442–450.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, and 12 others. 2022. [Scaling instruction-finetuned language models](#). *arXiv preprint*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

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

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

Mohamed Elaraby and Diane Litman. 2022. [ArgLegal-Summ: Improving abstractive summarization of legal documents with argument mining](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. 2021. [ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining](#). In *Proceedings of the 59th Annual Meeting of the Association for*

- Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6866–6880, Online. 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.
- Masayuki Kawarada, Tsutomu Hirao, Wataru Uchida, and Masaaki Nagata. 2024. [Argument mining as a text-to-text generation task](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2002–2014, St. Julian’s, Malta. Association for Computational Linguistics.
- John Lawrence and Chris Reed. 2019. [Argument mining: A survey](#). *Computational Linguistics*, 45(4):765–818.
- Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, pages 2108–2115, Santiago de Compostela, Spain. IOS Press.
- 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.
- Huy Nguyen and Diane Litman. 2018. [Argument mining for improving the automated scoring of persuasive essays](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.
- Joonsuk Park and Claire Cardie. 2018. [A corpus of eRulemaking user comments for measuring evaluability of arguments](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815, London. College Publications.
- Isaac Persing and Vincent Ng. 2016. [End-to-end argumentation mining in student essays](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.
- Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. [ECHR: Legal corpus for argument mining](#). In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75, Online. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Christian Stab and Iryna Gurevych. 2017. [Parsing argumentation structures in persuasive essays](#). *Computational Linguistics*, 43(3):619–659.
- Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2016. [Using argument mining to assess the argumentation quality of essays](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691, Osaka, Japan. The COLING 2016 Organizing Committee.
- Jason Wei and Kai Zou. 2019. [EDA: Easy data augmentation techniques for boosting performance on text classification tasks](#). 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA. Curran Associates Inc.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Yuxiao Ye and Simone Teufel. 2021. [End-to-end argument mining as biaffine dependency parsing](#). In

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 669–678, Online. Association for Computational Linguistics.

Ziyan Zhang, Yang Hou, Chen Gong, and Zhenghua Li. 2025. [Data augmentation for cross-domain parsing via lightweight LLM generation and tree hybridization](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11235–11247, Abu Dhabi, UAE. Association for Computational Linguistics.

A Dataset Details

AAEC The AAEC dataset originally defined three component labels ($C = \{\text{MAJORCLAIM}, \text{CLAIM}, \text{PREMISE}\}$) and four relation labels ($R = \{\text{FOR}, \text{AGAINST}, \text{SUPPORT}, \text{ATTACK}\}$). Since each CLAIM component always depends on a MAJORCLAIM through either a FOR or AGAINST relation, we simplified the label scheme to $C = \{\text{MAJORCLAIM}, \text{CLAIMFOR}, \text{CLAIMAGAINST}, \text{PREMISE}\}$ and $R = \{\text{SUPPORT}, \text{ATTACK}\}$. During evaluation, we merged CLAIMFOR and CLAIMAGAINST into CLAIM, following previous studies (Kawarada et al., 2024).

AbstRCT The AbstRCT dataset covers various diseases including neoplasm, glaucoma, hepatitis, diabetes, and hypertension. The dataset uses three component labels ($C = \{\text{MAJORCLAIM}, \text{CLAIM}, \text{EVIDENCE}\}$) and three relation labels ($R = \{\text{SUPPORT}, \text{ATTACK}, \text{PARTIAL-ATTACK}\}$).

