Argument Mining as a Text-to-Text Generation Task

Masayuki Kawarada[1], **Tsutomu Hirao**[2], **Wataru Uchida**[1] and **Masaaki Nagata**[2]
[1]Service Innovation Department, NTT DOCOMO, INC., Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Japan
{masayuki.kawarada.vw, uchidaw}@nttdocomo.com
{tsutomu.hirao, masaaki.nagata}@ntt.com

Abstract

Argument Mining (AM) aims to uncover the argumentative structures within a text. Previous methods require several subtasks, such as span identification, component classification, and relation classification. Consequently, these methods need rule-based postprocessing to derive argumentative structures from the output of each subtask. This approach adds to the complexity of the model and expands the search space of the hyperparameters. To address this difficulty, we propose a simple yet strong method based on a text-to-text generation approach using a pretrained encoder-decoder language model. Our method simultaneously generates argumentatively annotated text for spans, components, and relations, eliminating the need for task-specific postprocessing and hyperparameter tuning. Furthermore, because it is a straightforward text-to-text generation method, we can easily adapt our approach to various types of argumentative structures. Experimental results demonstrate the effectiveness of our method, as it achieves state-of-the-art performance on three different types of benchmark datasets: the Argument-annotated Essays Corpus (AAEC), AbstRCT, and the Cornell eRulemaking Corpus (CDCP).

1 Introduction

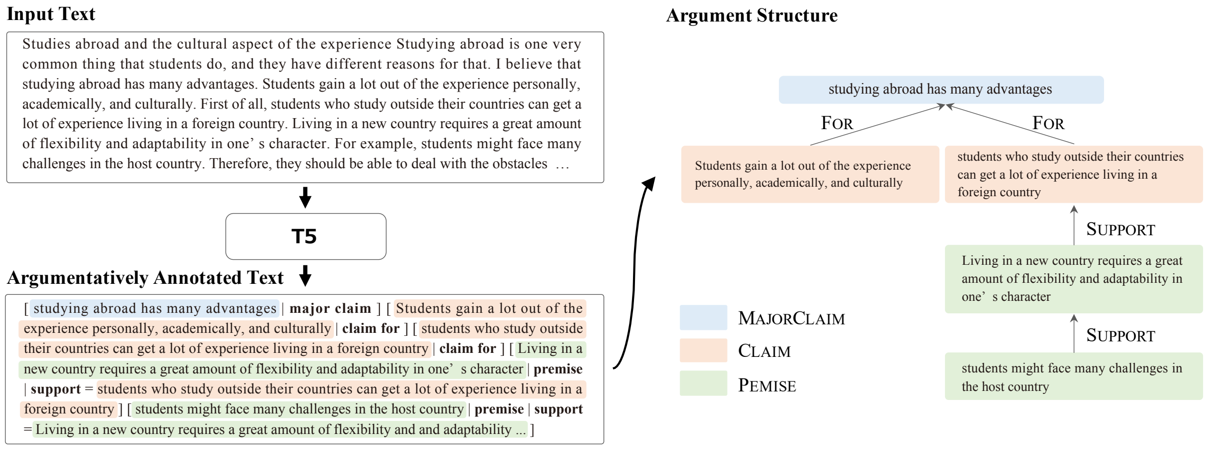
Argument Mining (AM) is a form of discourse analysis that seeks to identify the structure of an argument within a text (Lawrence and Reed, 2019). This structure is typically represented through a dependency tree or a directed acyclic graph, as shown on the right-hand side of Figure 1. In the dependency tree, nodes correspond to text spans that contain arguments, which are then classified into specific argument types. The edges between the nodes represent the relations between the arguments.

Annotated corpora have been constructed for argument mining to reveal the argumentative structure across various fields, such as student essays (Stab and Gurevych, 2014, 2017), biomedical research (Mayer et al., 2020), and more. These corpora serve as the standard benchmark datasets, allowing for the performance evaluation of argument mining systems. Given its practical applications in downstream tasks, such as text summarization (Fabbri et al., 2021; Elaraby and Litman, 2022) and automatic essay scoring (Nguyen and Litman, 2018), argument mining has recently gained significant attention in discourse analysis.

Neural models enhanced the performance of argument mining, along with other natural language processing tasks. Early models employed a pipeline approach involving three subtasks: identifying the argumentative text span, determining the argument type, and establishing the relation between the two arguments (Stab and Gurevych, 2017; Niculae et al., 2017). However, recent models treat argument mining as dependency parsing and perform it in an end-to-end manner (Ye and Teufel, 2021; Morio et al., 2022). These models are complex as they require separate mechanisms for each of the three tasks to be incorporated into the model. Therefore, postprocessing is necessary to build valid dependency trees. Furthermore, hyperparameter tuning poses difficulties in implementing these models.

