

ArgSum-X: A Structure-Aware Benchmark for Argument Summarization and Key Point Analysis

Zhiqiang Shi

Department of Informatics
King's College London
zhiqiang.shi@kcl.ac.uk

Oana Cocarascu

Department of Informatics
King's College London
oana.cocarascu@kcl.ac.uk

Abstract

Argument summarization requires capturing semantic content while preserving the underlying argumentative structure. We present ArgSum-X, a dataset supporting both argumentative text summarization (ATS) and key point analysis (KPA) constructed via a multi-stage annotation framework with hybrid LLM–human annotation and quality tiers, and provide a detailed analysis of annotation quality. We propose a comprehensive evaluation framework covering structural, faithfulness, and semantic aspects. Finally, we conduct an empirical analysis of structure-aware modeling, revealing the unique properties and challenges of argument summarization.¹

1 Introduction

Argument summarization seeks to condense complex argumentative content while retaining its underlying argumentative meaning (Bar-Haim et al., 2020; Li et al., 2024; Altemeyer et al., 2025). Unlike generic summarization, it requires not only semantic compression but also the preservation of arguments’ structure and logical flow.

However, argument summarization lacks a unified formulation, with existing work mainly split into two views: key point analysis (KPA) which extracts concise key points from a set of arguments given a topic along with the mapping between individual arguments and the key points they support (Bar-Haim et al., 2020; Altemeyer et al., 2025), and argumentative text summarization (ATS), which generates summaries that preserve the input’s argumentative structure (Li et al., 2024, 2025).

Thus, dataset development for argument summarization is fragmented. Existing ATS resources either assume pre-structured inputs rather than raw argumentative text (Li et al., 2024), use argumentative information for dataset construction without

targeting argument summarization (Fabbri et al., 2021; Syed et al., 2020), or provide summaries that do not preserve argumentative structure (Roush et al., 2024). KPA datasets (ArgKP (Bar-Haim et al., 2020), ArgKP21 (Friedman et al., 2021), ArgCMV (Gurjar et al., 2025)) exhibit limitations such as assuming independent arguments, relying on LLM-only annotations, or limited reproducibility due to restricted data access. Moreover, they do not capture settings where input arguments are closely related to each other and form coherent argumentative texts. Although KPA and ATS differ in definitions, both aim to compress argumentative content while preserving argumentative meaning. Yet no existing dataset supports both formulations.

Furthermore, evaluation remains a fundamental challenge. Current metrics are either for generic summarization (ROUGE or embedding-based measures (Lin, 2004; Zhang et al., 2019)) or tailored for KPA (Li et al., 2023). These approaches are insensitive to errors in argumentative structure, limiting their ability to assess whether generated summaries genuinely preserve argumentative reasoning.

In this paper, we address these limitations by introducing ArgSum-X, a dataset for argument summarization supporting both ATS and KPA within a unified annotation scheme. ArgSum-X is based on the ArgEssay dataset (Bao et al., 2022) and consists of student essays annotated through a staged pipeline that supports both tasks, including discourse unit segmentation, argument role labeling, structured and abstractive summaries, as well as key point generation and alignment. Each stage is annotated using LLMs and then corrected through systematic human intervention, resulting in multiple tiers of annotation quality. Further, we propose an annotation workflow combining LLM annotation, human correction, and adjudication, as well as a suite of evaluation metrics designed to assess the content and structure of argumentative summaries.

Our contributions are: 1) We introduce ArgSum-

¹Code and data can be found at: <https://github.com/zqshizhiqiang/ArgSum-X>.

X, a **unified argument summarization dataset** with multi-layer annotations supporting both ATS and KPA; 2) We propose a **multi-stage annotation framework** that constructs argumentative structure from raw text to abstractive summaries, bridging argument mining and summarization; 3) We develop a **scalable annotation workflow with multiple quality tiers**, combining LLM generation, human correction, and adjudication for studying annotation quality and robustness; 4) We introduce a comprehensive **structure-aware evaluation suite** framework that measures argumentative summary from structural, faithful and semantic aspects beyond traditional summarization metrics; 5) We conduct an **empirical analysis of structure-aware modeling**, showing that general summarization models perform well on semantic similarity and faithfulness metrics but underperform on structure-oriented metrics. We show that structure-aware prompting improves performance, and fine-tuning yields the strongest gains, particularly on structure metrics. Although larger models perform better overall, a gap to ground-truth structure remains.

2 Related Work

Argument Summarization Datasets. The development of ATS and KPA datasets has followed different trajectories. To the best of our knowledge, the only ATS-specific dataset is the multitask end-to-end argument summarization dataset of Li et al. (2024), which uses claim-evidence pairs as inputs and produces abstractive summaries that preserve the input structure; however, they use structured inputs rather than raw argumentative text. Other datasets (ConvoSumm (Fabbri et al., 2021), WebisEditorialSum (Syed et al., 2020)) incorporate argumentative structure during dataset creation but are not designed for argument summarization. Although OpenDebateEvidence (Roush et al., 2024) supports argument mining and summarization, it uses non-argumentative evidence paragraphs as input, and does not aim to preserve argumentative structure, limiting its suitability. For KPA, only three datasets exist: ArgKP (Bar-Haim et al., 2020), introduced with the original KPA task, ArgKP21 (Friedman et al., 2021), which extends ArgKP with three test topics, and ArgCMV (Gurjar et al., 2025) derived from Reddit ChangeMyView where arguments consist of multi-turn, long-context debates. However, they have limitations. ArgKP and ArgKP21 treat arguments independently, which

fails to reflect real-world discourse (Gurjar et al., 2025). ArgCMV is annotated entirely by GPT-4o with only partial human validation, and its usability is constrained by Reddit’s data policy changes as it provides IDs rather than text. As LLMs evolve rapidly, it is unclear whether ArgCMV annotated solely by earlier, weaker models are suitable for evaluating current powerful LLMs.

Argument Summarization Evaluation. Evaluation practices are shaped by the two views of argument summarization. In KPA, evaluation must capture both coverage and redundancy, making standard metrics like ROUGE insufficient (Li et al., 2023). Since lexical overlap poorly reflects semantic equivalence (Li et al., 2023), prior work has adopted semantic soft-scores (soft-precision, soft-recall, soft-F1), computed with metrics such as BLEURT (Sellam et al., 2020) and BARTScore (Yuan et al., 2021). Recent studies have also explored LLM-based judges to assess key point coverage and redundancy (Altemeyer et al., 2025). In contrast, ATS requires evaluation of both semantic content and argumentative structure. Existing work has proposed structure-aware evaluations, such as measuring discourse coverage (Syed et al., 2020) or assessing generated claim–reason structures along multiple dimensions (Tang et al., 2026). However, robust evaluation of ATS, particularly with respect to structural preservation, remains challenging.

3 Data Annotation

ArgSum-X is annotated through a multi-stage, LLM–human hybrid workflow designed to balance scalability with annotation reliability. The workflow deliberately produces multiple tiers of annotation quality, enabling both high-quality evaluation and empirical analysis of learning under heterogeneous supervision noise. This section describes the annotation stages for ATS and KPA, followed by the human correction and adjudication protocol.

3.1 Annotation Stages for Argumentative Text Summarization

Annotations for ATS are organized into four sequential stages, each representing a distinct level of argumentative abstraction. Each stage builds on the outputs of the preceding one, mirroring a bottom-up construction of argumentative structure. **Argumentative Discourse Unit Segmentation.** A raw student essays represents the input. The output is the same text segmented into *argu-*

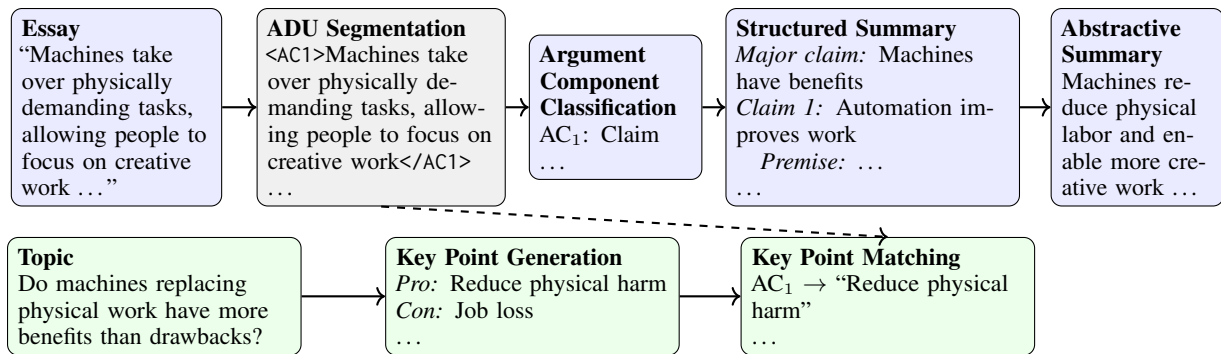


Figure 1: Annotation pipeline in ArgSum-X. Each task proceeds left-to-right. In ATS (top), the original essay text is preserved and augmented with AC tags during ADU segmentation, followed by argument component classification, structured summary construction, and abstractive summary generation. In KPA (bottom), the same ADUs are reused as atomic arguments and matched with key points via key point matching. See Appendix H for an example.

mentative discourse units (ADUs) using explicit markup tags (e.g., argument components (AC): `<AC0> . . . </AC0>`, `<AC1> . . . </AC1>`). Each tagged span corresponds to a minimal argumentative unit expressing a single argumentative function. This representation preserves the original text verbatim while making discourse units explicitly accessible for downstream processing.

Argument Component Classification. Given the ADU-tagged essay, each discourse unit is assigned an argumentative role: major claim, claim, premise, counterargument, or rebuttal. This stage’s output is a structured mapping from ADU identifiers to component labels. These labels provide the semantic foundation required to reason about argumentative organization beyond surface segmentation.

Structured Summary Generation. A structured argumentative summary is generated conditioned on the tagged essay and its AC labels. The structured summary takes the form of an explicit outline that identifies: the major claim of the essay, supporting claims and their associated premises, counterarguments and corresponding rebuttals. Rather than collapsing argumentative content into a flat textual summary, this representation explicitly encodes argumentative roles and hierarchical relations, serving as an interpretable intermediate form between the source text and abstractive summaries.

Abstractive Summary Generation. Finally, the structured summary is transformed into a free-form abstractive summary. The objective at this stage is not only semantic compression, but also the preservation of the argumentative meaning and organization specified by the structured outline. Crucially, this design decouples linguistic abstraction from structure induction: the abstractive summary is

generated from an explicit argumentative structure rather than directly from the raw essay.

3.2 Annotation Stages for Key Point Analysis

To support KPA, ArgSum-X follows the standard two-step formulation of key point generation and key point matching (Friedman et al., 2021; Gurjar et al., 2025), with design choices aligned to ensure compatibility with ATS annotations.

Key Point Generation. For each essay, key points (KPs) are generated given only the writing prompt (topic), without access to the essay text. Annotators correct pro/con KPs generated by the LLM for the topic. The essay is intentionally withheld from the LLM as a single essay cannot fully cover the given topic. Generating KPs from the topic alone may produce KPs not covered in the essay, leading to unmatched arguments (or KPs), which enables analyses beyond strict matching. This setup is similar to ArgKP (Bar-Haim et al., 2020), where KPs are written by experts given only the topic, but different from ArgCMV (Gurjar et al., 2025), where KPs are generated using the topic and the arguments.

Key Point Matching. Each ADU identified during ATS annotation is treated as an individual argument. Each argument is assigned a stance label (pro or con) relative to the topic and matched to zero or more generated key points. This step treats each ADU identified in ATS as an individual argument and maps these discourse-level units to key points, thereby connecting discourse-level argument structure modeling with key point analysis.

3.3 Human Correction and Annotation Quality Tiers

To systematically control annotation quality, we apply different human correction strategies across dataset splits, resulting in explicit supervision tiers. The full dataset contains 1,000 essays, split into 800/100/100 for training, development, and testing.

Training Set. This set includes: 1) *LLM-only annotations (400 essays)* where essays are annotated by LLMs without any human correction; 2) *Single-correction annotations (400 essays)* where each essay is first annotated by LLMs and then corrected by a single human annotator. Correction is applied only to ADU segmentation and argument component classification (ACC).² Corrections are distributed across multiple annotators, each responsible for a small subset of essays, ensuring per-essay consistency while reducing dataset-level annotator bias under a fixed annotation budget.

Development Set. The development split contains 100 essays annotated using the same LLM-based single-correction protocol as the training set. Unlike the partially corrected training data, all annotation stages are reviewed and corrected. Each essay is corrected by one annotator, with annotation effort distributed across multiple annotators.

Test Set. The test split consists of 100 essays annotated using a double-correction and adjudication protocol: 1) An LLM first produces annotations for each stage; 2) Two annotators independently correct the outputs; 3) A third annotator adjudicates disagreements to produce the final gold annotation.

For all splits, the corrected outputs from each stage are used as inputs to subsequent stages, ensuring consistency throughout the annotation pipeline.

3.4 Annotation Design Rationale

This hybrid annotation workflow is driven by two main considerations. First, it enables the construction of a dataset with rich argumentative annotations while remaining cost-effective. Second, the deliberate use of multiple quality tiers allows ArgSum-X to function not only as a benchmark, but also as a testbed for studying robustness to annotation noise. Appendix B presents dataset statistics.

²For structured summary, abstractive summary, and key point generation, we found most of the human corrections are not necessary. See Section 4 and Appendix A for more details.

4 Annotation Quality and Human Correction Analysis

We analyse the quality of LLM-generated annotations and the effect of human correction and adjudication across all stages of our annotation pipeline. Our aim is to understand which aspects of argument annotation are reliably captured by LLMs, and which still require human intervention. We further examine how different human correction strategies interact with the characteristics of each annotation stage. Detailed quantitative results are in Appendix A. Below, we summarise key findings.

Stage-Aware Human Correction. We adopt a stage-dependent human correction strategy tailored to the nature of each annotation task. For low-level tasks such as ADU segmentation and ACC, annotators directly correct LLM outputs. These tasks involve relatively local and well-defined decisions, making direct correction efficient and reliable. In contrast, for higher-level tasks including structured summary construction, abstractive summarization, and key point generation, annotators first evaluate the LLM output and only apply corrections when necessary. This selective correction strategy reflects the inherently higher ambiguity of these tasks, where multiple structurally valid outputs may exist and not all differences correspond to clear errors.

ADU Segmentation. LLM-generated segmentations achieve high coverage of argumentative content, with most gold units at least partially matched. Observed discrepancies primarily arise from boundary differences rather than missing spans, indicating that LLMs are effective at identifying argumentative content but less precise in determining exact span boundaries. Human corrections therefore focus mainly on refining segmentation boundaries.

Argument Component Classification. There is strong agreement between LLM predictions and human annotations for common component types (premises, major claims). Errors are more frequent for structurally complex or context-dependent categories (rebuttals), reflecting ambiguity in their interpretation. Overall, LLM outputs provide a strong initial labeling that requires limited correction.

Structured Summary, Abstractive Summary and Key Point Generation. At higher levels of abstraction, differences between LLM outputs and human annotations become more nuanced. In tasks such as structured summary construction, abstractive summarization, and key point generation, LLM outputs are often already valid and satisfy key con-

straints such as coverage and format. Human corrections in these stages less frequently address missing or incorrect content, many of these changes are not strictly necessary: they reflect alternative but valid ways of organising the same underlying content. This suggests that LLMs can reliably capture both the content and high-level structure of arguments, but that multiple valid representations may exist. This observation motivates our use of selective correction at these stages, allowing annotators to intervene only when outputs deviate substantially from acceptable structure, rather than enforcing a single canonical form.

Human Correction and Adjudication. Across all stages, human corrections primarily refine structural representation rather than introduce new content. We observe high semantic agreement but substantially lower structural agreement between annotators, indicating that disagreement arises from how arguments are organised rather than how they are understood. Rather than reflecting objective errors, these differences often correspond to alternative but valid structural interpretations. Adjudication is therefore necessary to provide a consistent reference for evaluation under this structural variability.

Summary. Overall, our analysis reveals a consistent pattern across annotation stages: LLMs provide strong coverage of argumentative content but show variability in structural representation. Lower-level tasks are relatively well-defined and benefit from direct correction, whereas higher-level tasks admit multiple valid structural organisations, making selective correction more appropriate.

5 Argumentative Summary Evaluation

5.1 Argumentative Structure Evaluation

We evaluate if an abstractive summary preserves the essay’s *argumentative structure*. Rather than relying on surface overlap or factual faithfulness, we focus on whether the organization, progression, and ordering of argument units specified by gold annotations can be recovered from the generated summary text. We evaluate argumentative structure at three complementary levels: i) *argument-unit coverage*, ii) *local argumentative flow consistency*, and iii) *global argumentative structure*, measured w.r.t. both the structured summary and the ground-truth abstractive summary.

5.1.1 Argument Units and Sentence-Unit Mapping

The structured summary defines a set of argument units $U = \{u_1, u_2, \dots, u_n\}$, where each unit corresponds to an annotated argumentative discourse unit with a role (e.g., major claim, claim, premise, counterargument, rebuttal).

We split each summary (generated or gold) into an ordered sequence of sentences $s_1, s_2, \dots, s_m$. For each sentence s_i and argument unit $u \in U$, we use a pretrained natural language inference (NLI) model to test whether s_i entails u . All argument units entailed by s_i are grouped into a set: $S_i = \{u \in U \mid s_i \Rightarrow u\}$. The summary is represented as an ordered sequence of sets of argument units: $\Sigma = (S_1, S_2, \dots, S_m)$ which allows each sentence to express multiple argument units, while the ordered sequence of sentence-level sets preserves the overall argumentative content and flow.

5.1.2 Argument-Unit Coverage

We first evaluate whether key argumentative elements are explicitly present in the summary.

Let u_{maj} denote the major claim. Major claim coverage is defined as:

$$\text{MC-Coverage} = \begin{cases} 1 & \text{if } \exists i \text{ such that } u_{\text{maj}} \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$

This metric evaluates whether the central argumentative stance is recoverable from the summary text. Similarly, coverage for each counterargument unit u_{ca} is defined analogously.