B Implementation Details

B.1 Argument Mining Model and General Setup

We conducted experiments using FLAN-T5 Base, Large, XL, and XXL as our argument mining models, all fine-tuned with QLoRA for computational efficiency. The QLoRA parameters were set as follows: $\text{lora_r} = 16$, $\text{lora_alpha} = 32$, and $\text{lora_dropout} = 0.05$. For our comparison, all data augmentation methods generated 200,000 augmented samples for the AAEC dataset and 100,000 samples for the AbstRCT dataset. The training data was randomly sampled from these augmented datasets. The training process consisted of two steps. First, we performed fine-tuning using the augmented data, followed by a second fine-tuning step using the original data. We employed the adamw_bnb_8bit optimizer and used bfloat16

precision throughout the training processes, maintaining a consistent batch size of 8 for both fine-tuning stages. Detailed hyperparameters can be found in Table 3 and Table 4. To ensure the reliability of our findings, each experiment was conducted three times with different random seeds, and we report the average of the results.

Hyperparameters	Base / Large	XL / XXL
max_steps	10,000	
batch_size	8	
learning_rate	2e-4	1e-4
bf16	true	
lr_scheduler_type	constant	
optim	adamw_bnb_8bit	

Table 3: Hyperparameters for the first training stage (fine-tuning on augmented data).

Hyperparameters	Base / Large	XL / XXL
max_steps	10,000	
batch_size	8	
learning_rate	4e-4	2e-4
bf16	true	
lr_scheduler_type	linear	
optim	adamw_bnb_8bit	

Table 4: Hyperparameters for the second training stage (fine-tuning on original data).

B.2 Topic Generator

The Topic Generator created 100 topics/titles for each dataset. Examples of generated essay topics and paper titles can be found in Table 5.

Essay Topic	Paper Title
The impact of artificial intelligence on employment.	Impact of Vitamin D Supplementation on Bone Density in Postmenopausal Women
How blockchain technology can revolutionize the financial industry.	Efficacy of Telemedicine in Managing Chronic Disease Outcomes
The role of 5G in enhancing global communication.	Genetic Variants Associated with Response to Cancer Immunotherapy
Virtual reality in education and training.	Molecular Mechanisms of Alzheimer’s Disease Progression
The ethics of gene editing and CRISPR technology.	Role of Gut Microbiota in Inflammatory Bowel Disease Pathogenesis

Table 5: Sample topics and titles generated by the Topic Generator.

B.3 Structure Generator

PCFG Training. We train the PCFG from the original training split after converting each annotated document into a rooted argument tree. Each non-root node stores its component label, and each edge stores the relation from the child to its parent. For AAEC, claims are normalized into `claim for` and `claim against` according to their relation to a major claim, while premises keep their support or attack relation. The grammar estimates production probabilities of the form $\alpha \rightarrow \beta$, where α is a parent label and β is the multiset of child label–relation pairs observed under that parent. The same pass also estimates parent-child and sibling ordering probabilities, which are used when the generated tree is linearized.

Beam search settings. At generation time, we start from a ROOT node and iteratively expand the first non-terminal node according to the learned production probabilities. For AAEC, we run beam search with maximum depth 6, maximum number of children 6, and beam size 2,000, collecting completed trees into a candidate pool. After discarding duplicate tree signatures, we select the top- $K = 2,000$ unique structures with the highest probabilities from the pool. For AbstRCT, we use the same procedure with maximum depth 4 and top- $K = 1,000$. We additionally require the generated set of labels for AAEC to include `major_claim`, `claim`, and `premise`. The Structure Generator therefore produces 2,000 unique argumentative structures for AAEC and 1,000 for AbstRCT. For the data-scaling experiment in Figure 3, only the top- K was changed to the required number of structures (100–2,500); all other settings, including the beam size of 2,000, were kept unchanged, as the candidate pool contains sufficiently many unique structures to cover the largest setting.

Structural filtering criteria. To filter out overly simplistic or overly deep structures, we require at least six nodes and a maximum depth of six for AAEC, and at least one node and a maximum depth of four for AbstRCT. We chose these thresholds to remove extreme structures while preserving almost all original training data. Table 6 reports, for each criterion, the proportion of original training documents whose gold structure satisfies it. Every criterion covers at least 99.7% of the training documents, so the filtering preserves the training distribution rather than reshaping it.