To tackle these difficulties, we exploited a simple text-to-text generation model with the Translation between Augmented Natural Languages (TANL) (Paolini et al., 2021), which has achieved state-of-the-art performance on sentence-level structured prediction tasks such as relation extraction, named entity recognition, and semantic role labeling. Implementing TANL into AM, we offer significant advantages: (1) a simple architecture that eliminates complex postprocessing and hyperparameter tuning, (2) the ability to adapt to various annotations based on the dataset, and (3) the potential to use recent large language models.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2002–2014
March 17-22, 2024 ©2024 Association for Computational Linguistics



Input Text
T5
Argumentatively Annotated Text
Argument Structure
studying abroad has many advantages
For
For
Students gain a lot out of the experience personally, academically, and culturally
students who study outside their countries can get a lot of experience living in a foreign country
Support
Living in a new country requires a great amount of flexibility and adaptability in one’ s character
Support
students might face many challenges in the host country
MajorClaim
Claim
Pemise

Studies abroad and the cultural aspect of the experience Studying abroad is one very common thing that students do, and they have different reasons for that. I believe that studying abroad has many advantages. Students gain a lot out of the experience personally, academically, and culturally. First of all, students who study outside their countries can get a lot of experience living in a foreign country. Living in a new country requires a great amount of flexibility and adaptability in one’ s character. For example, students might face many challenges in the host country. Therefore, they should be able to deal with the obstacles …

[studying abroad has many advantages | **major claim**] [Students gain a lot out of the experience personally, academically, and culturally | **claim for**] [students who study outside their countries can get a lot of experience living in a foreign country | **claim for**] [Living in a new country requires a great amount of flexibility and adaptability in one’ s character | **premise** | **support** = students who study outside their countries can get a lot of experience living in a foreign country] [students might face many challenges in the host country | **premise** | **support** = Living in a new country requires a great amount of flexibility and and adaptability ...]

Figure 1: Overview of our methods. For our methodology, we input text into a pretrained encoder-decoder, such as T5 and FLAN T5. This process generates an argumentatively annotated text with spans, components, and relations. We then postprocess the output text to extract the argumentative structure.

Experimental results from three benchmark datasets, Argument-annotated Essays Corpus (AAEC) (Stab and Gurevych, 2017), AbstRCT (Mayer et al., 2020) and Cornell eRulemaking Corpus (CDCP) (Park and Cardie, 2018), demonstrate that our method achieved the state-of-the-art scores on both Component-F1 and Relation-F1 when using FLAN T5 (Chung et al., 2022):XXL (11B). Furthermore, by preventing the model from generating irrelevant text spans, which cannot be arguments, we successfully reduced the computational time for inference in AbstRCT by 30% without compromising performance.

2 Related Work

2.1 Argument Mining

AM involves three critical subtasks: identifying the arguments within a text, determining their argument type, and establishing the relations between these arguments. These steps are crucially required in revealing the argumentative structure of a text. Earlier methods used a pipeline architecture, where argumentative span identification was performed first, followed by component classification[1] and relation classifications (Persing and Ng, 2016; Eger et al., 2017; Kuribayashi et al., 2019; Morio et al., 2020). However, such an approach can result in the accumulation of errors from previous subtasks.

To improve the pipeline-based approach, recent studies employ an end-to-end method (Morio et al., 2022; Bao et al., 2022; Ye and Teufel, 2021; Eger et al., 2017). Ye and Teufel (2021) and Morio et al. (2022) used a network architecture based on a biaffine parser, which achieved state-of-the-art performance on paragraph and essay level evaluation. This approach treats the argumentative structure as a dependency tree and uses a dependency parsing algorithm to parse them. Despite being end-to-end models, they often require hand-crafted rules (Eger et al., 2017; Ye and Teufel, 2021) or an optimum branching algorithm (Morio et al., 2022) to form dependency trees from the outputs of three layers corresponding to subtasks. Additionally, these models present challenges in tuning hyperparameters such as the learning rate, given the embedding of three subtasks within a network.

By contrast, Bao et al. (2022) use an encoder-decoder model to perform AM as a generation task. They employ a constraint pointer-mechanism (CPM) for BART (Lewis et al., 2020) to predict the index of words in the input text. However, our work differs from theirs as we focus on the text-to-text generation task. This enables us to maximize the use of the decoder without making any modifications to the pretrained language model.

2.2 Information Extraction as a Generation Task

The recent development of pretrained language models (Raffel et al., 2020; Lewis et al., 2020) has led researchers to tackle information extraction tasks such as relation extraction (Huguet Cabot and Navigli, 2021; Lu et al., 2022) and event extraction (Li et al., 2021; Lu et al., 2021) as generation

[1] An argumentative text span assigned with a label, such as Claim or Premise, is called a component.

	AAEC	AbstRCT	CDCP
# component	6,089	3,279	4,931
# relation	3,832	2,060	1,220
# components with multiple parents	0	31	160
% words in nonargumentative span[2]	28.09	49.30	0

Table 1: Statistics of AAEC, AbstRCT, CDCP.

tasks. Nayak and Ng (2020) compared two models for the relation extraction task: copy mechanism-based decoding and text-to-text generation. However, the results did not conclusively determine which method is superior.