Coverage metrics capture the *presence* of argumentative roles but do not assess ordering or coherence. To facilitate comparison, coverage scores are normalized by dividing the model’s scores by the corresponding ground-truth scores, resulting in a value in 0-100 (%), where 100 indicates performance equal to the reference summary.

5.1.3 Local Argumentative Flow Consistency

We also evaluate whether the summary exhibits coherent *local argumentative flow*, i.e., whether the argument progresses plausibly from one sentence to the next. From the structured summary, we extract an argumentative graph and compute a hop-bounded reachability relation over argument units, which encodes which units can plausibly follow others within a short argumentative span. Given the summary representation $\Sigma =$

$(S_1, S_2, \dots, S_m)$, we evaluate each adjacent sentence transition (S_i, S_{i+1}) . A transition is considered *locally valid* if there exists at least one pair of argument units $u \in S_i$ and $v \in S_{i+1}$ such that: $v = u$ or v is reachable from u according to the structured summary’s reachability constraints.

The *local argumentative flow consistency* score is defined as the proportion of adjacent sentence transitions satisfying this condition. To ensure comparability across summaries, this score is normalized by dividing it by the corresponding score computed on the ground-truth summary, yielding a relative consistency measure. This metric evaluates sentence-level argumentative coherence, rewarding summaries in which consecutive sentences maintain a plausible argumentative relationship (e.g., elaborating a claim, introducing supporting premises, or progressing from a counterargument to a rebuttal). It operates locally and does not enforce global ordering constraints.

5.1.4 Global Structure Score

To evaluate global argumentative organization, we convert both summaries and structured summaries into a shared representation: *unit-level precedence relations*.

Induced Precedence from Summaries. From the ordered sequence $\Sigma = (S_1, \dots, S_m)$, we induce a set of precedence relations: $P_\Sigma = \{(u, v) \mid \exists i < j, u \in S_i, v \in S_j\}$, which represent all argument-unit ordering constraints implied by the summary.

Precedence from the Structured Summary. The structured summary defines a directed acyclic argument graph. We compute a reachability closure over this graph (bounded by a maximum number of hops), yielding: $P_{\text{struct}} = \{(u, v) \mid v \text{ is reachable from } u\}$. This defines a partial order over argument units, encoding permissible argumentative ordering constraints.

Global Structure Consistency w.r.t. the Structured Summary. We evaluate whether the global ordering implied by a summary violates the constraints defined by the structured summary by comparing P_Σ and P_{struct} . We compute precision, recall, and F_1 over precedence relations: precision measures how many ordering relations implied by the summary are allowed by the structured summary, while recall measures how many structured ordering constraints are realized by the summary. These metrics are then normalized relative to the scores obtained on the ground-truth summary.

5.1.5 Global Abstractive Structure Score

We compare the generated summary with the ground-truth abstractive summary. Both summaries are mapped to ordered sequences of argument-unit sets and converted into precedence relations P_{gen} and P_{gt} . Precision, recall, and F_1 over these relations measure if the generated summary follows a human-like global argumentative flow, independently of the structured summary schema.

5.1.6 LLM-based Evaluation via Argumentative Flow Similarity

For LLM-based evaluation, we report a single global structure metric, $\text{FLOWF1}_{\text{gen,gt}}$, which measures the similarity between the argumentative flow of generated summaries and that of ground truth summaries. Specifically, we derive unit-level precedence relations from LLM-identified sentence-argument mappings for both the generated summary and the ground-truth summary. $\text{FLOWF1}_{\text{gen,gt}}$ is then computed as the F_1 score between these two sets of precedence relations, quantifying how closely the model reproduces the argumentative flow expressed by ground truth summarization.

5.1.7 Summary

Overall, our evaluation framework assesses argumentative structure at increasing levels of granularity: argument-unit coverage, local argumentative flow consistency, and global argumentative organization. This layered design enables fine-grained analysis of abstraction, coherence, and structural preservation in argumentative summarization.

5.2 Faithfulness Evaluation

We also assess the *faithfulness* of generated summaries w.r.t. the structured summary annotations. We evaluate faithfulness at the span level. Summaries (both generated and ground truth) are first segmented into spans using an LLM, without assigning argumentative roles.³ These spans correspond to minimal semantic units from the summary and are treated as atomic units for faithfulness evaluation. Each span is then checked against the structured summary using a pretrained NLI model. Given a span s and a structured argument unit u , we apply the NLI model with u as the premise and s as the hypothesis. A span is considered *faithful* if entailed by at least one argument unit from the structured summary; otherwise, it is counted as

³See Appendix F.2 for the span segmentation prompt.

hallucinated. We report hallucination rates as the proportion of hallucinated spans.

We define relative faithfulness by normalizing hallucination rates: $F_{\text{rel}} = \frac{H_{\text{gt}}}{H_{\text{pred}}}$, where H_{pred} and H_{gt} denote the hallucination rates of the generated summary and ground-truth summary, respectively. Value 1 indicates performance equal to the ground truth, while higher values indicate lower hallucination and thus better faithfulness. This faithfulness evaluation captures a dimension orthogonal to argumentative structure preservation. It does not account for the ordering of arguments, their rhetorical roles, or the coherence of argumentative flow. We thus present faithfulness and structure-oriented metrics as complementary perspectives, reflecting different trade-offs between semantic precision and abstract argumentative organization.

5.2.1 LLM-based Faithfulness Evaluation

In addition to the NLI-based faithfulness baseline, we conduct an LLM-based faithfulness evaluation that reflects reader-level judgments under abstraction. This evaluation focuses on whether the content expressed in a summary is supported by the underlying argument, allowing for paraphrase, compression, and implicit reasoning commonly found in abstractive summaries. Faithfulness is evaluated at the span level. Summaries are first segmented into proposition-level spans using an LLM, which serve as atomic units for faithfulness assessment. A second LLM is prompted to identify, for each span, which argument units support it.⁴ A span is *supported* if its content can be stated, paraphrased, summarized, or inferred from one or more argument units; otherwise, it is *unsupported*. The LLM outputs are the supporting argument units for each span. We use these outputs for a binary decision for scoring: a span is considered faithful if at least one supporting argument unit is identified. Dataset-level faithfulness is then reported as the proportion of non-hallucinated (i.e., supported) spans.

We treat NLI-based and LLM-based faithfulness evaluations as complementary. The former measures semantic precision conservatively, while the latter approximates human judgments of content support. Together, they offer a more complete view of faithfulness in argumentative summarization.

5.3 Semantic Similarity Evaluation

We additionally evaluate the semantic similarity between generated summaries and human-written

summaries using standard reference-based content metrics, capturing lexical and embedding-level similarity, rather than argumentative structure or faithfulness to underlying argument units. Specifically, we report ROUGE-1, ROUGE-2, and ROUGE-L scores (Lin, 2004) to measure n-gram overlap between generated summaries and ground-truth summaries. We also compute BERTScore (Zhang et al., 2019), reporting precision, recall, and F_1 , capturing contextual semantic similarity using pretrained language model embeddings.

All semantic metrics are computed by directly comparing the generated summary against the ground-truth summary. These metrics do not assess faithfulness to the structured summary, argumentative roles, or discourse organization. Instead, they provide a complementary view of how closely the generated summaries match human references in surface form and semantic content.

We emphasize that semantic similarity metrics capture a different aspect of summary quality from both faithfulness and argumentative structure. High semantic similarity does not imply correct or complete preservation of argumentative structure, nor does low semantic similarity necessarily indicate poor argumentative reasoning. We therefore report semantic similarity alongside structure-oriented and faithfulness evaluations to provide a comprehensive assessment of summarization behavior.

6 Experiments

6.1 Experimental Setup

We evaluate argument summarization on the ArgSum-X dataset. For the annotation and evaluation metrics, we use Qwen3 235B A22B Instruct 2507.⁵ We use BART (Lewis et al., 2019) as non-LLM baseline. BART-large is fine-tuned on our dataset and evaluated on the test set, BART-CNN is fine-tuned on CNN dataset and evaluated on our dataset. For LLMs, we consider both fine-tuned and prompting-based settings. For fine-tuning, we train LLaMA3.2 1B and Qwen2.5 1.5B using LoRA (Hu et al., 2022) and Qwen2.5 7B using QLoRA (Dettmers et al., 2023). For prompting, we evaluate LLaMA3.2 1B, Qwen2.5 1.5B, Qwen2.5 7B, Qwen3 235B and DeepSeek V3.2. For both fine-tuning and prompting, we use structured and unstructured prompts.⁶ Qwen3 235B and

⁵We also tried GPT4.1 in our preliminary experiments and found Qwen3 235B consistently outperforming GPT4.1.

⁶See Appendix F.3 for the prompts.

⁴See Appendix F.2 for the prompts

Model	MC Cov	CA Cov	Local Cons.	Global Struc F1	Global Abst F1	Flow F1	faith score	LLM faith	ROUGE L	BERTScore F1
BART-large	74.73	97.44	95.47	64.28	0.2485	0.4677	72.02	0.908	0.233	0.346
BART-CNN	45.05	81.37	42.49	30.75	0.0627	0.2339	80.77	0.920	0.184	0.224
<i>Fine-tuning Using Summary Prompt</i>										
LLaMA3.2 1B	69.23	90.34	89.67	48.07	0.2241	0.5170	66.90	0.933	0.251	0.397
Qwen2.5 1.5B	78.02	88.13	83.97	58.50	0.2487	0.5342	63.31	0.959	0.224	0.357
Qwen2.5 7B	84.62	90.34	100	63.13	0.3126	0.5973	73.30	0.970	0.271	0.411
<i>Fine-tuning Using Structure Summary Prompt</i>										
LLaMA3.2 1B	73.63	77.53	90.45	48.59	0.2233	0.5116	64.54	0.943	0.250	0.394
Qwen2.5 1.5B	79.12	79.39	90.08	58.36	0.2481	0.5413	62.74	0.957	0.227	0.366
Qwen2.5 7B	81.32	88.82	96.24	59.24	0.2827	0.5703	69.13	0.966	0.264	0.405
<i>Prompting Using Summary Prompt</i>										
LLaMA3.2 1B	59.34	84.75	53.99	30.89	0.1522	0.4489	51.57	0.890	0.212	0.307
Qwen2.5 1.5B	58.24	74.27	31.00	26.47	0.1077	0.3862	55.14	0.918	0.187	0.298
Qwen2.5 7B	69.23	82.00	58.26	35.20	0.1696	0.5053	65.47	0.956	0.238	0.367
Qwen3 235B	81.32	91.39	68.13	48.21	0.2556	0.5741	62.23	0.968	0.273	0.415
DeepSeek V3.2	73.63	79.86	64.12	33.55	0.1609	0.5139	63.07	0.971	0.243	0.370
<i>Prompting Using Structure Summary Prompt</i>										
LLaMA3.2 1B	62.64	83.82	58.42	34.60	0.1523	0.4740	62.05	0.920	0.227	0.316
Qwen2.5 1.5B	70.33	76.49	75.20	32.88	0.1175	0.4338	58.69	0.908	0.189	0.276
Qwen2.5 7B	79.12	89.76	67.73	53.43	0.2062	0.5496	56.14	0.956	0.253	0.356
Qwen3 235B	82.42	86.61	89.49	68.48	0.2805	0.5999	59.35	0.956	0.268	0.384
DeepSeek V3.2	80.22	85.23	77.10	53.42	0.2361	0.5643	65.00	0.967	0.253	0.365

Table 1: Experiment results for baseline models and LLMs with fine-tuning and prompting.

DeepSeek V3.2 are accessed via public APIs, all other models are run locally with a single 40GB A100 GPU.⁷

6.2 Evaluation Metrics

We evaluate models using three complementary dimensions: argument structure, faithfulness, and semantic similarity, as stated in Section 5.

Structure Evaluation. We use the structure-aware evaluation framework (Section 5.1): argument-unit coverage (**MC-Cov**, **CA-Cov**), local argumentative flow consistency (**Local Cons.**), global structure consistency w.r.t. the structure summary (**Global Struc F1**), global abstractive structure score (directly comparing the generated summary and the ground truth abstractive summary) (**Global Abst F1**), LLM based evaluation (**Flow F1**). MC Cov, CA Cov, Local Cons. and Global Struc F1 scores are normalized by the corresponding ground-truth scores to enable comparison across examples.

Faithfulness. We evaluate faithfulness using hallucination rate at the span level as stated in Section 5.2: relative faithfulness F_{rel} (**faith score**) and LLM based evaluation (LLM faith).

Semantic Similarity. ROUGE-L and BERTScore F1 measure surface-level and semantic similarity between generated and reference summaries.

6.3 Results and Discussion

Structure fine-tuning vs general summary fine-tuning. BART-CNN achieves equally good re-

sults for faithfulness and semantic evaluation, but the structure scores are significantly lower compared to BART-large, suggesting that general summary models fail to preserve the argumentative structure for argument summarization.

Summary prompt vs structure summary prompt. For prompting, guiding models to reason about argumentative structure (structure summary prompt) substantially improves performance across all metrics compared to plain summary prompt (summary prompt). This suggests that LLMs can reason about argument structure when appropriately prompted. However, this advantage disappears after fine-tuning: trained models show minimal differences between prompt variants, suggesting that fine-tuning already encodes structural constraints into the model, making explicit prompt guidance less important. More broadly, structure-aware behavior can be induced either through prompting (via prompt engineering) or through training (via structure-aware data).

Fine-tuning vs prompting. Across all models, fine-tuning yields substantial improvements over prompting-based approaches, particularly on structure-related metrics. This suggests that structure-aware summarization is not purely an inference-time phenomenon, but requires task-specific adaptation. The performance gap further shows that prompting alone is insufficient to enforce global structural constraints in a consistent way, especially for smaller models. In contrast, fine-tuning enables models to learn both local and

⁷See Appendix C for detailed experiment settings.

global argumentative patterns, resulting in more coherent and structurally consistent summaries.

Effect of Model Scale. Larger models perform better across all metrics, particularly in structural metrics. However, even the largest models fail to reach ground-truth-level performance for structure metrics, suggesting that scaling alone does not fully address structural reasoning challenges.

7 Conclusions

We introduced ArgSum-X, a dataset for argument summarization supporting both ATS and KPA. We proposed a multi-stage, hybrid LLM-human annotation framework with quality tiers. Moreover we introduced a comprehensive ATS evaluation framework covering structural, faithful and semantic aspects, and analyzed structure-aware modeling empirically.

8 Limitations

Our work has several limitations. First, although ArgSum-X provides detailed annotations of argumentative structure, the dataset is limited in size and domain diversity, which may restrict the generalizability of our findings to broader argument domains.

Second, although we design a multi-stage annotation pipeline with human verification and adjudication, argument structure remains inherently ambiguous. Many instances allow multiple valid structural interpretations, and the final annotations represent one consistent choice rather than a uniquely correct structure.

Third, several stages of our pipeline rely on Qwen3 235B, including initial annotation, some LLM-based evaluation procedures, and its inclusion as an experimental baseline. This may introduce evaluator-specific biases that could affect assessment of model outputs. Although prior work (Min et al., 2025) suggests that LLM self-evaluation effects are more complex than simple self-favoritism and may not consistently favor outputs from the same model family, we cannot fully exclude potential bias.

Fourth, ArgSum-X is constructed from student argumentative essays of approximately medium length and therefore primarily represents mid-length argumentative discourse. While this enables controlled annotation and evaluation of argumentative structure, it does not capture challenges associated with substantially longer contexts, such

as multi-document debates, legal arguments, scientific articles, policy discussions, or extended online conversations. Future work should investigate structure-aware argument summarization and evaluation in longer and more complex argumentative settings.

Finally, our analysis focuses on prompt-based and fine-tuned models under specific experimental settings. We do not explore alternative training objectives or architectures that explicitly model structure, and leave this as an avenue for future work.

References

- Moritz Altemeyer, Steffen Eger, Johannes Daxenberger, Yanran Chen, Tim Altendorf, Philipp Cimiano, and Benjamin Schiller. 2025. Argument summarization and its evaluation in the era of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35478–35499.
- Jianzhu Bao, Yasheng Wang, Yitong Li, Fei Mi, and Ruifeng Xu. 2022. *AEG: Argumentative essay generation via a dual-decoder model with content planning*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5134–5148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. *From arguments to key points: Towards automatic argument summarization*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- 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.
- Roni Friedman, Lena Dankin, Yufang Hou, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. *Overview of the 2021 key point analysis shared task*. In *Proceedings of the 8th Workshop on Argument Mining*, pages 154–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Omkar Gurjar, Agam Goyal, and Eshwar Chandrasekharan. 2025. [ArgCMV: An argument summarization benchmark for the LLM-era](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21870–21883, Suzhou, China. Association for Computational Linguistics.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. [Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). *Preprint*, arXiv:1910.13461.
- Hao Li, Viktor Schlegel, Riza Batista-Navarro, and Goran Nenadic. 2023. [Do you hear the people sing? key point analysis via iterative clustering and abstractive summarisation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14064–14080, Toronto, Canada. Association for Computational Linguistics.
- Hao Li, Yizheng Sun, Viktor Schlegel, Kailai Yang, Riza Batista-Navarro, and Goran Nenadic. 2025. [Arg-llada: Argument summarization via large language diffusion models and sufficiency-aware refinement](#). *arXiv preprint arXiv:2507.19081*.
- Hao Li, Yuping Wu, Viktor Schlegel, Riza Batista-Navarro, Tharindu Madusanka, Iqra Zahid, Jiayan Zeng, Xiaochi Wang, Xinran He, Yizhi Li, and Goran Nenadic. 2024. [Which side are you on? a multi-task dataset for end-to-end argument summarisation and evaluation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 133–150, Bangkok, Thailand. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Hyangsuk Min, Yuho Lee, Minjeong Ban, Jiaqi Deng, Nicole Hee-Yeon Kim, Taewon Yun, Hang Su, Jason Cai, and Hwanjun Song. 2025. [Towards multi-dimensional evaluation of LLM summarization across domains and languages](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14417–14450, Vienna, Austria. Association for Computational Linguistics.
- Allen Roush, Yusuf Shabazz, Arvind Balaji, Peter Zhang, Stefano Mezza, Markus Zhang, Sanjay Basu, Sriram Vishwanath, Mehdi Fatemi, and Ravid S Ziv. 2024. [Opendebatevidence: A massive-scale argument mining and summarization dataset](#). *Advances in Neural Information Processing Systems*, 37:34270–34293.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Shahbaz Syed, Roxanne El Baff, Johannes Kiesel, Khalid Al Khatib, Benno Stein, and Martin Potthast. 2020. [News editorials: Towards summarizing long argumentative texts](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5384–5396, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- An Quang Tang, Xiuzhen Zhang, Minh Ngoc Dinh, and Zhuang Li. 2026. [Arqusumm: Argument-aware quantitative summarization of online conversations](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33205–33213.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. [Bartscore: Evaluating generated text as text generation](#). *Advances in neural information processing systems*, 34:27263–27277.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. [Bertscore: Evaluating text generation with bert](#). *arXiv preprint arXiv:1904.09675*.

