A Generative Approach to Argument Structure Extraction

Rudrendra Bahadur Singh
BBDITM
Lucknow, Uttar Pradesh, India
rudra.rathor20@gmail.com

Manas Pandey
BBDITM
Lucknow, Uttar Pradesh, India
manaspandey.20feb@gmail.com

Mohd. Zubair Ahmed
BBDITM
Lucknow, Uttar Pradesh, India
jubaira646@gmail.com

Mohammed Yameen
BBDITM
Lucknow, Uttar Pradesh, India
yammirza793@gmail.com

Mohd Malik Samran
BBDITM
Lucknow, Uttar Pradesh, India
maliksamran105@gmail.com

Abstract

The goal of argument mining, or AM, is to identify the argumentative structures in a document. Prior approaches necessitate a number of subtasks, including component classification, relation classification, and span identification. Therefore, rule-based postprocessing is required for these methods to extract generative structures from each subtask's output. This method increases the model's complexity and broadens the hyperparameter search space. We suggest a straightforward yet effective technique based on a text-to-text generation strategy employing a pretrained encoder-decoder language model to overcome this challenge. Our approach eliminates the requirement for task-specific postprocessing and hyperparameter optimisation by producing argumentatively annotated text for spans, components, and relations all at once. Additionally, as it is a simple text-to-text creation method, we may readily modify our strategy to fit different kinds of argumentative frameworks. Experimental findings show that our strategy works well, achieving state-of-the-art performance on three distinct benchmark datasets: the Cornell eRulemaking Cor- pus (CDCP), AbstRCT, and the Argument-annotated Essays Corpus (AAEC).

Keywords: Argument Mining, Text-to-Text Generation, T5

Introduction

A type of discourse analysis called argument mining (AM) looks for an argument's structure in a text (Lawrence and Reed, 2019). As seen on the right-hand side of Figure 1, this structure is commonly depicted by a directed acyclic graph or dependency tree. Nodes in the dependency tree represent text spans with arguments, which are further categorised into different sorts of arguments. The relationships between the arguments are shown by the edges connecting the nodes.To uncover the argumentation structure in a variety of domains, including biomedical research (Mayer et al., 2020), student studies (Stab and Gurevych, 2014, 2017), and more, annotated corpora have been created for argument mining. These corpora are the usual benchmark datasets that are used to assess how well argument mining methods perform. Argument mining has recently attracted a lot of attention in discourse analysis due to its useful applications in downstream tasks like text summarisation (Fabbri et al., 2021; Elaraby and Litman, 2022) and automatic essay scoring (Nguyen and Litman, 2018).

Argument mining and other natural language processing tasks were improved by neural models. Identification of the argumentative text span, identification of the argument type, and establishment of the relationship between the two arguments were

the three subtasks of the pipeline technique used by early models (Stab and Gurevych, 2017; Niculae et al., 2017). Recently, however, argument mining is handled end-to-end by methods that regard it as dependency parsing (Ye and Teufel, 2021; Morio et al., 2022). Because each of the three goals must be integrated into the model via distinct techniques, these models are complicated. Therefore, in order to construct legitimate dependency trees, postprocessing is required. Moreover, the implementation of these models is complicated by hyper-parameter adjustment.

In order to overcome these challenges, we used the Translation between Augmented Natural Languages (TANL) (Paolini et al., 2021), a straightforward text-to-text generation model that has demonstrated state-of-the-art performance on sentence-level structured prediction tasks like semantic role labelling, named entity recognition, and relation extraction. By integrating TANL with AM, we provide several benefits: (3) the potential to apply modern large language models; (2) the flexibility to adjust to different annotations depending on the dataset; and (3) a straightforward architecture that does away with complicated postprocessing and hyperparameter tuning.

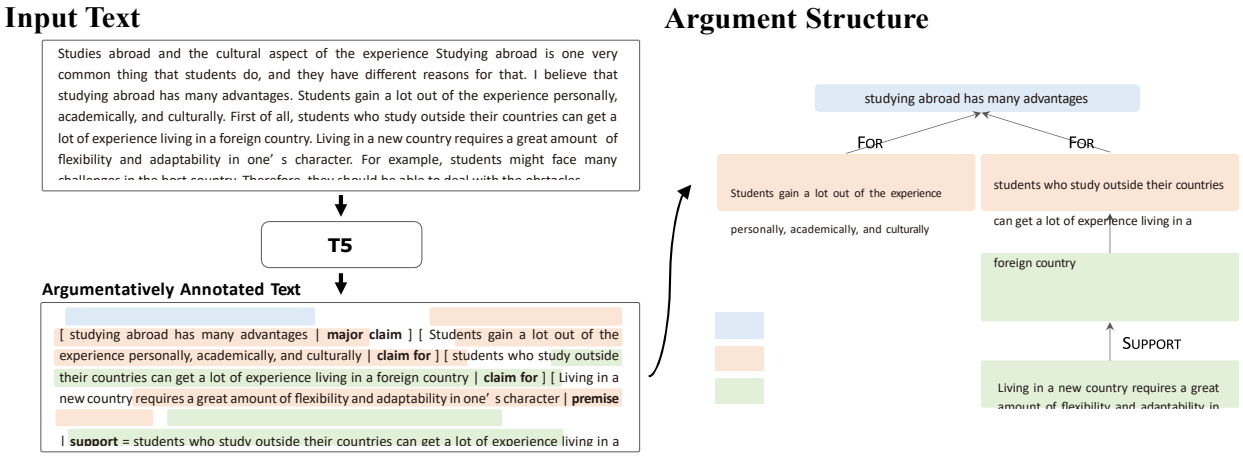


Figure 1: Overview of our methods. For our methodology, we input text into a pretrained encoder-decoder, such as T5 and FLAN T5. This process generates an argumentatively annotated text with spans, components, and relations. We then postprocess the output text to extract the argumentative structure.