Translation between Augmented Natural Languages (TANL) (Paolini et al., 2021) extends Nayak and Ng (2020)'s approach to text-to-text generation. This methodology has proved highly effective for tasks such as relation extraction, named entity recognition, semantic role labeling, and coreference resolution. This success can be attributed to the implementation of a more powerful pretrained encoder-decoder model, T5. In a recent study, Hu and Wan (2023) proposed using T5 for sentence-level RST parsing as a form of text-to-text generation, demonstrating its effectiveness in analyzing sentence structures. Their research motivates us to view document structure analysis, specifically argument mining, as more of a text generation task than a conventional natural language understanding task. To address this text-to-text generation task, we propose using T5 within the TANL framework, which could be a significant solution.

3 Proposed Methods

Figure 1 provides an overview of our approach based on TANL (Paolini et al., 2021). To obtain the argumentatively annotated text for TANL, we align the original text with the given argumentative spans, their types, and relations. Then, we fine-tune T5 with the TANL framework using the annotated texts.

3.1 Task Formalization

Given an input text x consisting of n words, it can be represented as $x = [x_1, \ldots, x_n]$. The objective of span identification is to extract spans $s = [x_{\text{start}}, \ldots, x_{\text{end}}]$ that include the argument. Here, start and end indicate the indices marking the beginning and end of the span, respectively. Such extracted spans are denoted as $(\mathrm{start}, \mathrm{end})$. Component classification labels the identified spans with component labels c from a set C. Set C contains all component labels present in the dataset. As a result of the classification, components are represented as $(\mathrm{start}, \mathrm{end}, c)$. Relation classification involves selecting source and target spans from the extracted spans and assigning a relation label r from a predefined set R. Set R holds all relation labels in the dataset. The source and target spans are expressed as $(\mathrm{start}^{\mathrm{src}}, \mathrm{end}^{\mathrm{src}})$ and $(\mathrm{start}^{\mathrm{tgt}}, \mathrm{end}^{\mathrm{tgt}})$ respectively. Consequently, the output from relation classification can be illustrated as $(\mathrm{start}^{\mathrm{src}}, \mathrm{end}^{\mathrm{src}}, \mathrm{start}^{\mathrm{tgt}}, \mathrm{end}^{\mathrm{tgt}}, \mathrm{r})$.

3.2 Argumentatively Annotated Text

We adapt the output format of TANL's joint entity and relation extraction task to AM. When a text span s^{src} with a specific component label c depends on another text span s^{tgt} by a relation label r, we represent it as "[s^{src} | c | $r = s^{\mathrm{tgt}}$]". However, if the span s^{src} does not depend on others, we omit s^{tgt} and the relation label r, denoting it as "[s^{src} | c]". Below is an example illustrating the application of TANL's method to AM:

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

3.3 Elimination of Unnecessary Text Spans

TANL attempts to annotate textual structure while maintaining integrity of the original input text. However, our focus moves to tasks that require documents as input, in contrast to the original TANL's requirement for sentence-level inputs. Reducing the maximum number of tokens in the encoder and decoder models is crucial for efficient computing. As a result, we exclude nonargumentative spans from TANL's annotation scheme. Table 2 shows examples of annotation with nonargumentative spans and without nonargumentative spans.

[2] We used SpaCy 3.6.1 as a tokenizer for word counts.

Input Text	Advantages and disadvantages of the prevalent of English With the development of globalization , English became the dominated language in national trade , conference and many important events . This phenomenon has aroused a heated discussion in public . Some people claim that the prevalent of English brings a great number of benefits for people .
w/ nonargumentative span	Advantages and disadvantages of the prevalent of English With the development of globalization , English became the dominated language in national trade , conference and many important events . This phenomenon has aroused a heated discussion in public . Some people claim that [the prevalent of English brings a great number of benefits for people | claim for] .
w/o nonargumentative span	[the prevalent of English brings a great number of benefits for people | claim for]

Table 2: Example of input text and output in TANL and our format. The table shows that our output format reduces the number of tokens compared to the TANL format by removing tokens that do not contain any components or relations.

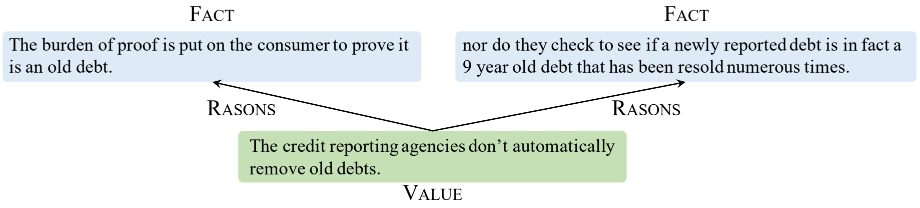
Argumentative Structure	 FACT The burden of proof is put on the consumer to prove it is an old debt. FACT nor do they check to see if a newly reported debt is in fact a 9 year old debt that has been resold numerous times. REASONS REASONS The credit reporting agencies don't automatically remove old debts. VALUE
repeated representation	[The credit reporting agencies don't automatically remove old debts . | value | reasons = nor do they check to see if a newly reported debt is in fact a 9 year old debt that has been resold numerous times .] [The credit reporting agencies don't automatically remove old debts . | value | reasons = The burden of proof is put on the consumer to prove it is an old debt .]
serial representation	[The credit reporting agencies don't automatically remove old debts . | value | reasons = nor do they check to see if a newly reported debt is in fact a 9 year old debt that has been resold numerous times . | value | reasons = The burden of proof is put on the consumer to prove it is an old debt .]