Dataset	Criterion	Train coverage
AbstRCT	$n_nodes \geq 1$	349 / 350 (99.7%)
AbstRCT	$max_depth \leq 4$	349 / 350 (99.7%)
AAEC	$n_nodes \geq 6$	286 / 286 (100.0%)
AAEC	$max_depth \leq 6$	286 / 286 (100.0%)

Table 6: Proportion of the original training documents whose gold argumentative structure satisfies each structural filtering criterion.

Linearization into TANL templates. Each generated tree is linearized into the same bracketed structure expected by the TANL parser. The linearizer assigns each component a deterministic placeholder according to its label and position, e.g., `<MC0>` for a major claim, `<C1>` for a claim, and `<P2>` for a premise. Premises additionally contain their relation type and the placeholder of the parent component. A generated tree can therefore be represented as a flat template such as `[ <MC0> | major claim ] [ <C1> | claim for ] [ <P2> | premise | support = <C1> ]`.

B.4 Augmenter

Generation setting. For AAEC augmentation, we generate 200,000 seed pairs by combining generated topics with TANL templates. The Augmenter is prompted in a 1-shot setting with structure-aware demonstration selection based on tree edit distance. Decoding uses `max_new_tokens=1024` and temperature 0.0.

Initial generation prompt. The initial prompt instructs the model to preserve the TANL bracket structure while replacing placeholders with fluent text. The prompt template is as follows.

Task:
Given a Topic and a TANL template with placeholders,
replace the placeholders with natural text
and output
a single-line fulltext TANL essay.

Rules:

- Keep bracket structure, labels, and order exactly as in the template.
- Replace every placeholder with natural text (no brackets inside).
- For each premise relation, the text after `"=` must exactly match the referenced claim text you wrote.
- You may insert non-argument text BETWEEN

bracketed spans to make the essay fluent.
- Do NOT add or remove any bracketed spans.
- Output one line only.

[demonstrations]

Now generate:

Topic:

[topic]

Template:

[structure]

Output:

Automatic revision. After generation, we validate each output against its target TANL template. An output is valid only if it can be parsed, contains no unresolved placeholders, has the same number of bracketed spans as the template, preserves the label sequence, preserves premise relation labels, and makes every premise reference exactly match the generated text of its target claim. Invalid outputs are revised for at most two rounds in our main setting. In each round, only the outputs that remain invalid are sent back to the same Augmenter. A revised output replaces the previous output only when it improves the validation score.

The revision setting keeps the initial generation context in the chat history and accumulates previous revision turns. Thus, the first revision prompt is appended after the initial user prompt and the model’s initial output; from the second revision onward, the prompt additionally includes the earlier diagnostic feedback and the model’s revised outputs. This gives the model increasingly specific evidence about what remains structurally wrong.

Revision prompt. The revision prompt asks the model to regenerate the full TANL essay, rather than patching only the broken span. The prompt template is as follows.

Task:
Fix the previous output so that it satisfies every required fix below.
Use the same Topic, Template, and rules from the first message.
Output exactly one single-line TANL essay.
Do not explain.
Do not output the template.

Previous output:
[previous output]

Required fixes:
[feedback]

Output:

Stage	Valid (%)	Gain (pp)
No revision	49.3	–
After revision 1	74.8	+25.5
After revision 2	90.7	+41.4
After revision 3	91.8	+42.5
After revision 4	92.0	+42.7
After revision 5	92.0	+42.7

Table 7: Effect of automatic revision on texts generated for AAEC with the Qwen3-30B-Instruct Augmenter. Validity denotes exact adherence to the target TANL structure. We adopt two revision rounds in our main setting.

The [feedback] field is generated automatically from the validator. In detailed-feedback mode, it reports the expected and observed span counts, the expected and observed label sequences, unbalanced or nested bracket errors, unresolved placeholders, missing or extra spans, label/order mismatches, and relation-field mismatches such as incorrect relation labels or references to the wrong target claim text.