A Annotation Evaluation

A.1 ADU Segmentation Annotation Evaluation

A.1.1 Evaluation Metrics

We evaluate the quality of ADU segmentation by comparing model outputs with human-corrected annotations. The model output contains ADUs marked by indexed tags (`<AC0>...</AC0>`), while the corrected annotations use equivalent span delimiters without numbering. Each ADU is represented as a text span within the original essay.

Span Matching Criteria. Given a set of predicted ADUs $\mathcal{M} = \{m_1, \dots, m_M\}$ and gold ADUs $\mathcal{G} = \{g_1, \dots, g_G\}$, we evaluate alignment using two matching criteria:

Exact Match. A predicted span m_i is considered an exact match if its normalized textual content is identical to that of a gold span g_j :

$$\text{match}_{\text{exact}}(m_i, g_j) = \mathbf{1}[\text{norm}(m_i) = \text{norm}(g_j)].$$

where the normalization function lowercases text, trims whitespace, and collapses multiple spaces into a single space.

Partial Match. To account for boundary mismatches, we also consider a relaxed overlap-based

criterion. For each predicted span m_i and gold span g_j , we compute their character-level overlap:

$$\text{overlap}(m_i, g_j) = \max(0, \min(e_i, e_j) - \max(s_i, s_j))$$

where (s_i, e_i) and (s_j, e_j) denote the start and end positions of the spans. A predicted span is considered a partial match if its maximum overlap with any gold span exceeds a threshold τ :

$$\text{match}_{\text{partial}}(m_i) = \left[\max_{g_j \in \mathcal{G}} \frac{\text{overlap}(m_i, g_j)}{|m_i|} \geq \tau \right]$$

where $|m_i|$ is the length of the predicted span, and τ is set to 0.5.

Evaluation Metrics. We compute precision, recall, and F1 scores under both exact and partial matching.

Exact Precision and Recall.

$$P_{\text{exact}} = \frac{\# \text{ exact matches}}{M}, R_{\text{exact}} = \frac{\# \text{ exact matches}}{G}$$

Partial Precision and Recall.

$$P_{\text{partial}} = \frac{\# \text{ partial matches}}{M}$$

$$R_{\text{partial}} = \frac{\# \text{ partial matches}}{G}$$

F1 Score.

$$F1 = \frac{2PR}{P+R}$$

Error Analysis. To better understand segmentation behavior, we report additional statistics:

- **Missing ADUs:** the number of gold spans not matched by any predicted span (false negatives).
- **Spurious ADUs:** the number of predicted spans that do not match any gold span (false positives).

Segmentation Bias. We further analyze the model’s segmentation tendency using:

- **Over-segmentation:** $\max(0, M - G)$, indicating the number of excess predicted spans.
- **Under-segmentation:** $\max(0, G - M)$, indicating the number of missing predicted spans.

A.1.2 Discussion

Table 2 evaluates the alignment between LLM-generated ADU annotations and human-corrected annotations, reflecting the correction effort required.

Annotation Quality. On the training and development sets, the LLM shows strong agreement with human annotations, indicating that it produces high-quality initial segmentations. The relatively low numbers of missing and spurious ADUs suggest that only limited corrections are needed in most cases.

Boundary vs Content Corrections. Across all splits, partial scores are consistently higher than exact scores, showing that most corrections involve boundary adjustments rather than missing content. In particular, partial recall remains above 0.92, indicating that the LLM captures nearly all argumentative content.

Annotation Variability. A large discrepancy is observed between the two test annotations: exact F1 drops from 0.93 to 0.53, while partial F1 remains high (0.92). This is accompanied by a substantial increase in missing and spurious ADUs, indicating that the second annotation follows a different segmentation strategy. Thus, disagreement is driven primarily by boundary differences rather than content omission.

Implications. These results show that LLM-generated annotations provide strong initial coverage and significantly reduce manual effort, with human correction mainly refining segmentation boundaries.

A.1.3 Annotation Agreement and Adjudication Analysis

We evaluate the consistency of ADU annotations by comparing two independent annotators and analysing their alignment with adjudicated annotations. Unlike model evaluation, this analysis focuses on the reliability of human annotation and the role of adjudication in resolving disagreements.

Evaluation Metrics. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ denote the sets of ADU spans annotated by annotator 1 and annotator 2, and $\mathcal{C}$ denote the adjudicated output. Each ADU is treated as a text span.

We compute span-level precision, recall, and F1 scores under two matching criteria:

Exact Match. A predicted span is considered correct if its normalized textual content exactly matches a reference span:

$$\text{match}_{\text{exact}}(m_i, g_j) = [\text{norm}(m_i) = \text{norm}(g_j)]$$

Partial Match. To account for boundary variation, we measure overlap between spans. A pre-

Metric	Train (400–799)	Dev	Test (Annot. 1)	Test (Annot. 2)
Exact Precision	0.903	0.888	0.914	0.531
Exact Recall	0.881	0.826	0.951	0.552
Exact F1	0.891	0.854	0.929	0.532
Partial Precision	0.982	0.989	0.942	0.894
Partial Recall	0.960	0.923	0.984	0.997
Partial F1	0.969	0.953	0.959	0.921
Missing ADUs	2.09	3.19	0.71	7.76
Spurious ADUs	1.66	1.97	1.50	8.51
Model ADUs	17.27	17.77	18.00	18.00
Gold ADUs	17.70	18.99	17.21	17.25
Over-segmentation	0.20	0.08	0.94	2.56
Under-segmentation	0.64	1.30	0.15	1.81

Table 2: ADU segmentation annotation evaluation across training, development, and test sets. Test results are reported separately for two independent annotations.

Metric	Agreement	Annot. 1 → Adj.	Annot. 2 → Adj.
Exact Precision	0.515	0.737	0.530
Exact Recall	0.512	0.748	0.534
Exact F1	0.506	0.739	0.527
Partial Precision	0.790	0.890	0.791
Partial Recall	0.801	0.912	0.807
Partial F1	0.782	0.896	0.791
Missing ADUs	8.25	3.95	7.59
Spurious ADUs	8.21	4.56	8.24

Table 3: Annotation agreement between annotators and alignment with adjudicated annotations.

dicted span is considered correct if it sufficiently overlaps with any reference span:

$$\text{match}_{\text{partial}}(m_i) = \left[\max_{g_j \in \mathcal{G}} \frac{\text{overlap}(m_i, g_j)}{|m_i|} \geq \tau \right]$$

where $\tau = 0.5$.

For both criteria, we compute precision, recall, and F1 scores. In addition, we report:

- **Missing ADUs:** reference spans not matched by predictions
- **Spurious ADUs:** predicted spans not matched by references

Results. Table 3 presents the agreement between annotators and their alignment with adjudicated annotations.

Discussion. The results reveal substantial disagreement between annotators under exact matching, with an exact F1 of 0.51. This indicates that annotators frequently disagree on ADU boundaries. However, partial matching yields a considerably higher F1 score of 0.78, suggesting that annotators largely agree on the underlying argumentative content.

This gap between exact and partial scores highlights that disagreement is primarily driven by boundary selection rather than content identification. In other words, annotators tend to identify similar argumentative units but differ in how they segment these units into spans.

When comparing annotations with the adjudicated output, clear differences between annotators emerge. Annotator 1 achieves substantially higher agreement with the adjudicated annotations (exact F1 = 0.74) compared to annotator 2 (exact F1 = 0.53). This is further supported by the lower number of missing and spurious ADUs for annotator 1, indicating that fewer corrections are required.

Overall, the analysis highlights two key challenges in ADU segmentation: (i) the inherent subjectivity of boundary decisions, and (ii) variability in annotator consistency. At the same time, it underscores the importance of adjudication as a normalization step that improves annotation reliability.

A.2 Argument Component Classification

A.2.1 Annotation Analysis

We evaluate argument component (AC) classification by comparing predicted labels with human-corrected annotations at the span level. Each argumentative discourse unit (ADU) is assigned a label (e.g., *claim*, *premise*, *counterargument*), and the task is to correctly classify each ADU.

Evaluation Metrics. We report standard classification metrics, including precision, recall, and F1-score for each class. In addition, we compute:

- **Accuracy:** the proportion of correctly classified ADUs
- **Macro F1:** the average F1-score across all classes, treating each class equally

Metric	Train	Dev	Test (Ann. 1)	Test (Ann. 2)
Accuracy	0.887	0.904	0.924	0.810
Macro F1	0.842	0.844	0.895	0.692

Table 4: Argument component classification annotation analysis across dataset splits.

Class	Precision	Recall	F1
Claim	0.9147	0.7913	0.8486
Counterargument	0.7809	0.9521	0.8580
Major Claim	0.9799	0.8848	0.9299
Premise	0.9509	0.9834	0.9669
Rebuttal	0.8246	0.9216	0.8704

Table 5: Per-class classification analysis on Test (Annotation 1).

Macro F1 is particularly important in our setting, as it accounts for class imbalance and ensures that performance on less frequent classes is not overshadowed by dominant categories (e.g., *premise*).

Results. Table 4 presents overall classification performance across the training, development, and test sets. The test set is evaluated against two independent annotations to assess robustness to annotation variability.

To provide a more detailed analysis, Table 5 reports per-class performance on the first test annotation.

Discussion. The results reflect the alignment between initial annotations and corrected annotations, and can be interpreted as an indicator of correction effort. Higher scores correspond to fewer required corrections.

Score is high across different sets, indicating that most labels are already correct and require minimal refinement.

At the class level, labels such as *premise* and *major claim* show stable and consistent annotations, while less frequent or more ambiguous classes (e.g., *rebuttal*) require more correction.

A.2.2 Annotation Agreement and Adjudication Analysis

To assess the reliability of the annotation process, we measure inter-annotator agreement (IAA) between two independent annotators at the level of argumentative components (ACs). We report both raw agreement and chance-corrected metrics.

Raw agreement is computed as:

$$\text{Agreement} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(y_i^{(1)} = y_i^{(2)}), \quad (1)$$

where $y_i^{(1)}$ and $y_i^{(2)}$ denote the labels assigned by annotator 1 and annotator 2 for the i -th AC, and N is the total number of annotated components.

To account for agreement occurring by chance, we compute Cohen’s κ :

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad (2)$$

where p_o is the observed agreement and p_e is the expected agreement under independence assumptions.

We also report Krippendorff’s α , a more general reliability coefficient that is robust to label imbalance and applicable to multiple annotators.

Annotator Accuracy We treat the adjudicated annotations as the gold standard. The accuracy of each annotator is computed as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(y_i^{(ann)} = y_i^{(gold)}), \quad (3)$$

where $y_i^{(ann)}$ is the annotator’s label and $y_i^{(gold)}$ is the adjudicated label for the i -th AC.

Label-wise Evaluation To provide a fine-grained analysis of annotation performance, we compute precision, recall, and F1-score for each argument component type (major claim, claim, premise, counterargument, rebuttal). We also report macro-averaged and weighted-averaged scores to account for class imbalance.

In addition, confusion matrices are used to identify systematic labeling errors across categories.

Results Our test set contains a total of 1,542 annotated argumentative components. The inter-annotator agreement reaches 77.82%, with Cohen’s $\kappa = 0.65$ and Krippendorff’s $\alpha = 0.65$, indicating substantial agreement.

In terms of annotator reliability, Annotator 1 achieves an accuracy of 86.77%, while Annotator 2 achieves 82.94% with respect to the adjudicated gold standard.

Analysis We observe substantial variation across argument component types. Premises are identified most reliably, achieving F1 scores above 0.93 for

Metric	Score
Agreement	77.82%
Cohen’s κ	0.65
Krippendorff’s α	0.65
Accuracy (Annotator 1)	86.77%
Accuracy (Annotator 2)	82.94%

Table 6: Inter-annotator agreement and annotator accuracy.

both annotators. In contrast, rebuttals are the most challenging category, particularly for Annotator 2, with an F1 score of 0.44.

Confusion matrix analysis reveals that the most common source of disagreement lies in distinguishing claims from premises. This reflects the inherent difficulty of separating assertions from supporting evidence. Additionally, rebuttals are frequently confused with claims and premises, highlighting the importance of contextual and relational information in argument mining.

A.3 Structured Summary Annotation and Correction Analysis

We adopt a hybrid annotation pipeline in which a large language model (LLM) is used to generate initial structured argument summaries, followed by human correction. In this section, we will assess the amount of human effort required to correct the LLMs’ outputs.

Given an input essay annotated with argumentative discourse units (ADUs) and the component labels, the LLM generates a structured outline containing major claims, claims, premises, counterarguments, and rebuttals. Human annotators then revise the outputs to ensure compliance with strict task constraints:

- each ADU must be represented exactly once,
- no omission, duplication, merging, or splitting is allowed,
- the output must follow a predefined hierarchical structure,
- no hallucinated content is permitted.

The corrected outputs are treated as references, and the differences between model outputs and corrected outputs are interpreted as the required correction effort.

A.3.1 Evaluation Metrics

We quantify correction effort using a combination of edit-based, structural, and content-based metrics.

Edit-Based Metrics We measure the degree of modification required using normalized edit distance. For a model output y^{model} and corrected output y^{gold} , we compute:

$$D^{norm} = \frac{D(y^{model}, y^{gold})}{|y^{gold}|} \quad (4)$$

where $D(\cdot, \cdot)$ denotes edit distance and $|y^{gold}|$ is the length of the reference.

We report:

- **Word-level edit distance (normalized):** lexical correction effort,
- **Line-level edit distance (normalized):** structural editing effort.

Structural Accuracy We evaluate the consistency of argument structure by comparing structural slots:

$$Accuracy_{struct} = \frac{\# \text{ matched structural units}}{\text{total units}} \quad (5)$$

This captures whether corrections involve reorganization of claims, premises, or relations.

ADU Coverage We measure completeness as:

$$\text{Coverage} = \frac{\# \text{ ADUs correctly represented}}{\# \text{ total ADUs}} \quad (6)$$

This penalizes missing or improperly mapped argumentative units.

Hallucination We estimate hallucination by comparing the textual units extracted from the model output and the corrected output. Specifically, we treat each non-header line (i.e., excluding lines such as “Major claim:” or “Claim k:”) as a rewritten argumentative unit.

Let U^{model} and U^{gold} denote the sets of extracted units from the model output and corrected output, respectively. Hallucinated units are defined as those present in the model output but not aligned with any unit in the corrected output. The hallucination count is computed as:

$$\text{Hallucination} = |U^{model} \setminus U^{gold}| \quad (7)$$

This metric captures the number of extra or unsupported statements introduced by the model. Lower values indicate higher faithfulness to the input ADUs and corrected structure.

Acceptance Rate We compute the proportion of outputs that require no correction:

$$\text{Acceptance Rate} = \frac{\# \text{unchanged outputs}}{\# \text{total outputs}} \quad (8)$$

This directly reflects annotation efficiency in the pipeline.

A.3.2 Results

Table 7 summarizes correction effort on the development set and two independently annotated test sets.

On the development set, the LLM outputs require minimal correction. The normalized word edit distance is low (0.08), and 80% of outputs are accepted without modification. Structural accuracy (0.93) and coverage (0.92) are also high, indicating that the LLM produces outputs that largely satisfy the task constraints.

On the test set, correction effort increases substantially. Under Annotation 1, normalized word edit distance rises to 0.63, structural accuracy drops to 0.34, and coverage decreases to 0.49, suggesting extensive modifications. Annotation 2 yields improved results, with lower edit distance (0.26) and higher structural accuracy (0.65), but still requires non-trivial correction.

A.3.3 Analysis of Correction Necessity

To understand the nature of corrections, we manually analyze a subset of test examples. We observe that a large proportion of corrections are not strictly necessary under the task definition.

In many cases, model outputs already satisfy all hard constraints of the annotation task:

- all ADUs are present,
- no duplication or hallucination occurs,
- the required format is followed.

However, annotators often apply additional modifications, including:

- reordering claims or structural components,
- regrouping premises under different claims,

- rewriting content without changing semantic meaning.

These changes reflect alternative interpretations of argument structure rather than corrections of objective errors.

A.3.4 Discussion

The results highlight a key distinction between *necessary* and *unnecessary* corrections.

First, conventional metrics such as edit distance and acceptance rate implicitly assume that all corrections correspond to genuine errors. However, our analysis shows that many corrections are preference-driven. As a result, these metrics may overestimate the true amount of required human effort.

Second, the results suggest that LLM-generated outputs are often structurally valid, even when metrics indicate high correction effort. In many cases, the required modifications involve alignment with annotator preferences rather than fixing fundamental violations.

Overall, our findings indicate that, given accurate ADU segmentation and argument component classification, LLMs are capable of generating high-quality structured summaries. The majority of subsequent human edits serve to refine presentation and improve clarity rather than correct fundamental structural errors.

A.3.5 Annotation Agreement

We evaluate agreement between annotators at three complementary levels: **coverage**, **structure**, and **semantic similarity**.

Metrics Coverage F1 (ADU-level agreement).