Table 3: Examples of repeated representation and serial representation in CDCP.

3.4 Representation of Components with Multiple Parents

The AM dataset contains components that depend on multiple parents. The output format of TANL's joint entity and relation extraction task cannot represent such structures in text, as it only adds annotations to the input text without repetition or deletion. To address components with multiple parents, we employ two representations: **repeated representation** and **serial representation**[3]. These examples are illustrated in Table 3. As an illustration, a component with VALUE labeled "*The credit reporting agencies don't automatically remove old debts.*" depends on two components labeled REASONS. The repeated representation treats these as two separate relations and represents them sequentially, while the serial representation represents the first relation followed by the second.

4 Experiments

4.1 Dataset

We used three major benchmark datasets: Argument-annotated Essay Corpus (AAEC) (Stab and Gurevych, 2017) , AbstRCT (Mayer et al., 2020) and the Cornell eRulemaking Corpus (CDCP) (Park and Cardie, 2018).

AAEC includes annotations of components and relations for essays written by students. It contains two types of data: essay-level and paragraph-level. At the essay level, AM is performed on the entire essay as input, whereas at the paragraph level, AM is performed on the predefined paragraphs. AAEC provides three component labels $C = \{$MAJORCLAIM, CLAIM, PREMISE$\}$ and four relation labels $R = \{$FOR, AGAINST, SUPPORT, ATTACK$\}$. According to the AAEC annotation guidelines, a CLAIM is always dependent on a MAJORCLAIM. Therefore, in our experiments, we adapted the labels to include four component labels $C = \{$MAJORCLAIM, CLAIMFOR, CLAIMAGAINST, PREMISE$\}$ and

[3] Serial representation is based on TANL's nested entities and multiple relations output format

two relation labels $R = \{\textsc{Support}, \textsc{Attack}\}$. During evaluation, we treat CLAIMFOR and CLAIMAGAINST as equivalent to CLAIM and evaluate them as in the previous studies. As shown in Table 1, the ratio of words in the nonargumentative span to the total words is 28.09%. It is worth noting that a component does not have multiple parents.

AbstRCT is obtained from Randomized Controlled Trials (RCT) from the MEDLINE for various diseases, such as neoplasm, glaucoma, hepatitis, diabetes, and hypertension. It features three component labels $C = \{\textsc{MajorClaim}, \textsc{Claim}, \textsc{Evidence}\}$ and three relation labels $R = \{\textsc{Support}, \textsc{Attack}, \textsc{Partial-Attack}\}$. As indicated in Table 1, this dataset contains a significant proportion of nonargumentative spans. Words in the nonargumentative spans account for 49.30% of the total word count.

CDCP is annotated with components and relations for comments from citizens. It provides five component labels $C = \{\textsc{Fact}, \textsc{Testimony}, \textsc{Value}, \textsc{Policy}, \textsc{Reference}\}$ and two relation labels $R = \{\textsc{Reasons}, \textsc{Evidence}\}$. As shown in Table 1, the CDCP does not contain any nonargumentative spans. Furthermore, we observed that CDCP contains a greater number of components that depend on multiple other components than the other two datasets.

4.2 Fine-tuning T5 with QLoRA

In previous studies, TANL performed fine-tuning on the T5-Base by updating all parameters, a process known as *full fine-tuning*. However, in this study, we aim to explore the effects of additional parameters. To this end, we employ QLoRA (Dettmers et al., 2023) tuning to reduce GPU memory during the training of large parameter models such as T5-XL (3B) and T5-XXL (11B). QLoRA is an adaptor that quantizes the model and applies Low-Rank Adapters (LoRA) (Hu et al., 2022) to it, helping reduce the number of parameters to be trained while maintaining performance levels comparable to full fine-tuning.

4.3 Settings

For the AAEC, we follow the train/dev/test split suggested by Eger et al. (2017). The number of essays for each dataset is 286, 36, and 80, while the number of paragraphs is 1587, 199, and 449, respectively. For the AbstRCT, we used the neoplasm test set and adopted the split provided in the original paper (Park and Cardie, 2018), using 300 for training, 50 for dev, and 100 for testing. Following (Niculae et al., 2017), we preserved 150 comments as a test set from the 731 comments within the CDCP. For the CDCP, 15% of training data were extracted as a dev set.

We examined T5 (Raffel et al., 2020) and FLAN-T5 (Wei et al., 2022) as pretrained encoder-decoder models used within the TANL framework. Our experiments were conducted with four different parameter models: Base (220M), Large (770M), XL (3B), and XXL (11B). In each experimental setting, we report the average scores from three runs using different seeds.

4.4 Compared Models

We compared our method with the following models, including the state-of-the-art model:

- **ILP** (Stab and Gurevych, 2017): A feature-based method that employs Integer Linear Programming (ILP) to parse each subtask in a pipelined fashion.