Effect of the revision loop. To determine how many revision rounds are needed, we ran the loop for up to five rounds on a sample of generations and measured validity after each round (Table 7). Validity rises from 49.3% without revision to 90.7% after the second round, an absolute improvement of 41.4 percentage points. The third through fifth rounds add only 1.3 percentage points in total, indicating that validity largely saturates after the second round and that the remaining outputs contain structural errors that this feedback loop cannot resolve. We therefore cap the revision loop at two rounds in all experiments, which retains virtually all of the attainable validity gain while reducing inference cost.

C Analysis of the Generated Data

C.1 Structural Statistics of the Generated Structures

Figure 2 shows that our generated structures cover more of the gold AAEC test tree shapes than the baselines. Coverage alone, however, leaves open whether the additional structures are plausible, since a generator could also raise coverage by emitting malformed or degenerate trees. We therefore compare the structural distribution of our generated structures for AAEC with that of the gold test set (Table 8).

The generated structures broadly match the gold structures: the node-count and depth ranges largely

overlap (7–24 vs. 8–28; depth 3–5 for both), and the number of root children (1.85 vs. 1.91), the maximum branching factor (4.47 vs. 4.44), and the premise relation ratios (support: 0.966 vs. 0.948) are closely aligned. On average, however, the generated trees are somewhat smaller and shallower (12.65 vs. 15.83 nodes; depth 3.23 vs. 3.95), with a correspondingly smaller maximum width and fewer leaves. The component ratios shift moderately from premises (0.520 vs. 0.639) toward claims; the largest relative deviation appears in the claim against ratio (0.073 vs. 0.039), where the PCFG assigns a non-negligible probability to a comparatively rare gold pattern. The Structure Generator thus broadens structural coverage while remaining within the overall distribution of human-annotated structures, which distinguishes it from random or heuristic recombination of component labels. Section C.2 further examines whether the text realized on top of these structures preserves inter-component relation consistency.

Metric	Gold test	Ours
<i>Tree shape</i>		
Node count (mean)	15.83	12.65
Node count (range)	8–28	7–24
Tree depth (mean)	3.95	3.23
Tree depth (range)	3–5	3–5
Max width (mean)	8.06	6.71
Root children (mean)	1.91	1.85
Max branching (mean)	4.44	4.47
Leaves (mean)	10.26	9.11
<i>Component ratio</i>		
Premise	0.639	0.520
Major claim	0.121	0.147
Claim for	0.201	0.260
Claim against	0.039	0.073
<i>Premise relation ratio</i>		
Support	0.948	0.966
Attack	0.052	0.034

Table 8: Structural distribution of the gold AAEC test structures and of the structures generated by our Structure Generator.

C.2 Relation Consistency of the Generated Text

To assess whether our augmented data maintains comparable quality to the original texts in terms of logical coherence, we conducted an intrinsic evaluation using GPT-4.1 as an external judge, based on Relation-based Argument Mining (RbAM) (Gorur et al., 2025). We extracted component pairs from both the original AAEC test set and pseudo-test data generated by our Augmenter using the same

	support	attack
Original	0.6838	0.2609
Augmented	0.6348	0.2121

Table 9: Comparison of F1 scores for “support” and “attack” relations between the original and augmented datasets.

topics and structures. For each component pair, GPT-4.1 predicted the argumentative relation in a zero-shot setting. Following Gorur et al. (2025), we simplified the four original relation types (For, Against, Support, Attack) into two classes: “support” (For, Support) and “attack” (Against, Attack).

As shown in Table 9, while GPT-4.1’s Relation-F1 is slightly lower on our augmented data than the original data, the differences are modest (both 0.049 absolute, corresponding to 4.9 F1 points for the support and attack relations). These small differences suggest that our augmented texts maintain comparable logical coherence to human-written essays in terms of inter-component relation inference. This evaluation provides evidence that our structure-aware augmentation produces high-quality training data suitable for argument mining tasks.