This metric measures whether the same argumentative discourse units (ADUs) appear in both annotations. Let C_1 and C_2 denote the sets of ADUs from two annotations. We compute:

$$\text{Precision} = \frac{|C_1 \cap C_2|}{|C_1|}, \quad \text{Recall} = \frac{|C_1 \cap C_2|}{|C_2|} \quad (9)$$

$$\text{Coverage F1} = \frac{2 \cdot P \cdot R}{P + R} \quad (10)$$

Structure F1 (edge agreement). Each structured summary is converted into a graph where nodes are ADUs and edges represent argument relations (e.g., claim $\rightarrow$ premise, claim $\rightarrow$ counterargument). Let E_1 and E_2 denote the edge sets:

$$\text{Structure F1} = \frac{2 \cdot |E_1 \cap E_2|}{|E_1| + |E_2|} \quad (11)$$

Dataset	Acc.	Word-ED _{norm}	Line-ED _{norm}	Struct.	Coverage	Halluc.
Dev	0.80	0.08	0.09	0.93	0.92	1.59
Test (Ann1)	0.32	0.63	0.62	0.34	0.49	11.61
Test (Ann2)	0.25	0.26	0.36	0.65	0.81	5.51

Table 7: Correction effort for LLM-assisted structured annotation.

Semantic Similarity. We measure semantic agreement using sentence embeddings. Each ADU is encoded using `all-mpnet-base-v2`, and cosine similarity is computed. For each ADU in C_1 , we take the maximum similarity with any ADU in C_2 and average:

$$\text{Semantic} = \frac{1}{|C_1|} \sum_{c \in C_1} \max_{c' \in C_2} \cos(c, c') \quad (12)$$

Metric	Value
Coverage F1	0.3348
Structure F1	0.2266
Semantic Similarity	0.8568

Table 8: Inter-annotator agreement results.

Results

Analysis We observe high semantic agreement but substantially lower coverage and structural agreement. This indicates that annotators generally agree on the meaning of arguments but diverge in how they segment and organize them into structured representations.

A.3.6 Agreement with Adjudicated Gold

We evaluate each annotator against the adjudicated gold standard using the same metrics.

Metric	Annotator 1	Annotator 2
Coverage F1	0.7769	0.5472
Structure F1	0.7141	0.4404
Semantic Similarity	0.9413	0.9349

Table 9: Agreement of annotators with adjudicated gold.

Analysis Both annotators achieve high semantic similarity with the adjudicated outputs (> 0.93), indicating consistent interpretation of content. However, structural agreement differs significantly, with Annotator 1 substantially closer to the adjudicated structure.

Correction Effort Analysis We quantify the effort required to transform annotations into the adjudicated gold using correction-based metrics from Appendix A.3.1.

Metrics Normalized Edit Distance. Measures text-level modification cost:

$$\text{Edit Distance} = \frac{\text{Levenshtein}(S_{\text{ann}}, S_{\text{gold}})}{|S_{\text{gold}}|} \quad (13)$$

Structural Accuracy. Measures correctness of predicted structural relations:

$$\text{Structural Accuracy} = \frac{|E_{\text{ann}} \cap E_{\text{gold}}|}{|E_{\text{gold}}|} \quad (14)$$

Hallucination Count. Measures extra components produced by the annotator:

$$\text{Hallucination} = |C_{\text{ann}} \setminus C_{\text{gold}}| \quad (15)$$

Metric	Annotator 1	Annotator 2
Edit Distance	0.2473	0.3978
Structural Accuracy	0.7089	0.4221
Hallucination Count	3.21	6.14

Table 10: Correction effort required to align annotations with adjudicated gold.

Results

Analysis Annotator 1 requires substantially less correction effort than Annotator 2, as reflected by lower edit distance and higher structural accuracy.

A.3.7 Discussion

Across all evaluations, we observe a consistent pattern:

- High semantic agreement (> 0.85)
- Moderate coverage agreement
- Low structural agreement (≈ 0.23 F1)

This indicates that disagreement arises primarily from differences in **argument organization** rather than **content interpretation**.

Furthermore, correction-based metrics show that even high-quality annotations require non-trivial edits, indicating that correction effort reflects **structural ambiguity** rather than purely annotation errors.

A.3.8 Key Insight

We conclude that: Structured argument annotation exhibits high semantic consistency but substantial structural variability.

This highlights the importance of:

1. Adjudication to establish a canonical structure
2. Evaluation frameworks that explicitly account for structural ambiguity

A.4 Abstractive Summary Annotation and Correction Analysis

A.4.1 Task Definition

We evaluate the quality of LLM-generated **abstractive summaries** in the final stage of the argumentative text summarization (ATS) pipeline.

Given a model-generated summary y_{model} , produced from the structured summary, and a human-corrected summary y_{gold} , we measure the **correction effort** required to transform the generated output into the corrected version.

Unlike earlier stages, this task involves **free-form natural language generation**, where multiple valid outputs may correspond to the same underlying argumentative structure.

A.4.2 Evaluation Metrics

We quantify correction effort using text-based metrics.

Normalized Word-Level Edit Distance. We compute the word-level Levenshtein distance and normalize by the reference length:

$$D_{\text{norm}} = \frac{D_{\text{word}}(y_{\text{model}}, y_{\text{gold}})}{|y_{\text{gold}}|} \quad (16)$$

This metric captures the degree of textual modification required.

Correction Rate. We compute the proportion of examples that require any modification:

$$\text{Correction Rate} = \frac{\# \text{examples modified}}{\# \text{total examples}} \quad (17)$$

Word Difference. We measure the absolute difference in summary length:

$$\Delta_{\text{words}} = ||y_{\text{model}}| - |y_{\text{gold}}|| \quad (18)$$

This reflects differences in compression and summarization style.

A.4.3 Results

Metric	Dev	Test (Ann. 1)	Test (Ann. 2)
Total Examples	100	100	100
Correction Rate	8.00%	47.00%	24.00%
Avg Norm Edit Dist	0.0314	0.7242	0.1039
Median Norm Edit Dist	0.0171	0.8125	0.0916
Avg Word Difference	0.25	7.60	4.92

Table 11: Correction effort for abstractive summary generation across dataset splits.

A.4.4 Discussion

High-quality outputs with minimal necessary corrections. Across all splits (Dev, Test Annotation 1, and Test Annotation 2), we observe that **most corrections are not strictly necessary under the task definition**. In many cases, LLM-generated summaries already preserve the core argumentative content, remain consistent with the structured summary, and satisfy key task constraints.

On the development set, this is reflected in the very low edit distance (0.0314) and low correction rate (8%), indicating that most corrections involve only minor lexical or stylistic refinements.

High correction effort does not imply poor model quality. On Test Annotation 1, we observe a high correction rate (47%) and large edit distance (0.7242). However, these corrections are predominantly driven by extensive rewriting rather than the correction of invalid outputs. In contrast, Test Annotation 2 yields significantly lower edit distance (0.1039) and correction rate (24%) for the same model outputs, indicating fewer required modifications.

Corrections reflect annotator preferences rather than errors. The discrepancy between the two test annotations demonstrates that correction effort is highly sensitive to annotator-specific preferences. Most corrections correspond to alternative phrasings, differences in abstraction level, or sentence reorganization, rather than missing content or incorrect reasoning.

Abstractive summaries exhibit high variability.

These results show that abstractive summarization is inherently a **one-to-many mapping task**: multiple summaries can validly express the same underlying argument. As a result, identical model outputs may be judged differently by annotators, leading to substantial variation in edit distance.

Key Insight. Abstractive summary generation exhibits **high semantic correctness but substantial variability in surface realisation**, where most human corrections reflect alternative valid outputs rather than objective errors.

A.4.5 Implications

These findings suggest that:

- LLMs are effective at semantic compression even without correction.
- Correction-based metrics (e.g., edit distance) conflate true errors with acceptable variation.
- Evaluation of argument summarization should go beyond text overlap and incorporate structure-aware and semantic measures.

A.5 Key Point Generation Annotation and Correction Analysis

A.5.1 Task Definition

We evaluate the quality of LLM-generated **key points** in the key point generation stage.

Given: - a set of generated key points $\mathcal{K}_{\text{model}}$, divided into *pro* and *con* categories - a human-corrected set $\mathcal{K}_{\text{gold}}$

we measure the **correction effort** required to align the generated outputs with the corrected annotations.

Each key point is treated as a short textual unit, and evaluation accounts for both:

- **set-level differences** (additions, deletions, stance changes)
- **textual differences** (modifications via paraphrasing)

A.5.2 Evaluation Metrics

We decompose correction effort into four types of operations.

Additions. Key points present in the corrected set but missing from the model output:

$$\text{Add} = |\mathcal{K}_{\text{gold}} \setminus \mathcal{K}_{\text{model}}| \quad (19)$$

Deletions. Key points generated by the model but removed during correction:

$$\text{Del} = |\mathcal{K}_{\text{model}} \setminus \mathcal{K}_{\text{gold}}| \quad (20)$$

Modifications. Key points that are retained but require textual edits. We compute the normalized word-level edit distance:

$$D_{\text{norm}} = \frac{D_{\text{word}}(k_{\text{model}}, k_{\text{gold}})}{|k_{\text{gold}}|} \quad (21)$$

A key point is considered *modified* if:

$$D_{\text{norm}} > \tau \quad (22)$$

where τ is a threshold (set to 0.15).

Moves (Stance Changes). Key points that change category between *pro* and *con*:

$$\begin{aligned} \text{Move} = & \left| \{k \mid k \in \text{pro}_{\text{model}}, k \in \text{con}_{\text{gold}}\} \right. \\ & \left. \cup \{k \mid k \in \text{con}_{\text{model}}, k \in \text{pro}_{\text{gold}}\} \right| \quad (23) \end{aligned}$$

No-Correction Rate. We compute the proportion of examples requiring no modification:

$$\text{NoCorrectionRate} = \frac{N_{\text{unchanged}}}{N_{\text{total}}} \quad (24)$$

where $N_{\text{unchanged}}$ is the number of examples with no changes and N_{total} is the total number of examples.

Key Point Change Rate. We measure the proportion of key points that are added, deleted, or modified:

$$\text{KP Change Rate} = \frac{\# \text{changed key points}}{\# \text{total key points}} \quad (25)$$

A.5.3 Results

Metric	Train	Dev	Test 1	Test 2
Num. Items	400	100	100	100
No-Correction Rate	0.785	0.860	0.940	0.850
Avg Edit Distance	0.787	0.740	0.290	2.260
Avg KP Change Rate	0.117	0.092	0.048	0.303

Table 12: Correction statistics for key point generation across dataset splits.

Error Type Distribution.

- Train: redundancy/completeness (65), stance error (19)

- Dev: redundancy/completeness (14), stance error (3)
- Test 1: redundancy/completeness (6)
- Test 2: redundancy/completeness (15)

A.5.4 Discussion

Most outputs require no correction. Across all splits, the majority of generated key point sets require no correction, as indicated by high no-correction rates ranging from 78.5% to 94.0%. This suggests that LLMs are highly effective at identifying core argumentative key points.

Dominance of redundancy and completeness corrections. Across all splits, the majority of corrections fall under **redundancy or completeness**, suggesting that:

- the model occasionally generates overlapping key points
- or slightly incomplete coverage of argument space

In contrast, **stance errors are rare**, particularly in the test sets, indicating that the model reliably captures argumentative polarity.

Most corrections are not strictly necessary. Across all datasets, we observe that most corrections are not strictly required under the task definition. In many cases, the generated key points already capture the intended argumentative meaning, and corrections primarily reflect alternative valid representations, including paraphrasing, changes in granularity, or stylistic adjustments.

Key Insight. Key point generation exhibits **high semantic coverage with limited structural error**, where most human corrections reflect representation-level variation rather than objective mistakes.

A.5.5 Implications

These findings suggest that:

- LLMs are effective at extracting semantically valid and stance-consistent key points.
- Correction effort is dominated by **representation-level differences**, rather than missing or incorrect content.

- Standard evaluation methods based on exact matching may overestimate error by conflating necessary corrections with acceptable variation.

A.6 Key Point Matching Annotation

Unlike other stages, key point matching is annotated entirely using LLMs without human correction.

Given a set of argumentative discourse units (ADUs) and a set of key points, the task is to assign each ADU a stance (pro or con) and match it to zero or more key points. This task focuses on identifying semantic relationships between argument units and abstract key points.

Rationale. Key point matching differs from generation tasks such as key point generation and summarization. While generation tasks require constructing new outputs and benefit from human correction, matching can be formulated as a structured prediction problem mapping argument units to predefined key points.

We adopt LLM-only annotation for this stage to ensure scalability.

Preliminary Analysis. To assess the reliability of LLM-generated matches, we conducted preliminary human annotation studies. These studies show that **most matching decisions require no correction**, indicating that LLM-generated annotations are generally consistent with human judgments.

The remaining corrections typically involve:

- ambiguous semantic alignment between arguments and key points,
- borderline cases of partial relevance,
- occasional stance inconsistencies.

Implications. Based on this observation, we treat key point matching annotations as **high-quality LLM-derived annotations**. While some noise remains, the majority of annotations are already valid, reducing the need for costly human correction.

B Dataset Statistics

Overview. The dataset consists of 800 training essays, as well as 100 development essay and 100 test. The training set is further divided into two subsets: 400 LLM-generated annotations (indices 0-399) and 400 human-corrected annotations (indices 400-799).

Global Statistics. Across all splits, essays are of consistent length, averaging approximately 291–293 words. Each essay contains a dense argumentative structure with 16–19 argument discourse units (ADUs) and is associated with around 7 key points.

Split Comparison. Table 13 summarizes the dataset statistics across all subsets.

Argument Component Distribution. The distribution of argument components is highly consistent across all splits, as shown in Table 14. Premises dominate, followed by claims, while major claims, counterarguments, and rebuttals form smaller proportions.

Key Point Matching. Each argument is aligned with approximately 1.07–1.19 key points on average.

Discussion. Overall, the dataset demonstrates strong consistency across training, development, and test splits. The similarity in essay length, argument structure, and component distributions suggests a well-balanced dataset.

C Fine-tuning and prompting configurations

We use HuggingFace SFTTrainer and PEFT⁸ for all the experiments. The hyperparameters for PEFT is shown in Table 15. The training hyperparameters are reported in Table 16. For generation, we use `do_sample = False`.

For Qwen3 235B and DeepSeek V3.2, we use greedy decoding for all the experiments, `temperature = 0.0`, `top_p = 1.0`. All the API-based experiments were conducted through the OpenRouter platform (<https://openrouter.ai>), which provides unified access to multiple model providers. The model and identifier are shown in Table 17.

D Artifact Documentation and Responsible Use

D.1 Artifact Use Consistent with Intended Use

We ensure that the use of all existing artifacts and the creation of new artifacts are consistent with their intended use and licensing conditions.

Use of Existing Artifacts. The ArgSum-X dataset is constructed based on the ArgEssay dataset (Bao et al., 2022). The original dataset

is intended for research use, and our usage strictly adheres to these conditions. We do not modify or redistribute the original raw data in ways that violate its licensing terms. All additional annotations introduced in this work, including argumentative discourse unit (ADU) segmentation, argument component labels, structured summaries, abstractive summaries, and key points, are produced solely for research purposes and remain consistent with the original dataset’s intended use.

Created Artifacts and Intended Use. We explicitly define the intended use of ArgSum-X as:

- Research in argument mining, argument summarization, and natural language processing
- Development and evaluation of machine learning models

The dataset is not intended for direct deployment in real-world decision-making systems or commercial applications without further validation and compliance checks.

Compatibility with Original Access Conditions. All derived artifacts are designed to remain compatible with the original dataset’s access conditions. In particular:

- Any restrictions associated with the original dataset are preserved and inherited by ArgSum-X
- We do not relax or override the original licensing terms
- Any redistribution includes proper attribution and reference to the original dataset source

Restrictions on Downstream Use. To support responsible use:

- Derived data should be used within research contexts unless explicitly permitted otherwise
- Users must comply with both the original dataset license and the ArgSum-X license
- We discourage misuse, including applications that could lead to misinformation or harmful outcomes

D.2 Dataset License

ArgSum-X is released under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license.

⁸https://huggingface.co/docs/trl/v0.26.1/en/sft_trainer#trl.SFTTrainer

Subset	Essay length	ADUs	Key Points	KP / Arg	Abstractive summary length
Train (LLM, 0–399)	291.21	17.71	7.36	1.19	104.41
Train (Corrected, 400–799)	290.77	17.70	7.05	1.07	105.85
Dev	292.60	18.99	7.21	1.11	105.05
Test	292.57	16.60	7.64	1.07	109.38

Table 13: Comparison of dataset statistics across different subsets. KP / Arg: number of matched key points per argument. Essay length and abstractive summary length are in words.

Subset	Premise	Claim	Major	Counter	Rebuttal
Train (LLM)	56.01%	20.70%	8.89%	10.10%	4.29%
Train (Corrected)	54.03%	23.70%	9.33%	9.83%	3.11%
Dev	58.56%	23.38%	7.69%	7.85%	2.53%
Test	52.41%	26.33%	10.18%	8.25%	2.83%

Table 14: Distribution of argument components across dataset splits.

Hyperparameter	Value
r(rank)	8
lora_alpha	16
bias	none
target_module	all_linear

Table 15: Hyperparameters for PEFT.

Hyperparameter	value
max_length	1024
per_device_train_batch_size	8
gradient_accumulation_steps	1
num_train_epochs	5
lr_scheduler_type	cosine
warmup_ratio	0.1
learning_rate	1e-4
bf16	True

Table 16: Training hyperparameters used in the experiments.

License Terms. Under this license, users are free to:

- **Share** — copy and redistribute the dataset in any medium or format
- **Adapt** — remix, transform, and build upon the dataset

under the following condition:

- **Attribution** — users must provide appropriate credit to this work and the original data source (ArgEssay dataset), and indicate if changes were made

Full license text: <https://creativecommons.org/licenses/by/4.0/>

Model	Identifier
Qwen 3 235B	qwen/qwen3-235b-a22b-2507
DeepSeek V3.2	deepseek/deepseek-v3.2

Table 17: Identifier for the models we use in the prompting experiments.

Compatibility with Source Data. ArgSum-X is derived from the ArgEssay dataset, and therefore:

- The dataset inherits any applicable restrictions from the original source
- Users must comply with both the ArgEssay license and CC BY 4.0
- In cases of conflict, the original dataset’s license takes precedence

Usage Restrictions. Although CC BY 4.0 permits broad reuse, we recommend the following:

- Use primarily for research and academic purposes
- Avoid deployment in high-stakes or real-world decision-making systems without further validation
- Do not use the dataset in ways that could cause harm or misinformation

D.3 Documentation of Artifacts

We provide comprehensive documentation describing the dataset’s domain, language, structure, and coverage characteristics.