- **BLCC** (Eger et al., 2017): A method that treats Argument Mining (AM) as a sequence tagging problem.

- **LSTM-ER** (Eger et al., 2017): An end-to-end relation extraction model that uses LSTM-ER (Miwa and Bansal, 2016), which combines tree structure with sequential LSTM models.

- **BiPAM-syn** (Huang et al., 2021): A model that employs BERT as a language model for end-to-end dependency parsing, incorporating biaffine operations and syntactic information.

- **BART-CPM** (Bao et al., 2022): An encoder-decoder model similar to ours, employing BART (Lewis et al., 2020) as the language model and the Constrained Pointer Mechanism (CPM).

- **Single Task (ST) model** (Morio et al., 2022): A state-of-the-art model in AM, employing a biaffine neural approach akin to BiPAM-syn, yet using Longformer (Beltagy et al., 2020) as the language model.

		Essay		Paragraph	
	Params	Component	Relation	Component	Relation
ILP (Stab and Gurevych, 2017)	-	-	-	62.61	34.74
BLCC (Eger et al., 2017)	-	63.23	34.82	66.69	39.83
LSTM-ER (Eger et al., 2017)	-	66.21	29.56	70.83	45.52
BiPAM-syn (Ye and Teufel, 2021)	110M	-	-	73.5	46.4
BART-CPM (Bao et al., 2022)	139M	-	-	75.94	50.08
ST Model (Morio et al., 2022)	149M	76.55	54.66	76.48	59.55
T5-Base	220M	73.75	49.69	74.85	57.16
T5-Large	770M	75.65	51.17	75.55	57.47
T5-3B	3B	77.95	55.95	77.43	59.53
T5-11B	11B	79.48	57.06	77.17	59.02
FLAN T5-Base	220M	75.17	51.99	75.55	58.51
FLAN T5-Large	770M	77.75	56.06	76.93	58.57
FLAN T5-XL	3B	78.51	56.80	77.89	60.94
FLAN T5-XXL	11B	**80.15**	**61.19**	**78.40**	**61.87**

Table 4: Evaluation results at both the essay and paragraph levels obtained from AAEC. “Params” indicates the model parameters of the pretrained language model used by each comparison model. **Bold** indicates the highest F1 score for each task.

	C	R
ST Model (Morio et al., 2022)	64.16	38.38
FLAN T5-Base	68.76	38.31
FLAN T5-Large	71.11	44.47
FLAN T5-XL	71.27	45.80
FLAN T5-XXL	**72.86**	**47.66**

Table 5: Evaluation results for Component-F1 (C) and Relation-F1 (R) in AbstRCT. **Bold** denotes the highest F1 score for each task.

	C	R
BART-CPM (Bao et al., 2022)	57.72	16.57
ST Model (Morio et al., 2022)	68.90	31.94
FLAN T5-Base	66.80	23.19
FLAN T5-Large	68.94	28.42
FLAN T5-XL	72.12	31.01
FLAN T5-XXL	**72.68**	**33.96**

Table 6: Evaluation results for Component-F1 (C) and Relation-F1 (R) in CDCP. **Bold** denotes the highest F1 score for each task.

4.5 Evaluation Measures

Our methods were evaluated using the Component-F1 score and the Relation-F1 score[4], which are considered the de-facto standard evaluation metrics in the field (Stab and Gurevych, 2017; Eger et al., 2017; Ye and Teufel, 2021; Morio et al., 2022; Bao et al., 2022). Unlike AAEC, many studies have evaluated the component classification and the relation classification tasks of AbstRCT and CDCP given an oracle span (Kuribayashi et al., 2019; Morio et al., 2020; Mayer et al., 2020). For the evaluation of AbstRCT and CDCP, we benchmarked our scores against those from prior research that also measured Component-F1 and Relation-F1 scores in an end-to-end fashion.

4.6 Implementation Details

We trained the model using a batch size of 32 for AAEC at the paragraph level, and a batch size of 8 for AAEC at the essay level, as well as for AbstRCT and CDCP. We set the maximum token length to 512, for AAEC at the paragraph level, and 1,024 for the other datasets. The learning rate for both the Base and Large models was set at 0.0005, while a learning rate of 0.0002 was used for the XL (3B) and XXL (11B) models. All training took place on a single A100 (80GB) GPU over 10,000 steps, with checkpoints every 200 steps.

Typically, encoder-decoder models may not accurately replicate the input text, resulting in identified text spans that differ from those in the original text. To mitigate this issue, TANL uses the Needleman-Wunsch alignment algorithm (Needleman and Wunsch, 1970) to establish alignment between the output and input text spans. This process determines the position of words in the input text within the output text, an approach that we also adopted.

Following the method of Bao et al. (2022), we performed inference using the development set with the fine-tuned model, selecting the best checkpoint based on the average scores of Component-F1

[4] For the AAEC evaluation, we used the scripts of Eger et al. (2017).