D Additional Experiments

D.1 Effect of the Topic–Structure Sampling Scheme

Our main setting pairs 100 generated topics with 2,000 generated structures, yielding 200,000 augmented examples for AAEC. The motivation for using multiple topics per structure is that each argumentative structure should receive several semantic realizations, so that the augmented corpus is diverse lexically and topically as well as structurally.

To examine whether a smaller or differently balanced sampling scheme would suffice, we varied the number of topics and structures while keeping the rest of the pipeline fixed (Table 10). Both reduced settings still improve over the no-augmentation baseline, so our method remains effective with half of the augmented data. The main setting is nonetheless the strongest. Against $50 \times 2,000$, which holds the number of structures fixed and halves the topical realizations per structure, it gains 0.73 Component-F1 and 0.19 Relation-F1. Against $100 \times 1,000$, which halves the structural coverage instead, it gains 0.15 Component-F1 and 0.17 Relation-F1. Topical realizations and

Setting	Top.	Str.	Aug.	Comp.	Rel.
w/o Augmentation	–	–	0	77.09	56.91
50 × 2,000	50	2,000	100k	78.24	57.00
100 × 1,000	100	1,000	100k	78.82	57.02
100 × 2,000	100	2,000	200k	78.97	57.19

Table 10: Effect of the number of topics and structures on AAEC with FLAN-T5 Large. “Top.,” “Str.,” and “Aug.” denote the number of topics, the number of generated structures, and the resulting number of augmented examples, respectively.

structural coverage therefore make complementary contributions, and the larger drop under the reduced-topic setting suggests that component identification is the more sensitive of the two to lexical variety.

All settings here sample topics and structures uniformly. Non-uniform schemes, such as allocating more topical realizations to structures that are rare in the training distribution, are a natural extension that we leave to future work.

D.2 Evaluation with a Different Downstream Framework

To verify that the gains from our augmentation are not specific to the FLAN-T5/TANL framework, we conducted an additional experiment on AAEC using the official implementation of Bao et al. (2025)² as the downstream argument mining model. For a fair comparison, we followed their setting and generated synthetic data of twice the size of the gold training data for every method. For our method, we first randomly sampled the required number of topic–structure seed pairs from the 200,000 pairs used in our main experiments, so that the amount of synthetic data was identical across all methods, and then generated the corresponding texts with GPT-4o as the Augmenter backbone.

Table 11 shows the results. Although our method is slightly below QOS in Relation-F1, it achieves the best Span-F1, Component-F1, and average score, suggesting that our structure-aware augmentation is also effective outside the FLAN-T5/TANL framework.

D.3 Comparison of Augmenter Backbones

To verify the generality of our proposed method, we conducted additional experiments using Mistral-Small³ and Qwen3-4B-Instruct⁴ as other back-

Method	Span	Comp.	Rel.	Avg.
QOS	85.70	77.13	57.00	73.28
DOS	86.05	77.51	56.40	73.32
QOS + DOS	85.91	77.20	56.25	73.12
Ours	86.14	77.82	56.66	73.54

Table 11: Performance on AAEC with the downstream framework of Bao et al. (2025). “Span,” “Comp.,” and “Rel.” stand for Span, Component, and Relation-F1 scores, respectively.

bone models. The experimental results are presented in Table 12. These results show that Mistral-Small and Qwen3-4B-Instruct exhibit performance comparable to that of Qwen3-30B-Instruct, demonstrating that our method is not tied to a particular LLM.

Model	Component	Relation
Qwen3-30B-Instruct	77.48	56.36
Qwen3-4B-Instruct	77.18	55.00
Mistral-Small	77.01	55.75

Table 12: Performance comparison of the Proposed Method with Different LLMs on AAEC.

²https://github.com/YuqiHuang2003/QOS_DOS

³mistralai/Mistral-Small-3.1-24B-Instruct-2503

⁴Qwen/Qwen3-4B-Instruct-2507