Domain Coverage. ArgSum-X consists of student-written argumentative essays covering general-purpose topics such as technology, education, environment, and society. These essays reflect common argumentative writing tasks and are not limited to a specialized domain.

Language. All essays and annotations are in English, making the dataset suitable for English-language argument mining research but limiting its applicability to multilingual settings.

Linguistic and Argumentative Phenomena. The dataset captures a wide range of linguistic and discourse-level phenomena relevant to argumentation, including:

- Argumentative discourse unit (ADU) segmentation
- Argument component classification (major claim, claim, premise, counterargument, rebuttal)
- Hierarchical argument structure (e.g., claim–premise relations, counterargument–rebuttal pairs)
- Structured summaries representing global argumentative organization
- Abstractive summaries preserving argumentative meaning
- Key point generation and argument–key point alignment

Dataset Size and Structure. The dataset contains 1,000 essays split into 800 training, 100 development, and 100 test instances.

Each essay typically includes:

- Approximately 16–19 argumentative discourse units
- Structured argument annotations
- Abstractive summaries of approximately 80–120 words
- Around 7 key points per essay

D.4 Ethical Considerations and Responsible Use

We include documentation to support responsible use and interpretation of the dataset.

Annotation and Data Collection. The dataset is constructed via a hybrid LLM–human annotation pipeline, combining automated generation with human correction and adjudication. This design balances scalability with quality but may introduce subtle biases from both annotators and models.

Bias and Representation. Due to the source and annotation process:

- Linguistic style reflects student essay conventions
- Argument structures may reflect annotator preferences in cases of ambiguity
- Some disagreement arises from alternative valid structural interpretations rather than objective errors

Intended Scope. The dataset is intended to:

- Support reproducible research in argument mining and summarization
- Enable development and evaluation of structure-aware models
- Provide a benchmark for studying reasoning and explainability

Responsible Usage Recommendations. We recommend that users:

- Interpret outputs with awareness of structural ambiguity
- Avoid deploying models trained on this dataset in high-stakes settings without further validation
- Respect licensing and attribution requirements

E Annotator recruitment and payment

We recruit all the annotators on the Prolific platform. We select annotators who are native English speakers and have passed the AI test. The annotators are paid 9 pounds per hour, the total cost for the whole dataset is around 1000 pounds.

F Prompts

F.1 Prompts for LLM annotation

Prompt for LLM ADU segmentation:

You are an expert in argument mining. Your task is to annotate the following essay with inline Argumentative Discourse Unit (ADU) tags.

You must insert tags <AC0>...</AC0>, <AC1>...</AC1>, <AC2>...</AC2>, ... directly into the original essay text. The output must be the full essay with these tags added.

1. WHAT IS AN ADU?

An ADU is the *smallest argumentative unit* that expresses exactly ONE argumentative function:

- major claim
- claim
- premise
- counterargument
- rebuttal

ADUs contain argumentative content such as:

<AC0>modern buildings add variety and interest</AC0>
<AC1>banning private cars would reduce pollution</AC1>
<AC2>AI reduces accidents caused by human error</AC2>
<AC3>some people believe this harms cultural identity</AC3>
<AC4>this argument is unconvincing</AC4>

2. WHAT IS *NOT* AN ADU?

1) Rhetorical connectors (do NOT include inside ADUs)

- However,
- Therefore,
- On the other hand,
- For example,
- Moreover,
- As a result,
- Because (when functioning only as a linker)

Correct:

However, <AC0>improving public transport is more effective</AC0>

Incorrect:

<AC0>However, improving public transport is more effective</AC0>

2) Reporting frames (never wrap)

- Many people believe that
- Some argue that
- Researchers claim that
- It is widely thought that

Correct:

Many people believe that <AC0>banning cars is necessary</AC0>

Incorrect:

<AC0>Many people believe that banning cars is necessary</AC0>

3) Structural / meta-discourse text

Do NOT wrap:

- In this essay I will discuss...
- I will compare these issues...
- To begin with...

- As mentioned earlier...
- In conclusion...

Correct:

In this essay I will compare these issues.

Incorrect:

<AC0>In this essay I will compare these issues</AC0>

4) Whole sentences containing multiple argumentative ideas
A single sentence may contain several ADUs.

Example:

The government restricts building heights because they want to protect the temple.

Correct segmentation:

<AC0>The government restricts building heights</AC0>
because <AC1>they want to protect the temple</AC1>

Incorrect:

<AC0>The government restricts building heights because they want to protect the temple</AC0>

3. SPLITTING & MERGING

Split the text when it contains more than one argumentative idea.

General principle:

If X and Y express DIFFERENT argumentative ideas, then:

<ACk>X</ACk> connector <AC(k+1)>Y</AC(k+1)>

Examples (conceptual, not templates):

- <AC0>X is a claim</AC0> because <AC1>Y is a separate reason</AC1>
- <AC0>X is a counterargument</AC0> but <AC1>Y is a rebuttal</AC1>
- <AC0>X is evidence</AC0> which means <AC1>Y is an inference</AC1>

Merge when text expresses **one single argumentative idea**.

Incorrect:

<AC0>AI reduces mistakes</AC0> <AC1>and increases safety</AC1>

Correct:

<AC0>AI reduces mistakes and increases safety</AC0>

IMPORTANT:

These examples illustrate *meaning*, not rigid grammatical patterns.

Do NOT split just because “and” or “because” appears.

Split ONLY when the argumentative meaning changes.

4. MAIN REQUIREMENTS

1) Preserve the original essay EXACTLY (no editing, paraphrasing, or reordering).

- 2) ADUs must be the smallest meaningful argumentative clauses.
- 3) Do NOT wrap rhetorical connectors, reporting frames, or structural phrases.
- 4) Punctuation belongs inside an ADU only if it is part of the argumentative idea.
- 5) Number tags consecutively: <AC0>, <AC1>, <AC2>, ... (no gaps).
- 6) Output ONLY the fully annotated essay (no explanations).
- 7) If the essay contains no argumentative content, wrap one central sentence in <AC0>...</AC0> and leave everything else unchanged.

5. DEMONSTRATION EXAMPLE

Input:

Some people argue that banning private cars is necessary to reduce pollution.
However, improving public transport is more effective.
For example, many cities still lack sufficient metro and bus coverage, which makes a full ban impractical.

Correct output:

Some people argue that <AC0>banning private cars is necessary to reduce pollution</AC0>.
However, <AC1>improving public transport is more effective</AC1>.
For example, <AC2>many cities still lack sufficient metro and bus coverage</AC2>, <AC3>which makes a full ban impractical</AC3>.

Now annotate this essay:

Prompt for LLM argument component classification:

You are an expert in argument mining. Return a strict JSON mapping from AC ids to labels.

Task:

Given an essay where argumentative discourse units are marked as <Ack>...</Ack> (k = 0,1,2,...), assign one label to each AC from this set (exact strings):

- major claim
- claim
- premise
- counterargument
- rebuttal

Definitions:

major claim:

The author's overall thesis or main position about the topic of the entire essay. This is the central conclusion the essay aims to justify.

claim:

A sub-claim or intermediate conclusion that supports the major claim. A point the author asserts at paragraph or local level, typically supported by premises.

premise:

A supporting reason, justification, explanation, example, or piece of evidence that supports a claim or the major claim.

counterargument:

A statement presenting an opposing view or objection (not the author's belief).

rebuttal:

The author's response that refutes or counters a counterargument, restoring support for the author's position.

Guidelines:

- Classify by argumentative function, not by rhetorical markers.
- Each <Ack> must receive exactly one label from the list above.
- If an ADU expresses the author's global thesis, use major claim; otherwise, if it is a local asserted point, use claim.
- Reasons/evidence/examples are premise. Opposing view is counterargument. The author's comeback to that opposing view is rebuttal.

Output format (strict JSON, no extra text):

```
{
  "<AC0>": "LABEL",
  "<AC1>": "LABEL",
  ...
}
```

Constraints:

- Use the exact label strings (lowercase) listed above.
- Include all AC ids present in the essay.
- Do not return the essay text or any prose.

Here is an example:

Essay (with AC tags):

Some people argue that <AC0>banning private cars is the only way to reduce urban pollution</AC0>.
However, <AC1>this view ignores the fact that public transportation systems are still underdeveloped in many major cities</AC1>. For example, <AC2>several metropolitan areas lack sufficient bus and metro coverage</AC2>. Instead, <AC3>cities should focus on improving public transit and investing in cleaner vehicle technologies</AC3>. For instance, <AC4>electric buses and low-emission cars have already reduced pollution levels in many regions</AC4>. In conclusion, <AC5>enhancing transportation infrastructure and adopting clean technologies is the most effective long-term solution to urban pollution</AC5>.

Labels (JSON):

```
{
  "<AC0>": "counterargument",
  "<AC1>": "rebuttal",
  "<AC2>": "premise",
  "<AC3>": "claim",
  "<AC4>": "premise",
  "<AC5>": "major claim"
}
```

Now classify this essay:

<ESSAY_WITH_AC_TAGS>

Prompt for LLM structured summary generation:

You are an expert in argument mining and structured summarization. Your task is to transform an AC-tagged, AC-typed essay into a STRICT structured argument outline. This outline will later be used to generate an abstractive summary, so it must be complete, canonical, and faithful.

INPUT:

- Essay text with <ACi>...</ACi> tags.
- A mapping of each <ACi> to one of: major claim, claim, premise, counterargument, rebuttal.

GOAL:

Produce a structured outline that includes exactly one rewritten line for every ADU, capturing the argumentative structure fully and without hallucination.

STRICT OUTPUT FORMAT:

Major claim:

<rewritten major claim>

Claim 1:

<rewritten claim>

Premise: <text>

Premise: <text>

...

Counterargument: <text or None>

Rebuttal: <text or None>

...

(Premise, Counterargument, Rebuttal may appear zero or many times)

Claim 2:

...

RULES (MUST FOLLOW):

1. Each ADU must be represented exactly once in the output. No omissions, no duplicates, no merging, no splitting.
2. Rewrite each ADU in a concise, canonical, non-rhetorical style.
 - Keep only the essential argumentative meaning.
3. Use ONLY the information in the ADUs. Do not invent or infer additional claims, reasons, or examples.
4. The major claim must be the ADU labeled "major claim."
5. Claims must appear in the same order as in the original essay.
6. Under each claim:
 - First list all supporting Premise lines.
 - Then list all Counterargument lines (zero or many).
 - Immediately after each counterargument, list its Rebuttal lines (zero or many).
7. If a claim has no counterarguments, write exactly: Counterargument: None
 - If a counterargument has no rebuttal, write exactly: Rebuttal: None
8. Do NOT output AC tags or AC IDs.
9. Output only the structured outline. No comments, no explanations.

Now generate the structured argument summary for the following essay with the component labels:

Prompt for LLM abstractive summary generation:

You are an expert in argument-focused abstractive summarization.

Your task is to generate a clear, concise, and faithful summary using ONLY the structured argument outline provided.

You are given a structured argument summary extracted from an essay.

Your job is to write a fluent, coherent abstractive summary (80-120 words) that faithfully follows the structure.

Requirements:

1. Use ONLY the information contained in the structured outline.
2. Do NOT bring in any new claims, evidence, studies, examples, or reasoning.
3. Preserve the argumentative flow:
 - Start with the major claim.
 - Present the main claims.
 - Include important premises that support those claims.
 - If present, include counterarguments and their rebuttals in a balanced, concise way.
4. Do NOT copy text verbatim from the structured outline; paraphrase naturally.
5. The output must be a single cohesive paragraph.
6. Do NOT include Claim numbers, labels, bullets, or structural markers.
7. Do NOT output the outline itself-only the final summary paragraph.

Now write the abstractive summary (80-120 words) using ONLY the structured outline below:

Prompt for LLM key point generation:

You are an expert in Key Point Analysis (KPA). Your task is to generate concise, reusable key points for a single topic.

Task:

Generate two sets of key points:

- 1) PRO key points – key reasons that SUPPORT the topic.
- 2) CON key points – key reasons that OPPOSE the topic.

Definition:

A key point (KP) is a short, general, reusable statement that summarizes a central idea expressed by many possible arguments on the topic.

Requirements for key points:

- Each key point must be a single, concise sentence (20 words).
- Key points must be general and reusable (no examples, evidence, anecdotes, or specific scenarios).
- Key points must be semantically distinct (no overlap or paraphrasing across KPs).
- PRO key points must support the topic; CON key points must oppose it.
- Produce 3-6 key points for each side (if the topic meaningfully allows both sides).

Output Format (STRICT JSON):

{

```

"pro_keypoints": [
  "<KP1>",
  "<KP2>",
  "<KP3>"
],
"con_keypoints": [
  "<KP1>",
  "<KP2>",
  "<KP3>"
]
}

```

Every key point MUST be a double-quoted string ("...").
If one side has no meaningful key points (rare), return an empty list for that side.

Now generate the key points for the topic:

Prompt for LLM key point matching:

You are an expert in Key Point Matching.

You will receive:

- a topic
- an argument (ADU)
- a list of pro key points
- a list of con key points

Your tasks:

1. Determine the stance of the argument toward the topic.
Output "pro" if the argument supports the topic.
Output "con" if the argument opposes the topic.
2. Match the argument to key points of the SAME stance.
For matching, use the INDEX of the key point in the provided list:
 - pro key points are indexed as PRO_0, PRO_1, PRO_2, ...
 - con key points are indexed as CON_0, CON_1, CON_2, ...
3. A key point matches if the argument expresses the same idea.

Output format (JSON):

```

{
  "stance": "pro" or "con",
  "matched_keypoints": ["PRO_0", "PRO_3"] // or ["CON_1"], or []
}

```

If the stance of the argument is 'pro', return indices such as ["PRO_1", ...].
If the stance of the argument is "con", return indices such as ["CON_1", ...].
If no key points match, return an empty list [].

Rules:

- Do NOT rewrite key points.
- Do NOT explain your reasoning.
- Only output the JSON object.

F.2 Prompt for LLM evaluation

Prompt for LLM structure evaluation:

You are evaluating an argumentative summary.

Below are:

- (1) A list of argument units from the original essay.
- (2) A list of summary sentences, in order.

Your task:

For EACH summary sentence, identify which argument units can be reasonably recovered from that sentence.

Important:

- Evaluate each sentence independently, even though you see the full summary.
- Allow abstraction and paraphrase.
- Do NOT infer content that is not clearly suggested by the sentence.
- If uncertain, do NOT include the argument unit.

Argument Units:

{argument_units}

Summary Sentences:

{summary_sentences}

Output format (strictly follow):

{output_format}

Prompt for span segmentation in faithfulness evaluation:

Split the following text into a Python-style list of contiguous text spans.

Each span must:

- Express exactly one coherent semantic idea or proposition.
- Be minimal: if a span contains more than one idea, split it.
- Be exhaustive: every part of the text must appear in exactly one span.
- Preserve the original wording, punctuation, and order exactly.
- Include background, framing, transitions, and conclusions – nothing should be omitted.

Important constraints:

- Do NOT identify, label, or classify arguments, claims, premises, rebuttals, or any roles.
- Do NOT omit, rewrite, summarize, or normalize any part of the text.
- If you are unsure whether two phrases express separate ideas, split them.
- Do not output spans that consist only of discourse markers or connectors without a main predicate.
- Treat causal, concessive, or contrastive clauses (e.g., because, as, although, while, but) as potential span boundaries when they introduce a new idea.

Example (illustration only):

Input:

"The device failed because the battery was empty, although the software worked."

Output:

```
[
  "The device failed",
  "because the battery was empty,",
  "although the software worked."
]
```

Output format:

- Return ONLY a Python-style list of strings.
- Example: ["span 1", "span 2", "..."]
- Do not include any text outside the list.

This is the text you need to split:

Prompt for LLM faithfulness evaluation:

You are performing structure-grounded faithfulness evaluation for argumentative summarization.

STRUCTURED SUMMARY (SOURCE):

Below is a numbered list of argument units that together define the intended argumentative content.

These units are the ONLY source of information you may use.

{structured_block}

SUMMARY SPANS:

Below is a numbered list of proposition-level summary spans extracted from an abstractive summary.

{span_block}

TASK:

For EACH summary span, identify which argument units support it.

Guidelines:

- A span is SUPPORTED if its content is stated, paraphrased, summarized, or reasonably inferred from ONE OR MORE argument units.
- Multiple argument units may jointly support a span.
- Do NOT use outside knowledge or assumptions.
- Do NOT infer beyond what the structured summary allows.
- If no argument units support a span, it is UNSUPPORTED.

OUTPUT FORMAT (JSON only):

Return one key-value pair per span.

The key must be the span ID (e.g., "S1").

The value must be a list of supporting argument unit numbers.

```
{{
  "S1": [list of argument unit numbers],
  "S2": [list of argument unit numbers]
}}
```

Notes:

- If a span is unsupported, its list MUST be empty.
- Do NOT add explanations, comments, or extra text.
- Output JSON only.

F.3 Prompts for prompting and fine-tuning experiments.

Summary prompt:

Write an abstractive summary (80 - 120 words) for the given essay.

Structure summary prompt:

Write an abstractive summary (80-120 words) of the following argumentative essay.

The summary should:

- Clearly state the author's main position.
- Preserve the argumentative flow of the essay by:
 - Starting with the major claim.
 - Presenting the main claims made in support of that position.
 - Including the key premises or evidence that support those claims.
 - If the essay presents an opposing view, briefly summarizing the counterarguments and how the author responds.
- Maintain the logical progression of the argument rather than listing points.
- Do not add new arguments or omit major argumentative components.

G Instructions for human evaluation and correction

Consent form:

****Consent Form****

Please read the following information carefully before continuing.

****Purpose of the Study****

You will correct AI-generated annotations for short essays.

Your responses will be used for research and dataset improvement.

****What data we collect:****

- Your corrected labels
- Your Prolific PID
- No personal identifying information is collected

****Voluntary Participation:****

You may withdraw at any time by closing the task window before submission.

****Risks/Benefits:****

Minimal risk. You may stop at any time.

By checking the box below, you confirm you understand the above information and consent to participate.