		AAEC (Essay)		AAEC (Paragraph)		AbstRCT	
Output Format	**Model**	**Component**	**Relation**	**Component**	**Relation**	**Component**	**Relation**
w/ nonargumentative span	FLAN-T5 Base	74.99	50.87	74.97	57.54	65.30	34.55
	FLAN T5-Large	77.76	55.62	76.53	59.09	69.47	39.66
	FLAN T5-XL	78.73	57.21	77.17	61.03	73.13	42.39
	FLAN T5-XXL	**80.59**	60.37	**79.06**	**62.38**	72.78	47.11
w/o nonargumentative span	FLAN-T5 Base	75.17	51.99	75.55	58.51	68.76	38.31
	FLAN T5-Large	77.75	56.06	76.93	58.57	71.11	41.49
	FLAN T5-XL	78.51	56.80	77.89	60.94	71.27	45.80
	FLAN T5-XXL	80.15	**61.19**	78.40	61.87	**72.86**	**47.66**

Table 7: Comparison of Component-F1 and Relation-F1 scores with and without nonargumentative span output for essay-level and paragraph-level tasks in AAEC and AbstRCT.

		Full Dataset		Multiple Parents	
Output Format	**Model**	**Component**	**Relation**	**Component**	**Relation**
repeated representation	FLAN-T5 Base	66.94	22.40	60.12	22.64
	FLAN T5-Large	66.80	23.19	64.15	27.43
	FLAN T5-XL	72.12	31.01	68.67	32.81
	FLAN T5-XXL	**72.68**	33.96	**69.97**	35.26
serial representation	FLAN-T5 Base	67.11	23.64	63.17	23.81
	FLAN T5-Large	67.57	30.36	65.18	33.93
	FLAN T5-XL	70.86	32.98	66.95	33.66
	FLAN T5-XXL	71.34	**34.96**	68.53	**40.14**

Table 8: Comparison of Component-F1 (C) and Relation-F1 (R) with different representations in CDCP. **Full dataset** shows results using all CDCP data, while **Multiple Parent** shows results using only data with multiple parents.

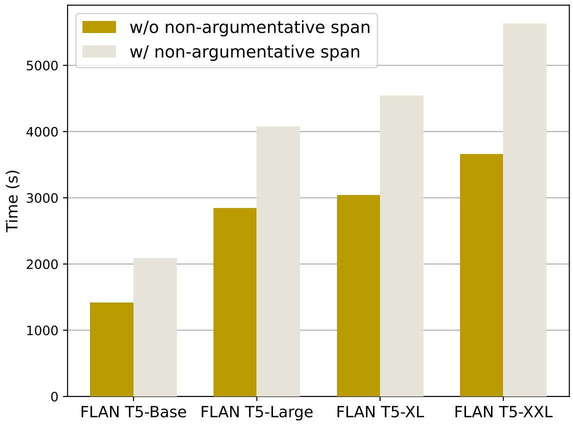


Figure 2: Comparison of inference time with and without nonargumentative spans in AbstRCT. We set the batch size to 2 for all models during inference and measured the time required to complete the process on the entire test dataset.

and Relation-F1. For training with QLoRA, we applied 4-bit quantization to the model and set the training hyperparameters as $r = 16$ and $\alpha = 32$. In addition, we trained the model by integrating the adapter into all linear layers. Further details of our implementation can be found in Appendix A.

5 Results and Discussion

5.1 Main Results

Table 4 shows the results obtained from the AAEC. The results suggest that an increase in parameters tends to yield significant performance gains. Although our models do not outperform state-of-the-art models when employing base or Large models, their performance is notably enhanced using models with billion-scale parameters. Our model with 3B parameters (T5-XL and FLAN-T5-XL) achieved F1 scores comparable to current state-of-the-art models, while those with 11B parameters exceeded the existing top F1 scores. Specifically, our FLAN T5-XXL model yielded Component-F1 scores of 80.15 at the essay level and 78.40 at the paragraph level, and Relation-F1 scores of 61.19 and 61.87, respectively.

Despite the larger parameter count, our straightforward model architecture using QLoRA proves practical. The table also reveals that FLAN T5 outperforms T5, indicating that instruction-tuning on various tasks using FLAN (Wei et al., 2022) has a positive impact on the AM task.

We also present the results obtained from AbstRCT and CDCP in Table 5 and Table 6, respectively. For AbstRCT, our FLAN T5-XXL achieved state-of-the-art performance with a Component-F1 score of 72.86 and Relation-F1 score of 47.66. Similarly, the CDCP results also demonstrate state-of-the-art performance, with Component-F1 and Relation-F1 scores of 72.68 and 33.96, respectively. Our model significantly outperforms existing models in both datasets, reinforcing the effectiveness of our approach: text-to-text generation with the TANL framework.

5.2 Difficulties of Hyperparameter Tuning

Our method offers an advantage over previous methods owing to the simplicity of our model. Previous methods have individual hyperparameters for subtasks, including span identification, component classification, and relation classification. Morio et al. (2022) employed Optuna (Akiba et al., 2019) to determine the optimal hyperparameters. However, hyperparameter tuning can become challenging owing to the interdependence of the three subtasks. This complexity becomes more pronounced when using large models, potentially complicating the training process.

By contrast, as our method relies solely on the learning rate as its hyperparameter, it avoids the difficulties associated with hyperparameter tuning that are inherent in previous methods. This simplicity is particularly advantageous when employing large-scale language models.

5.3 Eliminating Nonargumentative Spans

Our model does not output irrelevant text spans that cannot be classified as arguments. To assess the impact of this elimination, we conducted evaluations of the model without eliminating the following three tasks[5]: AAEC at the essay level, AAEC at the paragraph level, and AbstRCT.

Table 7 shows the results. Across all three tasks, it is evident that the performance is not degraded by the eliminations. Furthermore, inference time can be reduced by eliminating irrelevant spans, an effect that is particularly pronounced in AbstRCT owing to its high count of nonargumentative spans. Figure 2 illustrates the run time of the inference on AbstRCT. The figure clearly shows the effectiveness of the eliminations. Our method reduces the inference time by approximately 30% for all models: Base, Large, XL, and XXL.

5.4 Comparison of Component Representations with Multiple Parents

Table 8 presents a comparison of two different representations, repeated and serial, across the complete test dataset from the CDCP (Full Dataset). The table also includes the Component-F1 and Relation-F1 scores, calculated only for components with multiple parents (Multiple Parents). The data demonstrates that there is no significant difference in performance between the two representations in the component classification. However, in relation classification, the serial representation consistently outperforms the other models across all models. The results suggest that the repeated representation may not be optimal for tasks requiring the extraction of long-term relationships, such as relation classification. This could be attributed to the model's difficulty in capturing the full scope of relationships in the text, as the repeated representation breaks down one-to-many relationships into several separate one-to-one relations.

5.5 Performance Improvement with Increasing Model Parameters

In the AAEC task at the essay-level task, FLAN T5-XXL showed a significant improvement in Relation-F1 performance compared to FLAN T5-XL (Table 4). The gain is surprisingly around 4.4 points. To investigate the results in more detail, we discuss the F1 scores for each Component and Relation label. Figure 3 shows the results. For the component label, we observed a significant improvement in CLAIM (66.12 vs. 71.57) compared to MAJORCLAIM (79.34 vs. 82.96) and PREMISE (82.87 vs. 83.01). When focusing on relation labels, we found the largest improvement is seen in the MAJORCLAIM-CLAIM relation (45.33 vs.57.14). According to the AAEC annotation rules, PREMISE to CLAIM and PREMISE to PREMISE are classified as relations within a paragraph, while CLAIM to MAJORCLAIM can be a relation spanning different paragraphs. Therefore, these results imply that FLAN T5-XXL can capture a longer dependency between arguments. The substantial contribution of a large number of parameters to the improvement of distant dependency detection has a significant impact on the argument mining research community.

Table 9 shows example outputs for gold, FLAN T5-XL, and FLAN T5-XXL. In the table, FLAN T5-XL incorrectly predicts "*the advertising ex-*

[5] We excluded CDCP from this evaluation as it does not contain irrelevant text spans.

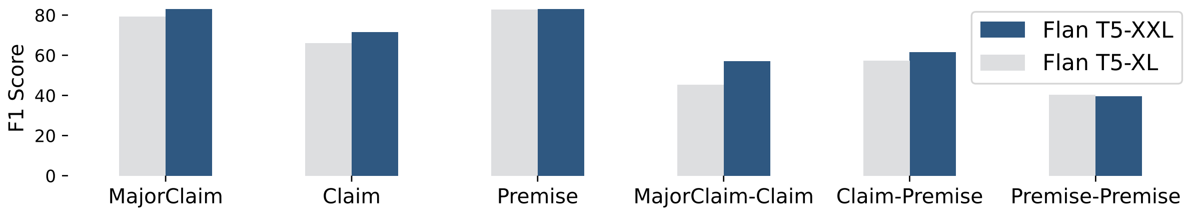


Figure 3: Comparison of F1 score output by label for the Component and Relation tasks.

Gold	[the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies ∣ **claim against**] [its merits still outweigh these downsides ∣ premise ∣ attack = the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies]
FLAN T5-XL	[the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies ∣ **premise** ∣ **support = advertisements have no downsides**] [its merits still outweigh these downsides ∣ premise ∣ attack = the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies]
FLAN T5-XXL	[the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies ∣ **claim against**] [its merits still outweigh these downsides ∣ premise ∣ attack = the advertising expenses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies]

Table 9: Example of output text from FLAN T5-XL and FLAN T5-XXL.

penses lead to a higher product price and some of them express fake information , creating information asymmetry between consumers and companies" to PREMISE and depend on "*advertisements has no downsides*". On the other hand, FLAN T5-XXL correctly predicts that it is a CLAIM with AGAINST relation to MAJORCLAIM.

6 Conclusion

In this paper, we introduced a simple yet strong approach for argument mining (AM) using the TANL framework for text-to-text generation. To simplify and streamline annotations, we eliminated irrelevant text spans from the reference texts. Experimental results obtained from AAEC, AbstRCT, and CDCP demonstrated that our approach outperformed the current state-of-the-art method on these datasets. Our research also indicated the efficacy of employing the TANL framework to predict document structures at the document level. Furthermore, we found that removing irrelevant text spans decreased the inference time by approximately 30% on AbstRCT.