Instructions for ADU segmentation correction:

ADU Segmentation Correction Instructions

Task Overview

You will correct ADU (Argumentative Discourse Unit) boundaries in short essays.

The model has placed brackets ``< ... >`` around what it thinks are ADUs. Your task is to **fix the boundaries only** – do **not** change the essay text.

Each ADU must contain:

- Exactly **one** argumentative function (major claim, claim, premise, counterargument, or rebuttal)
- **Only** argumentative content (no scaffolding, no reporting frames, no structural sentences)

You are **NOT** labeling the type – you only adjust the bracket positions.

What **Is** an ADU?

An ADU is the **smallest argumentative unit**, containing exactly **one** argumentative idea.

A. Claims (opinions, positions, conclusions)

Examples:

- ``<modern buildings add variety and interest>``
- ``<people behave differently when wearing formal clothing>``
- ``<junk food taxes will reduce consumption>``

B. Premises (reasons, explanations, evidence)

Examples:

- ``<banning cars would reduce pollution>``
- ``<AI reduces accidents caused by human error>``
- ``<children today have fewer responsibilities than before>``

C. Counterarguments (opposing views)

Correct:

Some people believe that `<modern buildings damage cultural identity>`.

D. Rebuttals (responses to counterarguments)

Example:

- ``<this is unconvincing because tradition and modern design can coexist>``

What **Is NOT** an ADU?

The following **must NOT** be included inside ``< ... >``.

1. Rhetorical connectors

Examples that must stay **outside** ADUs:

- However,
- Therefore,
- On the other hand,
- For example,
- Moreover,
- As a result,
- Because (when used only to connect clauses)

Incorrect:

`<However, improving public transport is more effective>`

Correct:

However, `<improving public transport is more effective>`

2. Reporting frames

These introduce an argument but are **not** the argument.

Do NOT wrap:

- Many people believe that
- Some argue that
- Researchers claim that
- It is widely thought that

Incorrect:

`<Many people believe that banning cars is necessary>`

Correct:

Many people believe that `<banning cars is necessary>`

3. Structural / meta-discourse sentences

These describe the essay flow, not the argument.

Do NOT annotate:

- In this essay I will discuss...
- I will compare these issues...
- To begin with...
- As mentioned earlier...
- In conclusion...

Incorrect:

`<In this essay I will compare these issues and give my view>`

Correct:

In this essay I will compare these issues and give my view.

4. Background / contextual information

Background text introduces the topic or provides general context, but does NOT itself argue for or against a position.

Do NOT annotate background information as ADUs.

Examples of background text:

- General descriptions of a situation or trend

- Historical or factual setup without evaluative force
- Topic introductions that do not yet make a claim

Incorrect:

<Sophisticated constructions in the city center have become more common in recent decades>

Correct:

Sophisticated constructions in the city center have become more common in recent decades.

Correct when argumentative meaning is present:

<This trend damages the cultural identity of cities>

Splitting ADUs (when one sentence contains multiple argumentative ideas)

You must split **only** when there are two or more separate argumentative ideas, regardless of the connectors used.

Correct principle:

- If a sentence expresses **X** (an argumentative idea)
 and also expresses **Y** (another argumentative idea),
 then **X** and **Y** must be separate ADUs, no matter how the sentence is structured.

Examples (do NOT treat these as templates):

Example 1:

<<X expresses one argumentative idea>
because <Y expresses a different argumentative idea>

Example 2:

<X is a claim>
and <Y is a further justification>

Example 3:

<X is a counterargument>
but <Y is a rebuttal>

Example 4:

<X is a general claim>
which means <Y is a new inference>

Important:

These examples show the **principle**, not a rule about specific words like “because”, “and”, “but”, “which means”.

You must decide based on **whether X and Y express different argumentative ideas**, not based on whether these words appear.

Merging ADUs (when the model split one idea incorrectly)

You must merge ADUs when the content expresses **a single argumentative idea**, even if there are multiple clauses.

Correct principle:

- If X and Y together express **one idea**, keep them **in one ADU**.

Examples (again, do NOT treat as templates):

Example 1:

<X and Y together form one single claim>

Example 2:

<The writer gives one reason, described in two connected clauses>

Example 3:

<The entire span supports the same argumentative function>

Incorrect (model over-split):

<AI reduces mistakes> <and increases safety>

Correct (one idea):

<AI reduces mistakes and increases safety>

IMPORTANT WARNING FOR ANNOTATORS

These examples demonstrate concepts, not patterns.

Do NOT copy the formatting or assume that:

- “because” requires a split
- “and” requires a merge
- every clause is automatically an ADU
- every sentence is exactly one ADU

You must judge based on **argumentative meaning**, not connector words.

What You SHOULD Do

- Add missing ADUs
- Remove brackets around non-argumentative text
- Move brackets to the correct boundaries
- Split ADUs containing more than one argumentative idea
- Merge ADUs the model split incorrectly

What You MUST NOT Do

- Do NOT change the essay text (no edits, no paraphrasing)
- Do NOT add commentary or explanations
- Do NOT use tags other than `<` and `>`
- Do NOT wrap rhetorical or structural content
- Do NOT leave unmatched `<` or `>`

Final Reminder

- ONLY argumentative content goes inside `<` ... `>`.
- ONE ADU = ONE argumentative idea.
- A sentence may contain multiple ADUs.
- Some ADUs are shorter than a sentence.
- Adjust the brackets only – never modify the original text.

IMPORTANT: CONSERVATIVE CORRECTION & OPENING-SENTENCE POLICY

Default Principle

The model's ADU segmentation is generally **high quality**.
Your role is to **review the entire essay**, but to **edit only when clearly necessary**.

Step 1 – Special Handling of the First 1-2 Sentences

You must **carefully check the first 1-2 sentences of the essay**, as they are more likely to be ambiguous.

These opening sentences may be:

- **Pure background** → **do NOT annotate**, or
- **Contain argumentative content** (e.g. a claim, stance, or evaluative judgment)

If the opening sentence(s) require annotation, you **MUST** annotate them using the same ADU segmentation rules as the rest of the essay.

Do NOT automatically wrap the entire sentence as one ADU.

Instead:

- Identify the **exact argumentative clause(s)**
- Exclude background phrases, reporting frames, and rhetorical or structural content
- Split or merge **by argumentative meaning**, following the full ADU rules

Correct example:

`Some people argue that <modern buildings improve city life>`.

Incorrect example:

`<Some people argue that modern buildings improve city life.>`

Step 2 – Full Review with Conservative Editing (All Remaining Sentences)

You **MUST** read and check all sentences in the essay, including the body and the conclusion.

For **each sentence after the opening**:

1. **Verify** that the existing ADU boundaries are reasonable
2. **Decide** whether a correction is clearly required

You may **change ADU boundaries** only if there is a clear and confident violation of the ADU rules, such as:

- meta-discourse or structural text inside an ADU (e.g. "this essay will discuss...", "in conclusion...")
- reporting frames wrapped inside an ADU (e.g. "many people believe that...")
- one ADU clearly containing **more than one independent argumentative idea**

If none of these apply, **leave the ADU boundaries unchanged**.

What "leave unchanged" does NOT mean

- skipping or ignoring sentences
- assuming the rest is correct without checking
- stopping after the opening sentences

It means:

You reviewed the sentence and determined that no correction was necessary.

Step 3 – Rule for Uncertainty

If you are unsure whether a boundary is wrong, do NOT change it.

- Conservative decisions are preferred
- Over-editing reduces annotation quality

Final Reminder

- > **Review everything. Edit sparingly.**
- > Pay special attention to the opening sentences,
- > apply the **same ADU rules everywhere**,
- > and change boundaries **only when you are confident they are incorrect**.

Instructions for argument component classification correction:

Argument Component Classification Instructions

Task Overview

For each annotation, you will be given:

1. **The writing prompt** – this explains what the essay is responding to.
2. **A tagged essay**, where each argumentative unit is marked using tags such as `- 3. **The model's predicted labels**, shown in the following format:

```
```text
{
 '<AC0>': 'claim',
 '<AC1>': 'premise',
 '<AC2>': 'counterargument'
}
```
```

Your task is to review these labels and correct any that are wrong.
You only need to change the label words – keep the format exactly the same.

Allowed Labels (with explanations)
Use only the following five labels:

- major claim
 - The author's main overall conclusion or central stance of the entire essay that directly answers the essay prompt.

Important:

- An essay may contain at most TWO major claims.
- A major claim must not support any other claim.
- If a statement supports another point in the essay, it is a claim, not a major claim.
- If more than two statements seem equally central, choose the two most general ones.

- claim
 - A supporting assertion that directly backs up the major claim.

Claim test:

If the statement could be supported by other reasons in the essay,
then it is a **claim**, not a premise.

- premise
 - A reason, justification, or evidence that supports a claim.
It answers “why is this claim true?”

Premise test:

If the statement answers “Why should I believe this claim?”,
then it is a **premise**, not a claim.

- counterargument
 - A statement that opposes the author's position or presents an alternative viewpoint.

Counterarguments present an opposing view WITHOUT evaluating it.
If the sentence criticizes or weakens an opposing view, it is NOT a counterargument.

- rebuttal
 - A statement that responds to a counterargument and explains why it is weak or incorrect.

Rebuttal clarification:

A rebuttal MUST refer to or respond to an opposing view
(even if the counterargument appears earlier or is implied).

Quick Labeling Strategy (Follow This Order)

When assigning or correcting a label, decide in this order:

1. Is this the author's overall position about the essay question?
→ If yes, label it **major claim**.
2. Does this statement oppose the author's position?
→ If yes, label it **counterargument**.
3. Does this statement respond to or weaken a counterargument?
→ If yes, label it **rebuttal**.
4. Is this a position or conclusion that supports the major claim?
→ If yes, label it **claim**.
5. Otherwise, if it gives a reason, explanation, or evidence:
→ Label it **premise**.

What You Should Change

Only change the label on the RIGHT side, inside the single quotes.

Example correction:

```
```text
'<AC3>': 'premise' → '<AC3>': 'claim'
```
```

What You Must NOT Change

To keep the format valid:

- Do NOT change or remove the single quotes
- Do NOT change the keys ('<AC0>', '<AC1>', etc.)
- Do NOT add or delete commas, braces { }, or colons :
- Do NOT add explanations or extra text
- Do NOT edit the essay text or ADU tags

Your only task is to replace incorrect label words with correct ones.

Example

Model Output:

```
```text
{
 '<AC0>': 'counterargument',
 '<AC1>': 'premise',
 '<AC2>': 'claim'
}
```

---

### Corrected Version:

```text

```
{
  '<AC0>': 'claim',
  '<AC1>': 'premise',
  '<AC2>': 'major claim'
}
```

Only the labels changed – the structure stays exactly the same.

Final Reminder

Only edit the label inside the single quotes.
Do not change the formatting, keys, punctuation,
or structure.

You MUST use **only** the following five labels:

- major claim
- claim
- premise
- counterargument
- rebuttal

> **Important Note**

>

>

> When reviewing annotations:

> - Make corrections **only** if a label is clearly incorrect** according to the definitions.

> - **Keep** changes as minimal as possible.**

> - **If** you are unsure or the case is ambiguous, keep the original LLM output unchanged.**

>

> The purpose of human correction is to **fix** clear errors**, not to re-annotate or reinterpret the entire argument structure.

Instructions for structured summary evaluation and correction:

ANNOTATION INSTRUCTIONS: STRUCTURED ARGUMENT SUMMARY VERIFICATION & CORRECTION

1. TASK OVERVIEW

You will be shown:

- An essay where argumentative discourse units (ADUs) are marked using `- argument component labels for each ADU.
- A structured argument summary generated by an LLM.

Your job:

- Decide whether the structured summary is correct.
- If it needs correction, provide a corrected version.

2. REQUIRED OUTPUT FORMAT

The structured summary MUST follow exactly this format:

```text

Major claim:

<rewritten major claim>

Claim 1:

<rewritten claim>

Premise: <rewritten premise>

Premise: <rewritten premise>

...

Counterargument: <rewritten counterargument or None>

Rebuttal: <rewritten rebuttal>

...

Claim 2:

<rewritten claim>

Premise: <rewritten premise>

...

Counterargument: <...>

Rebuttal: <...>

...

---

...

#### FORMAT REQUIREMENTS:

- No AC tags or AC IDs.
- Major claim appears once at the top.
- Claims appear in essay order: Claim 1, Claim 2, Claim 3, ...
- Each ADU appears as exactly ONE rewritten unit.
- If a claim has no counterargument → "Counterargument: None"
- If a counterargument has no rebuttal → "Rebuttal: None"
- No invented content, no inferences, no new examples.
- No paragraphs, no bullet points, no commentary.

#### 3. DEFINITIONS OF ARGUMENT COMPONENTS

major claim:

The author's global thesis.

claim:

A local sub-claim expressing an opinion or conclusion.

premise:

A supporting reason, justification, explanation, or example.

counterargument:

An opposing view or objection (not the author's own stance).

rebuttal:

The author's response to a counterargument.

#### 4. VERIFICATION CHECKLIST (READ-ONLY)

Before deciding whether correction is needed, check mentally:

##### A. COMPLETENESS

- No ADUs missing or duplicated.
- No merged or split ADUs.

##### B. FAITHFULNESS

- Meaning and stance preserved.

- No hallucinated content.

#### C. CANONICAL STYLE

- Rewriting is concise and non-rhetorical.
- Each ADU appears as one unit.

#### D. STRUCTURAL CORRECTNESS

- Major claim correct.
- Claims in essay order.
- Premises under the correct claim.
- Counterarguments and rebuttals placed correctly.

#### E. FORMAT

- Exact required format followed.
- Correct use of "None".
- No AC tags or commentary.

You do NOT answer these individually – they guide your final decision.

#### 5. FINAL DECISION

After reviewing the structured summary:

Does this structured summary require correction?

- YES → You must provide a corrected version.
- NO → Approve the LLM output.

Only this one decision is recorded.

#### 6. CORRECTION GUIDELINES (IF NEEDED)

If correction IS required:

1. Fix errors in completeness, faithfulness, canonical style, structure, and formatting.
2. Rewrite only what is necessary.
3. Preserve the required structure exactly.
4. Ensure each ADU appears once and only once.
5. Do not add or infer new content.
6. Place corrected structured summary in the provided text box.

#### 7. WHEN CORRECTION \*IS\* REQUIRED

Examples:

- Wrong major claim.
- Premise placed under wrong claim.
- Missing counterargument or rebuttal.
- ADU omitted, duplicated, merged, or split.
- Hallucinated content present.
- Incorrect formatting or structure.

#### 8. WHEN CORRECTION IS \*NOT\* REQUIRED

Examples:

- All ADUs included once.
- Rewrites are concise and faithful.
- Claims and premises are grouped correctly.
- No hallucinations.
- Format matches the required outline.

Instructions for abstractive summary evaluation and correction:

#### # Annotator Instructions for Abstractive Summary Evaluation

Overview:

You will evaluate and, if needed, correct an AI-generated abstractive summary. Each example contains:

- (1) A writing prompt.
- (2) A structured argument outline.
- (3) A model-generated summary (80–120 words).

Your task:

- Step 1 – Evaluate the summary.
- Step 2 – Decide if it needs correction (Yes/No).
- Step 3 – If Yes, correct it with minimal edits.

#### STEP 1 – EVALUATE THE SUMMARY

Read the structured outline carefully.  
Read the model-generated summary.

Use the following criteria INTERNALLY to guide your judgment.

You DO NOT need to rate them individually.

##### [ ] Faithfulness:

- The summary must use ONLY information in the outline.
- No hallucinations or invented details.
- No new reasoning, examples, or interpretations.

##### [ ] Completeness:

- Includes the major claim.
- Includes the key supporting points.
- Includes counterarguments/rebuttals if appropriate.

##### [ ] Argument Flow:

- Follows the argumentative order:  
major claim → main reasons → counterarguments (if any).

##### [ ] Clarity & Coherence:

- Reads smoothly as one paragraph.
- No redundancy or confusing phrasing.

##### [ ] Length:

- Approximately 80–120 words.
- Small deviations allowed if clarity requires it.

#### STEP 2 – DECISION

Answer ONE question:

Does this summary need correction?

- YES
- NO

If NO:

→ Move to the next example.

If YES:

→ Proceed to Step 3.

### STEP 3 – CORRECTION (ONLY IF NEEDED)

When correction is required:

1. Make the SMALLEST number of edits needed to fix issues.
2. Do NOT rewrite the paragraph unless absolutely necessary.
3. Ensure the corrected summary:
  - Uses only information from the outline.
  - Accurately reflects the major claim and key reasons.
  - Maintains argumentative flow.
  - Is clear, concise, and coherent.
  - Contains no hallucinations.
  - Is one single paragraph.
  - Fits (or roughly fits) the 80–120 word range.

After revising, re-check:

- [ ] Faithful to the outline
- [ ] Complete (major claim + key support)
- [ ] Correct argument flow
- [ ] Clear and coherent
- [ ] No new information
- [ ] One paragraph
- [ ] Within desired length

Submit the corrected summary when all conditions are met.

### Instructions for key point generation evaluation and correction:

#### # Key Point Generation Evaluation & Correction Instructions

##### ## 0. Task Overview

You will be shown the following information on the annotation page:

- A **topic**.
- A set of **PRO** key points generated by the model.
- A set of **CON** key points generated by the model.

Your job is to:

1. **Evaluate** whether the model-generated key points meet the quality standards.
2. **Decide** whether the key points are acceptable as-is.

3. If NOT acceptable, **correct** the key points to meet the required quality.

You will evaluate the key points using the criteria below, BUT you do not need to answer each criterion separately.

You only answer one final question:  
**“Do the key points need correction?”** → YES / NO

If YES, you will **correct** the key points to meet the required quality.

---

#### ## 1. Evaluation Criteria

Use the following criteria to judge overall quality.

##### ### A. Generality

- Key points must be general and reusable.
- No examples, anecdotes, statistics, or specific scenarios.

##### ### B. Conciseness

- Each key point must be one sentence.
- Maximum 20 words.

##### ### C. Stance

- PRO key points express a supportive or positive meaning.
- CON key points express an opposing or negative meaning.

##### ### D. Non-Redundancy

- No two key points express the same idea.
- No paraphrasing or overlap.

##### ### E. Relevance

- Every key point must directly address the topic.

##### ### F. Completeness

- The set covers the major argument spaces for the topic.
- Typically 3–6 key points per side when possible.

---

#### ## 2. Final Decision (YES / NO)

After reviewing the criteria above, answer **one** single question:

**“Do the key points need correction?”** → YES / NO

##### ### YES – Needs Correction

- One or more criteria are not met.
- You must revise the key points.

##### ### NO – Does Not Need Correction

- All criteria pass.
- No edits needed.

---

### ## 3. How to Correct Key Points

Make the minimum necessary edits.

#### ### A. Fix Stance

- Move a key point to PRO or CON based on meaning.
- Rewrite if stance is unclear.

#### ### B. Fix Specificity

- Remove examples, specific facts, numbers, or anecdotes.
- Rewrite into a general argument template.

#### ### C. Fix Redundancy

- Remove or merge repetitive or overlapping key points.

#### ### D. Fix Clarity

- Rewrite unclear, vague, or awkward statements.

#### ### E. Fix Relevance

- Remove any key points that do not address the topic.
- Replace them if needed.

#### ### F. Fix Completeness

- Add missing major key points.
- Ensure a reasonable number of key points on each side.

---

### ## 4. Final Output Format

After correction (if needed), return the final key points in the following JSON structure:

```
```text
{
  "pro_keypoints": [
    "<KP1>",
    "<KP2>",
    "<KP3>"
  ],
  "con_keypoints": [
    "<KP1>",
    "<KP2>",
    "<KP3>"
  ]
}
```
```

Each key point must be a double-quoted string.

### Instructions for ADU segmentation adjudication:

#### # Introduction (For Adjudicators)

You are acting as an **adjudicator** for ADU (Argumentative Discourse Unit) segmentation.

For each essay, you will receive:

- The **writing prompt**
- The **original essay text**
- **Two independent human-corrected ADU segmentations**
- The **ADU Segmentation Correction Instructions**

#### - The **ADU Segmentation Adjudication Instructions**

Before adjudicating, you must **first read the ADU Segmentation Correction Instructions**, which define what counts as an ADU and how boundaries should be set. You should then read the **Adjudication Instructions**, which explain how to apply these rules to resolve disagreements.

Your task is to **resolve boundary disagreements between the two corrections** and produce **one final, rule-consistent ADU segmentation**. Adjudication enforces the existing correction guidelines and does **not** introduce new rules or interpretations.

#### # ADU Segmentation Correction Instructions

##### ## Task Overview

You will correct ADU (Argumentative Discourse Unit) boundaries in short essays.

The model has placed brackets `< ... >` around what it thinks are ADUs.

Your task is to **fix the boundaries only** – do **not** change the essay text.

Each ADU must contain:

- Exactly **one** argumentative function (major claim, claim, premise, counterargument, or rebuttal)
- **Only** argumentative content (no scaffolding, no reporting frames, no structural sentences)

You are **NOT** labeling the type – you only adjust the bracket positions.

---

##### # What **Is** an ADU?

An ADU is the **smallest argumentative unit**, containing exactly **one** argumentative idea.

#### ### A. Claims (opinions, positions, conclusions)

Examples:

- `<modern buildings add variety and interest>`
- `<people behave differently when wearing formal clothing>`
- `<junk food taxes will reduce consumption>`

#### ### B. Premises (reasons, explanations, evidence)

Examples:

- `<banning cars would reduce pollution>`
- `<AI reduces accidents caused by human error>`
- `<children today have fewer responsibilities than before>`

#### ### C. Counterarguments (opposing views)

Correct:

---

Some people believe that `<modern buildings damage cultural identity>`.

```

D. Rebuttals (responses to counterarguments)
Example:
- <this is unconvincing because tradition and
modern design can coexist>

What Is *NOT* an ADU?

The following **must NOT** be included inside <
... >`.

1. Rhetorical connectors
Examples that must stay **outside** ADUs:
- However,
- Therefore,
- On the other hand,
- For example,
- Moreover,
- As a result,
- Because (when used only to connect clauses)

Incorrect:
```
<However, improving public transport is more
effective>
```

Correct:
```
However, <improving public transport is more
effective>
```

2. Reporting frames
These introduce an argument but are **not** the
argument.

Do NOT wrap:
- Many people believe that
- Some argue that
- Researchers claim that
- It is widely thought that

Incorrect:
```
<Many people believe that banning cars is
necessary>
```

Correct:
```
Many people believe that <banning cars is
necessary>
```

3. Structural / meta-discourse sentences
These describe the essay flow, not the argument.

Do NOT annotate:
- In this essay I will discuss...
- I will compare these issues...
- To begin with...
- As mentioned earlier...

```

```

- In conclusion...

Incorrect:
```
<In this essay I will compare these issues and
give my view>
```

Correct:
```
In this essay I will compare these issues and
give my view.
```

4. Background / contextual information

Background text introduces the topic or provides
general context,
but does NOT itself argue for or against a
position.

Do NOT annotate background information as ADUs.

Examples of background text:
- General descriptions of a situation or trend
- Historical or factual setup without evaluative
force
- Topic introductions that do not yet make a
claim

Incorrect:
<Sophisticated constructions in the city center
have become more common in recent decades>

Correct:
Sophisticated constructions in the city center
have become more common in recent decades.

Correct when argumentative meaning is present:
<This trend damages the cultural identity of
cities>

Splitting ADUs (when one sentence contains
multiple argumentative ideas)

You must split **only when there are two or more
separate argumentative ideas**, regardless of
the connectors used.

Correct principle:
- If a sentence expresses **X** (an argumentative
idea)
and also expresses **Y** (another
argumentative idea),
then **X and Y must be separate ADUs**, no
matter how the sentence is structured.

Examples (do NOT treat these as templates):

Example 1:
```
<X expresses one argumentative idea>
because <Y expresses a different argumentative
idea>
```

```

Example 2:  
 <X is a claim>  
 and <Y is a further justification>

Example 3:  
 <X is a counterargument>  
 but <Y is a rebuttal>

Example 4:  
 <X is a general claim>  
 which means <Y is a new inference>

**Important:**  
 These examples show the *principle*, not a rule about specific words like “because”, “and”, “but”, “which means”.  
 You must decide based on *whether X and Y express different argumentative ideas*, not based on whether these words appear.

---  
**## Merging ADUs (when the model split one idea incorrectly)**  
 You must merge ADUs when the content expresses *a single argumentative idea*, even if there are multiple clauses.

**Correct principle:**  
 - If X and Y together express *one idea*, keep them *in one ADU*.

Examples (again, do NOT treat as templates):

Example 1:  
 <X and Y together form one single claim>

Example 2:  
 <The writer gives one reason, described in two connected clauses>

Example 3:  
 <The entire span supports the same argumentative function>

**Incorrect (model over-split):**  
 <AI reduces mistakes> <and increases safety>

**Correct (one idea):**  
 <AI reduces mistakes and increases safety>

---

**### IMPORTANT WARNING FOR ANNOTATORS**  
**These examples demonstrate concepts, not patterns.**  
 Do NOT copy the formatting or assume that:  
 - “because” requires a split  
 - “and” requires a merge  
 - every clause is automatically an ADU  
 - every sentence is exactly one ADU

You must judge based on *argumentative meaning*, not connector words.

---

**# What You SHOULD Do**  
 - Add missing ADUs  
 - Remove brackets around non-argumentative text  
 - Move brackets to the correct boundaries  
 - Split ADUs containing more than one argumentative idea  
 - Merge ADUs the model split incorrectly

---

**# What You MUST NOT Do**  
 - Do NOT change the essay text (no edits, no paraphrasing)  
 - Do NOT add commentary or explanations  
 - Do NOT use tags other than `<` and `>`  
 - Do NOT wrap rhetorical or structural content  
 - Do NOT leave unmatched `<` or `>`

---

**# Final Reminder**  
 - ONLY argumentative content goes inside `<` ... `>`  
 - ONE ADU = ONE argumentative idea.  
 - A sentence may contain multiple ADUs.  
 - Some ADUs are shorter than a sentence.  
 - Adjust the brackets only – never modify the original text.

**# ADU Segmentation Adjudication Instructions**

**### Task Overview**  
 As an adjudicator, your task is to produce *one final ADU segmentation* by resolving disagreements between *two human-corrected versions* of the same essay.

You will be given:  
 - The *writing prompt*  
 - The *original essay text*  
 - *Two independent human ADU corrections*  
 - The *ADU Segmentation Correction Instructions*

Your adjudication must strictly *follow the correction instructions*.  
 Adjudication does *not* introduce new rules or interpretations.

---

**### Your Role**  
 Your role is to:  
 - Compare the two corrected versions

- Resolve **boundary disagreements only**
- Apply the correction guidelines consistently
- Produce a **single, rule-compliant ADU segmentation**

You are **not** evaluating annotators or the model, and you are **not** labeling ADU types.

---

### ### What You Should Do

You may:

- Adopt Annotator A's boundary decision for a specific ADU
- Adopt Annotator B's boundary decision for a specific ADU
- Modify boundaries if **both are incorrect**
- Add missing ADUs
- Remove ADUs that include non-argumentative text
- Split or merge ADUs according to the rules

All decisions must be justified by the **ADU Segmentation Correction Instructions**.

---

### ### What You Must NOT Do

You must NOT:

- Change, rewrite, or paraphrase the essay text
- Add or remove content
- Assign argumentative labels (e.g. claim, premise)
- Add comments or explanations
- Use tags other than `<` and `>`
- Let the writing prompt override the segmentation rules

---

### ### Decision Principles

When resolving disagreements:

1. **Rule compliance first**  
Choose the version that best follows the correction instructions.
2. **Argumentative meaning over form**  
Decide based on whether a span expresses **one** or **multiple** argumentative ideas, not sentence structure or connectors.
3. **Minimal units**  
One ADU must contain **exactly one** argumentative idea.
4. **Independence**  
If both annotators are wrong, correct the boundaries yourself.

---

### ### Final Output Requirements

Your final output must:

- Contain only the original essay text

- Include correct `<` `>` brackets
- Have no unmatched or overlapping brackets
- Follow all correction rules

The final adjudicated version will be treated as the **gold ADU segmentation**.

---

### ### Final Reminder

Adjudication is a **rule-based consolidation step**.

Apply the ADU Segmentation Correction Instructions **consistently and strictly** when resolving disagreements.

### Instructions for argument component classification adjudication:

#### # Introduction (For Adjudicators)

You are acting as an adjudicator for Argument Component Classification (ACC).

For each annotation, you will receive:

- The writing prompt
- The tagged essay containing ADUs (e.g., `- Two independently corrected label sets (Annotator A and Annotator B)
- The ACC label definitions and decision rules

Before starting, you must:

- Carefully read the **ACC Correction Instructions** (which define the labels and how they should be applied)
- Carefully read the **ACC Adjudication Instructions** (which explain how to resolve disagreements)
- Understand the definitions of the five allowed labels
- Understand the decision rules for assigning labels

#### # ACC Correction Instructions

#### ## Task Overview

For each annotation, you will be given:

1. **The writing prompt** – this explains what the essay is responding to.
2. **A tagged essay**, where each argumentative unit is marked using tags such as `- 3. **The model's predicted labels**, shown in the following format:

```
```text
{
  '<AC0>': 'claim',
  '<AC1>': 'premise',
  '<AC2>': 'counterargument'
}
```

Your task is to review these labels and correct any that are wrong.
You only need to change the label words – keep the format exactly the same.

Allowed Labels (with explanations)
Use only the following five labels:

- major claim
 - The author's main overall conclusion or central stance of the entire essay that directly answers the essay prompt.

Important:

- An essay may contain at most TWO major claims.
- A major claim must not support any other claim.
- If a statement supports another point in the essay, it is a claim, not a major claim.
- If more than two statements seem equally central, choose the two most general ones.

- claim
 - A supporting assertion that directly backs up the major claim.

Claim test:

If the statement could be supported by other reasons in the essay,
then it is a **claim**, not a premise.

- premise
 - A reason, justification, or evidence that supports a claim.
It answers “why is this claim true?”

Premise test:

If the statement answers “Why should I believe this claim?”,
then it is a **premise**, not a claim.

- counterargument
 - A statement that opposes the author's position or presents an alternative viewpoint.

Counterarguments present an opposing view WITHOUT evaluating it.
If the sentence criticizes or weakens an opposing view, it is NOT a counterargument.

- rebuttal
 - A statement that responds to a counterargument and explains why it is weak or incorrect.

Rebuttal clarification:

A rebuttal MUST refer to or respond to an opposing view
(even if the counterargument appears earlier or is implied).

Quick Labeling Strategy (Follow This Order)

When assigning or correcting a label, decide in this order:

1. Is this the author's overall position about the essay question?
→ If yes, label it **major claim**.
2. Does this statement oppose the author's position?
→ If yes, label it **counterargument**.
3. Does this statement respond to or weaken a counterargument?
→ If yes, label it **rebuttal**.
4. Is this a position or conclusion that supports the major claim?
→ If yes, label it **claim**.
5. Otherwise, if it gives a reason, explanation, or evidence:
→ Label it **premise**.

What You Should Change

Only change the label on the RIGHT side, inside the single quotes.

Example correction:

```
```text
'<AC3>': 'premise' → '<AC3>': 'claim'
```
```

What You Must NOT Change

To keep the format valid:

- Do NOT change or remove the single quotes
- Do NOT change the keys ('<AC0>', '<AC1>', etc.)
- Do NOT add or delete commas, braces { }, or colons :
- Do NOT add explanations or extra text
- Do NOT edit the essay text or ADU tags

Your only task is to replace incorrect label words with correct ones.

Example

Model Output:

```
```text
{
 '<AC0>': 'counterargument',
 '<AC1>': 'premise',
 '<AC2>': 'claim'
}
```
```

Corrected Version:

```
```text
{
 '<AC0>': 'claim',
 '<AC1>': 'premise',
 '<AC2>': 'major claim'
}
```
```

Only the labels changed – the structure stays exactly the same.

Final Reminder
Only edit the label inside the single quotes.
Do not change the formatting, keys, punctuation,
or structure.

You MUST use **only** the following five labels:

- major claim
- claim
- premise
- counterargument
- rebuttal

> **Important Note**

>

> When reviewing annotations:

> - Make corrections **only** if a label is clearly incorrect** according to the definitions.

> - **Keep changes as minimal as possible.**

> - **If you are unsure or the case is ambiguous, keep the original LLM output unchanged.**

>

> The purpose of human correction is to **fix** clear errors**, not to re-annotate or reinterpret the entire argument structure.

Argument Component Classification (ACC)
Adjudication Instructions

Task Overview

Your task is to:

Resolve disagreements between Annotator A and Annotator B
and produce one final, rule-consistent label set

You are NOT re-annotating from scratch.
You are consolidating two corrected versions into a single gold annotation.

Allowed Labels

You must use **ONLY** the following five labels:

- major claim
- claim
- premise
- counterargument
- rebuttal

Label Definitions (Concise)

- major claim
The author's overall position answering the essay prompt
At most TWO per essay

- claim
A statement that supports the major claim

- premise
A reason or evidence supporting a claim
(answers: Why should I believe this?)

- counterargument
A statement that opposes the author's position

- rebuttal
A response that weakens or rejects a counterargument

Decision Order (Follow Strictly)

When deciding a label, use this order:

1. Is this the author's overall position?
→ major claim
2. Does this oppose the author's position?
→ counterargument
3. Does this respond to an opposing view?
→ rebuttal
4. Does this support the major claim?
→ claim
5. Otherwise (reason or evidence)
→ premise

Input Format

Tagged Essay:

<AC0>...</AC0>

<AC1>...</AC1>

Annotator A:

```
{  
  '<AC0>': 'claim',  
  '<AC1>': 'premise'  
}
```

Annotator B:

```
{  
  '<AC0>': 'premise',  
  '<AC1>': 'premise'  
}
```

What You Should Do

For each ADU:

1. If Annotator A and Annotator B agree AND the label follows the rules
→ Keep the label

2. If they disagree
→ Choose the label that best follows the ACC rules and definitions

3. If both labels are incorrect (whether they agree or disagree)
→ Assign the correct label yourself

Global Consistency Rules

Ensure the final annotation satisfies:

- At most TWO major claims
- Rebuttal must respond to a counterargument
- Premise = reason, not a standalone claim
- Labels form a coherent argument structure

What You MUST NOT Do

- Do NOT change AC IDs (`<AC0>`, etc.)
- Do NOT change formatting, punctuation, or structure
- Do NOT edit the essay or ADU tags
- Do NOT add explanations or comments
- Do NOT introduce new labels

Decision Principles

1. Rule compliance first
Always prioritize the formal definitions and rules.
2. Minimal intervention
Prefer Annotator A or B when one is clearly correct
Only override both if necessary
3. Meaning over form
Base decisions on argumentative role, not wording or position
4. Consistency across essay
Keep labeling coherent and logically structured

Output Format (Strict)

```
...  
{  
  '<AC0>': '...',  
  '<AC1>': '...',  
  '<AC2>': '...',  
}
```

Only modify label values
Do NOT change anything else

Final Reminder

- Resolve disagreements only
- Follow the decision rules strictly

- Maintain minimal necessary changes
- Produce a clean, consistent final annotation
- Do NOT include any explanations

Instructions for structured summary adjudication:

ANNOTATION INSTRUCTIONS: STRUCTURED ARGUMENT SUMMARY ADJUDICATION

1. ROLE AND GOAL

You are the adjudicator.

You will be given:

- An essay where argumentative discourse units (ADUs) are marked using `<<AC0>...</AC0>`, `<<AC1>...</AC1>`, etc.
- Argument component labels for each ADU.
- Two structured argument summaries:
 - Annotation A
 - Annotation B

Your goal:

- Produce the best possible structured argument summary by selecting, combining, and correcting A and B.

2. KEY PRINCIPLE

You are NOT restricted to choosing a single annotation.

You may:

- choose Annotation A
- choose Annotation B
- combine A and B
- modify or correct either (or both)

Your goal is to produce the most accurate, complete, and well-structured summary.