Limitations

Although our method achieves state-of-the-art Component-F1 and Relation-F1 scores across multiple datasets, its inference time poses a significant hurdle for practical implementation. Inference time strongly relies on the length of the input text. Although eliminating irrelevant spans can help reduce this time, our approach still necessitates longer inference time compared to previous methods.

Even though QLoRA reduces memory requirements during training, these large parameter models still require GPUs with a substantial memory capacity, such as the A100 (80GB).

Finally, we note that we only experimented with the TANL framework on encoder-decoder models such as T5 and FLAN T5. Further research is necessary to verify whether our proposed method is compatible decoder-based Large Language models (LLMs) such as GPT-4 (OpenAI, 2023) and LLAMA2 (Touvron et al., 2023)[6].

[6] As a preliminary experiment, we performed few-shot learning using GPT-4 turbo, however, the results were not satisfactory. Further details are provided in appendix B.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

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

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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

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

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

Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. 2021. ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6866–6880, Online. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xinyu Hu and Xiaojun Wan. 2023. Rst discourse parsing as text-to-text generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3278–3289.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4691–4698, Florence, Italy. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based Argument Mining for Healthcare Applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, Santiago de Compostela / Online, Spain.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, Online. Association for Computational Linguistics.

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

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8528–8535.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Huy V. Nguyen and Diane J. Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Joonsuk Park and Claire Cardie. 2018. A corpus of eRulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online. Association for Computational Linguistics.

	AAEC (Paragraph)	AAEC (Essay)
Batch size	32	8
Max length	512	1024
Step	10,000	
Dropout	0.1	
Adam beta1	0.9	
Adam beta2	0.998	

Table 10: Hyperparameters for AAEC.

	AbstRCT	CDCP
Batch size	8	8
Max length	1024	1024
Step	10,000	
Dropout	0.1	
Adam beta1	0.9	
Adam beta2	0.998	

Table 11: Hyperparameters for AbstRCT and CDCP.

A Implementation Details

Our code implementation is based on TANL[7] and QLoRA[8] scripts. We used the T5[9] and FLAN T5[10] models available on Hugging Face.

Table 10 and Table 11 detail the hyperparameters for fine-tuning. We tuned the hyperparameters using the AAEC at the essay level. The learning rates for Base, Large, XL (3B), and XXL (11B) models in both T5 and FLAN T5 were set at 0.0005, 0.0005, 0.0002, and 0.0002, respectively. These were determined through experimenting with all models, adjusting in increments of 0.0001 from 0.0001 to 0.0005. It was observed that models with larger parameter sizes, such as XL (3B) and XXL (11B), showed improved performance with the lower learning rates. However, we confirmed that it was also possible to conduct training using the default learning rate of 0.0005, as suggested in the original TANL paper. Across all tasks, the Adam optimizer was employed. For QLoRA fine-tuning, we applied the same LoRA hyperparameters for all tasks, shown in Table 12.

r	16
lora alpha	32
lora dropout	0.05
bias	none
task type	SEQ_2_SEQ_LM
target modules	q, v, k, o, wo, wi_0, wi_1
load_in_4bit	True
bnb_4bit_quant_type	nf4
bnb_4bit_use_double_quant	True
bnb_4bit_compute_dtype	torch.bfloat16

Table 12: Hyperparameters for fine-tuning with QeLoRA.

	Component	Rrelation
FLAN T5-Base	75.17	51.99
FLAN T5-Large	77.75	56.06
FLAN T5-XL	78.51	56.80
FLAN T5-XXL	**80.15**	**61.19**
GPT4-turbo 3-shot	48.99	24.78
GPT4-turbo 5-shot	51.37	26.43
GPT4-turbo 10-shot	52.79	26.76
GPT4-turbo 20-shot	55.51	28.38

Table 13: Evaluation results of the few-shot learning using GPT-4-turbo. **Bold** denotes the highest F1 score for each task.

B Decoder-based Large Language Models

Table 13 shows the outcomes from experiments using 3-shot, 5-shot, 10-shot, and 20-shot learning with GPT-4-turbo[11]. The Component-F1 and Relation-F1 scores for 3-shot learning were 48.99 and 24.78, respectively, while 20-shot learning improved to 55.51 and 28.38, respectively. These scores are notably lower than those achieved with the fine-tuned FLAN T5 model. Although fine-tuning large language models (LLMs) is a promising direction for future research, finding the most effective prompts for such fine-tuning is still a challenge. Consequently, our study did not extensively investigate the application of LLMs in Argument Mining (AM). Nonetheless, we believe that the text-to-text framework could be effectively integrated with decoders for AM tasks.

[7] https://github.com/amazon-science/tanl

[8] https://github.com/artidoro/qlora

[9] https://huggingface.co/google-t5

[10] https://huggingface.co/google/flan-t5-base

[11] Specifically, we used the gpt-4-1106-preview version of the OpenAI API.