3. DEFINITIONS OF ARGUMENT COMPONENTS, QUALITY GUIDELINES

major claim:

The author's global thesis.

claim:

A local sub-claim expressing an opinion or conclusion.

premise:

A supporting reason, justification, explanation, or example.

counterargument:

An opposing view or objection (not the author's own stance).

rebuttal:

The author's response to a counterargument.

Use the following checklist to guide your judgment when comparing and improving A and B.

A. COMPLETENESS

- No ADUs missing or duplicated.
- No merged or split ADUs.

B. FAITHFULNESS

- Meaning and stance preserved.
- No hallucinated content.

C. CANONICAL STYLE

- Rewriting is concise and non-rhetorical.
- Each ADU appears as one unit.

D. STRUCTURAL CORRECTNESS

- Major claim correct.
- Claims in essay order.
- Premises under the correct claim.
- Counterarguments and rebuttals placed correctly.

E. FORMAT

- Exact required format followed.
- Correct use of "None".
- No AC tags or commentary.

These criteria are guidelines for improving the output.
Minor issues can be lightly adjusted, while clear errors should be corrected.

4. ADJUDICATION STRATEGY

Step 1 – Compare A and B:

- Identify which parts are correct in A
- Identify which parts are correct in B
- Identify any missing, incorrect, or misplaced elements

Step 2 – Construct the best version:

You may:

- take correct parts from A
- take correct parts from B
- merge complementary information
- fix errors where necessary

Step 3 – Minimise unnecessary changes:

- Only modify what improves correctness
- Do not rewrite everything from scratch unless necessary

5. REQUIRED OUTPUT FORMAT

Your output must be ONLY the final structured summary.

Do NOT include:

- explanations
- comments
- labels (e.g., A, B, Combined)
- JSON or metadata

The output MUST follow this format exactly:

```text

Major claim:

<rewritten major claim>

Claim 1:

<rewritten claim>

Premise: <text>

Premise: <text>

...

Counterargument: <text or None>

Rebuttal: <text or None>

Claim 2:

...

FORMAT REQUIREMENTS:

- No AC tags or AC IDs.
- Major claim appears once at the top.
- Claims appear in essay order: Claim 1, Claim 2, Claim 3, ...
- Each ADU is represented once as a single rewritten unit.
- If a claim has no counterargument → "Counterargument: None"
- If a counterargument has no rebuttal → "Rebuttal: None"
- No invented content, no inferences, no new examples.
- No paragraphs, no bullet points, no commentary.

### 6. IMPORTANT RULES

- Use only information from the provided ADUs.
- Do not invent or infer new arguments.
- Ensure structure is clear, consistent, and complete.
- Prefer combining and refining A and B rather than discarding both.

FINAL SUMMARY:

Produce a single clean structured argument summary by selecting, combining, and minimally correcting Annotations A and B, following the required format exactly.

Instructions for abstractive summary adjudication:

ADJUDICATION INSTRUCTIONS FOR ARGUMENTATIVE SUMMARIES

Overview:

You are acting as a third annotator (adjudicator).

Your task is to compare two independently produced summaries and produce a final corrected version.

Each example contains:

- (1) A writing prompt
- (2) A structured argument outline (the only source of truth)
- (3) Annotation 1 (a summary)
- (4) Annotation 2 (a summary)

Your goal is to produce a single adjudicated summary that is correct, complete, and faithful to the outline.

## STEP 1 – ANALYZE BOTH ANNOTATIONS (INTERNALLY)

Read the structured outline carefully.  
Then read Annotation 1 and Annotation 2.

Evaluate both summaries using the following criteria:

- [ ] Faithfulness:
  - The summary must use ONLY information in the outline.
  - No hallucinations, no added examples, no new reasoning.
- [ ] Completeness:
  - Includes the major claim.
  - Includes the main supporting points.
  - Includes counterarguments/rebuttals if present.
- [ ] Alignment to Outline:
  - Every statement must be traceable to the outline.
  - No missing key information.
  - No incorrect merging or splitting of ideas.
- [ ] Argument Flow:
  - Follows a logical structure:  
major claim → supporting arguments → counterarguments (if any).
- [ ] Clarity & Coherence:
  - Reads as a smooth, single paragraph.
  - No redundancy or confusing phrasing.

## STEP 2 – COMPARE AND IDENTIFY ISSUES

Decide which annotation is more accurate and identify problems:

- Typical issues include:
- Missing important points
  - Hallucinated or unsupported content
  - Incorrect emphasis (wrong major claim)
  - Poor structure or ordering
  - Lack of clarity

Do NOT output this analysis – use it to guide your decision.

## STEP 3 – PRODUCE THE ADJUDICATED SUMMARY

Create ONE final corrected summary that improves upon both annotations.

Follow these rules:

1. Use both annotations when possible:
  - Combine correct parts from each.
  - Do NOT blindly select one annotation.
2. Fix all problems:
  - Remove hallucinations.
  - Add missing key points from the outline.

- Correct structure and ordering.

3. Stay strictly grounded:
  - Use only information from the outline.
  - Do NOT introduce new ideas, examples, or interpretations.
4. Balance minimal editing and correctness:
  - Reuse phrasing when correct.
  - Rewrite only when necessary for accuracy or clarity.

## FINAL OUTPUT REQUIREMENTS

Your final summary MUST:

- Be ONE single paragraph
- Be approximately 80–120 words
- Include:
  - the major claim
  - key supporting arguments
  - counterarguments/rebuttals if present
- Follow a clear argumentative flow
- Contain NO hallucinated content
- Be clear, concise, and coherent

## FINAL CHECKLIST (MANDATORY)

Before submitting, ensure:

- [ ] All content is supported by the outline
- [ ] Major claim is correctly represented
- [ ] Key arguments are included
- [ ] No important points are missing
- [ ] No hallucinated or invented information
- [ ] Logical argument flow is preserved
- [ ] The summary is one coherent paragraph
- [ ] Length is roughly within 80–120 words

Important:

Your role is to resolve disagreements and produce the most accurate summary – do not rewrite freely, and do not introduce new content.

Instructions for key point generation adjudication:

# Key Point Generation \*\*Adjudication  
Instructions\*\*

## 0. Task Overview

You are acting as an \*\*adjudicator\*\*.

For each topic, you will be shown:

- The \*\*topic\*\*
- The \*\*model-generated PRO and CON key points\*\*
- \*\*Two independently human-corrected versions\*\*:
  - \*\*Correction A\*\*
  - \*\*Correction B\*\*

Your task is to \*\*produce one final adjudicated version\*\* of the key points.

---

## ## 1. Adjudication Goal

Your goal is to produce a **single, high-quality set of PRO and CON key points** that best satisfies the criteria.

You may:

- Select **one correction as-is**
- Make **minimal edits** to one correction
- **Merge** the best elements of both corrections

---

## ## 2. Quality Criteria

The final adjudicated key points must satisfy all of the following:

### ### A. Generality

- Key points must be general and reusable.
- No examples, anecdotes, statistics, or specific cases.

### ### B. Conciseness

- Each key point must be **one sentence**.
- Maximum **20 words**.

### ### C. Stance

- **PRO** key points express a supportive or positive meaning.
- **CON** key points express an opposing or negative meaning.

### ### D. Non-Redundancy

- No duplicate or overlapping ideas.
- No paraphrasing of the same argument.

### ### E. Relevance

- Every key point must directly address the topic.

### ### F. Completeness

- The set reasonably covers the major argument spaces.
- Typically **3-6 key points per side**, when possible.

You do **not** need to explicitly score each criterion, but your final output must satisfy all of them.

---

## ## 3. Adjudication Procedure

Follow these steps:

### ### Step 1: Compare Corrections A and B

Evaluate which correction better satisfies the criteria overall, paying special attention to:

- Redundancy
- Missing major arguments
- Overly specific phrasing
- Incorrect or unclear stance

### ### Step 2: Choose an Adjudication Strategy

Select exactly **one** of the following approaches:

#### #### Option A: Select One Correction As-Is

- Use this when one correction clearly meets all criteria.
- No changes are required.

#### #### Option B: Minor Edit to One Correction

- Apply small fixes such as:
  - Rewording for clarity or generality
  - Removing redundancy
  - Adding or removing **at most one** key point
  - Correcting PRO/CON placement

#### #### Option C: Merge the Two Corrections

- Use this when each correction contains valid but complementary key points.
- Merge carefully to avoid redundancy.
- Do not exceed a reasonable number of key points.

---

## ## 4. Editing Principles

- Make the **minimum necessary changes**
- Do **not** introduce new ideas unless required for completeness
- Prefer **general, neutral, and abstract phrasing**
- When in doubt:
  - Prefer **generality over specificity**
  - Prefer **clarity over coverage**

---

## ## 5. Common Mistakes to Avoid

- Creating key points that express the same idea
- Allowing vague, empty, or generic statements
- Mixing or weakening PRO/CON stance
- Over-editing or rewriting everything unnecessarily

---

## ## 6. Final Output Format

Return **only** the final adjudicated key points in the following JSON format:

```
```text
{
  "pro_keypoints": [
    "<KP1>",
    "<KP2>",
    "<KP3>"
  ],
  "con_keypoints": [
    "<KP1>",
    "<KP2>",
    "<KP3>"
  ]
}
```
```

Each key point must be a double-quoted string.

## H Dataset example

An example in our dataset:

```
{
 "index": 0,
 "topic": "Some people say that modern buildings are ugly and are ruining our towns and cities, especially when these buildings are very different from those around them. Others say that they add variety and interest. Discuss both these views and give your own opinion.",
 "essay": "Sophisticated constructions in the city center is not merely give worth-looking for the society. Numerous people argue that these virtually showing poor appearance toward that city, but others convince that a dozens of sacrifice building it could be make that region livelier. In this essay I will compare these issue and give my view.\n\nWith regard to, globalization literally has been escorting people to build up several monumental buildings with kinds of design. However, it is not mean that creating new establishment also replaced original form of that region. Many people believe that impressive architecture owed by one region would damage cultural character of that area. Consequently, local communities could lost their old houses as a symbol of that region. As an example, in Bali, Indonesia, the governments proscribe their society to build up structures no more than four storey because they believe that their prayer building (Pura) should be higher than others construction.\n\nOn the other hand, some people regard as remarkable building located in inner city will attract the occupant attention because of the unique appearance. Furthermore, it is also can be used as tourist's attraction because in a big city poorly beautiful place. Besides, the increasing number of modern building in one town is also symbolize of their prosperous citizen live there.\n\nAll in all, there are convincing argument both the presence of modern buildings in the central city, I would argue that cutting-edge design of building belonged to the cities, it cannot be restricted by traditionalist understanding, but it would be better if it mixed to construct the new buildings."
}
```

```
"essay_tagged": "Sophisticated constructions in the city center is not merely give worth-looking for the society. Numerous people argue that <AC0>these virtually showing poor appearance toward that city</AC0>, but others convince that <AC1>a dozens of sacrifice building it could be make that region livelier</AC1>. In this essay I will compare these issue and give my view.\n\nWith regard to, <AC2>globalization literally has been escorting people to build up several monumental buildings with kinds of design</AC2>. However, <AC3>it is not mean that creating new establishment also replaced original form of that region</AC3>. Many people believe that <AC4>impressive architecture owed by one region would damage cultural character of that area</AC4>. Consequently, <AC5>local communities could lost their old houses as a symbol of that region</AC5>. As an example, <AC6>in Bali, Indonesia, the governments proscribe their society to build up structures no more than four storey</AC6> because <AC7>they believe that their prayer building (Pura) should be higher than others construction</AC7>.\n\nOn the other hand, some people regard as <AC8>remarkable building located in inner city will attract the occupant attention because of the unique appearance</AC8>. Furthermore, <AC9>it is also can be used as tourist's attraction because in a big city poorly beautiful place</AC9>. Besides, <AC10>the increasing number of modern building in one town is also symbolize of their prosperous citizen live there</AC10>.\n\nAll in all, there are convincing argument both the presence of modern buildings in the central city, I would argue that <AC11>cutting-edge design of building belonged to the cities, it cannot be restricted by traditionalist understanding</AC11>, but <AC12>it would be better if it mixed to construct the new buildings</AC12>.",
 "argument_components": {
 "<AC0>": "counterargument",
 "<AC1>": "counterargument",
 "<AC2>": "claim",
 "<AC3>": "rebuttal",
 "<AC4>": "claim",
 "<AC5>": "premise",
 "<AC6>": "premise",
 "<AC7>": "premise",
 "<AC8>": "claim",
 "<AC9>": "premise",
 "<AC10>": "premise",
 "<AC11>": "major claim",
 "<AC12>": "claim"
 },
}
```

```

"structured_summary": "Major
claim:\nModern buildings in cities should
not be restricted by traditionalist
views, but should instead be integrated
with them.\n\nClaim 1:\nGlobalization
has led to the construction of many
monumental buildings with diverse
designs.\nPremise: None\nCounterargument:
Modern buildings give cities a poor
appearance.\nCounterargument:
Sacrificing buildings can make a region
livelier.\nRebuttal: Creating new
establishments does not necessarily
replace the original form of a
region.\n\nClaim 2:\nImpressive
architecture in a region may damage its
cultural character.\nPremise: Local
communities may lose their old houses as
symbols of the region.\nPremise: In Bali,
Indonesia, the government prohibits
buildings taller than four
stories.\nPremise: Religious buildings
(Pura) should remain taller than other
constructions.\nCounterargument:
None\nRebuttal: None\n\nClaim
3:\nRemarkable buildings in city centers
attract people because of their unique
appearance.\nPremise: Beautiful places
in large cities can become tourist
attractions.\nPremise: Modern buildings
symbolize prosperous
citizens.\nCounterargument:
None\nRebuttal: None\n\nClaim 4:\nIt
would be better to mix traditional and
modern elements when constructing new
buildings.\nPremise:
None\nCounterargument: None\nRebuttal:
None",
"abstractive_summary": "Modern buildings
should not be restricted by
traditionalist views but thoughtfully
integrated with them. Globalization has
led to the construction of diverse,
monumental designs that can invigorate
cities, but some argue these give a poor
appearance, and impressive architecture
may damage cultural character -as
communities lose old houses that
symbolize the region. In Bali, for
instance, height restrictions ensure
religious structures remain the tallest.
However, creating new establishments
does not necessarily replace a region's
original form. Remarkable buildings in
city centers attract tourists and
symbolize prosperity. Rather than
choosing between old and new, it is
better to mix traditional and modern
elements when constructing new
buildings.",
"key_points": {
 "pro_keypoints": [
 "Modern buildings add visual
 variety and break urban
 monotony.",
 "Innovative designs reflect
 contemporary culture and
 technological progress.",
 "Architectural diversity can
 stimulate interest and inspire
 creativity."
],
 "con_keypoints": [
 "Modern buildings often clash
 with traditional architectural
 styles.",
 "Unsympathetic designs can
 disrupt the aesthetic harmony of
 historic areas.",
 "Excessive modernization may
 erode local character and
 cultural identity."
]
},
"argument_kps": [
 {
 "argument_text": "these virtually
 showing poor appearance toward
 that city",
 "stance": "con",
 "matched_kps": [
 "Unsympathetic designs can
 disrupt the aesthetic
 harmony of historic areas."
]
 },
 {
 "argument_text": "a dozens of
 sacrifice building it could be
 make that region livelier",
 "stance": "pro",
 "matched_kps": [
 "Architectural diversity can
 stimulate interest and
 inspire creativity."
]
 },
 {
 "argument_text": "globalization
 literally has been escorting
 people to build up several
 monumental buildings with kinds
 of design",
 "stance": "pro",
 "matched_kps": [
 "Innovative designs reflect
 contemporary culture and
 technological progress."
]
 },
 {
 "argument_text": "it is not mean
 that creating new establishment
 also replaced original form of
 that region",
 "stance": "con",
 "matched_kps": [
 "Excessive modernization may
 erode local character and
 cultural identity."
]
 }
],
"argument_text": "impressive
architecture owed by one region
would damage cultural character
of that area",
"stance": "con",

```

```

 "matched_kps": [
 "Excessive modernization may
 erode local character and
 cultural identity."
]
 },
 {
 "argument_text": "local
 communities could lost their old
 houses as a symbol of that
 region",
 "stance": "con",
 "matched_kps": [
 "Excessive modernization may
 erode local character and
 cultural identity."
]
 },
 {
 "argument_text": "in Bali,
 Indonesia, the governments
 proscribe their society to build
 up structures no more than four
 storey",
 "stance": "con",
 "matched_kps": [
 "Excessive modernization may
 erode local character and
 cultural identity."
]
 },
 {
 "argument_text": "they believe
 that their prayer building (Pura)
 should be higher than others
 construction",
 "stance": "con",
 "matched_kps": [
 "Unsympathetic designs can
 disrupt the aesthetic
 harmony of historic areas."
]
 },
 {
 "argument_text": "remarkable
 building located in inner city
 will attract the occupant
 attention because of the unique
 appearance",
 "stance": "pro",
 "matched_kps": [
 "Modern buildings add visual
 variety and break urban
 monotony.",
 "Architectural diversity can
 stimulate interest and
 inspire creativity."
]
 },
 {
 "argument_text": "it is also can
 be used as tourist's attraction
 because in a big city poorly
 beautiful place",
 "stance": "pro",
 "matched_kps": [
 "Modern buildings add visual
 variety and break urban
 monotony.",
 "Architectural diversity can
 stimulate interest and
 inspire creativity."
]
 }
],
{
 "argument_text": "the increasing
 number of modern building in one
 town is also symbolize of their
 prosperous citizen live there",
 "stance": "pro",
 "matched_kps": [
 "Innovative designs reflect
 contemporary culture and
 technological progress."
]
},
{
 "argument_text": "cutting-edge
 design of building belonged to
 the cities, it cannot be
 restricted by traditionalist
 understanding",
 "stance": "pro",
 "matched_kps": [
 "Innovative designs reflect
 contemporary culture and
 technological progress.",
 "Architectural diversity can
 stimulate interest and
 inspire creativity."
]
},
{
 "argument_text": "it would be
 better if it mixed to construct
 the new buildings",
 "stance": "con",
 "matched_kps": [
 "Unsympathetic designs can
 disrupt the aesthetic
 harmony of historic areas."
]
}
]
}
}

```