According to experimental results from three benchmark datasets—the Argument-annotated Essays Corpus (AAEC) (Stab and Gurevych, 2017), AbstRCT (Mayer et al., 2020), and Cornell eRulemaking Corpus (CDCP) (Park and Cardie, 2018)—our approach used FLAN T5 (Chung et al., 2022):XXL (11B) to achieve the state-of-the-art scores on both Component-F1 and Relation-F1. Furthermore, we were able to cut the computational time for inference in AbstRCT by 30% without sacrificing performance by stopping the model from producing irrelevant text spans, which are not arguments.

Related Work

Argument Mining

Finding the arguments in a text, defining the type of argument, and establishing the relationships between the arguments are the three crucial subtasks that comprise AM. These actions are essential for exposing a text's argumentation structure. Previous approaches employed a pipeline architecture, with relation classifications and component classification1 coming after argumentative span identification (Persing and Ng, 2016; Eger et al., 2017; Kuribayashi et al., 2019; Morio et al., 2020). Nevertheless, this method may lead to the accumulation of mistakes from earlier subtasks.

Recent research use an end-to-end strategy to enhance the pipeline-based approach (Morio et al., 2022; Bao et al., 2022; Ye and Teufel, 2021; Eger et al., 2017). Using a network design based on a biaffine parser, Ye and Teufel (2021) and Morio et al. (2022) obtained state-of-the-art performance on evaluation at the paragraph and essay level. This method parses them using a dependency parsing algorithm and interprets the argumentation structure as a dependency tree. Even though they are end-to-end models, they frequently need an optimal branching algorithm (Morio et al., 2022) or hand-crafted rules (Eger et al., 2017; Ye and Teufel, 2021) to create dependency trees from the outputs of three layers that correspond to subtasks. Because these models integrate three subtasks within a network, they also make it difficult to adjust hyperparameters like the learning rate.

In contrast, Bao et al. (2022) perform AM as a generation task using an encoder-decoder architecture. To anticipate the index of words in the input text, they use a constraint pointer-mechanism (CPM) for BART (Lewis et al., 2020). However, because we concentrate on the text-to-text generation task, our work is different from theirs. This enables us to maximize the use of the decoder without making any modifications to the pretrained language model.

Information Extraction as a Generation Task

Researchers are now tackling information extraction tasks like relation extraction (Huguet Cabot and Navigli, 2021; Lu et al., 2022) and event extraction (Li et al., 2021; Lu et al., 2021) as generation tasks as a result of the recent development of pretrained language models (Raffel et al., 2020; Lewis et al., 2020). For the relation extraction task, Nayak and Ng (2020) contrasted two models: text-to-text generation and copy mechanism-based decoding. However, the findings did not definitively establish the superiority of any approach.

	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.

The method of text-to-text creation developed by Nayak and Ng (2020) is extended by Translation between Augmented Natural Languages (TANL) (Paolini et al., 2021). For tasks like relation extraction, named entity recognition, semantic role labelling, and coreference resolution, this methodology has shown itself to be quite effective. The use of T5, a more potent pretrained encoder-decoder model, is responsible for this achievement. Recent research by Hu and Wan (2023) showed how successful T5 is in analysing sentence structures and suggested using it for sentence-level RST parsing as a type of text-to-text generation. Their work encourages us to consider argument mining, a subset of document structural analysis, as a text generation problem rather than a traditional natural language understanding one. We suggest utilising T5 within the TANL framework to tackle this text-to-text generation issue, which may be a noteworthy solution.

Proposed Methods

An outline of our TANL-based methodology is shown in Figure 1 (Paolini et al., 2021). We match the original text with the specified argumentative spans, their kinds, and relations in order to produce the argumentatively annotated text for TANL. Next, we use the annotated texts to refine T5 using the TANL framework.

Task Formalization

An input text x with n words can be expressed as follows: x = [x1,..., xn]. Extracting spans s = [xstart,..., xend] that include the argument is the goal of span identification. In this case, start and end stand for the indices that denote the span's start and finish,

respectively. The notation for these extracted spans is (start, end). The discovered spans are given component labels c from a set C using component categorisation. All of the component labels found in the dataset are contained in Set C. Components are shown as (start, end, c) as a result of the classification. Assigning a relation label r from a specified set R and choosing source and target spans from the extracted spans are the steps involved in relation classification. All of the dataset's relation labels are stored in Set R. (startsrc, endsrc) and (starttgt, endtgt) are the expressions for the source and tar-get spans, respectively. Thus, the relation classification output can be represented as follows: (startsrc, endsrc, starttgt, endtgt, r).

Argumentatively Annoted Text

We modify TANL's joint entity and relation extraction task's output format for AM. We express it as "[ssrc | c | r = stgt]" when a text span ssrc with a particular component label c depends on another text span stgt via a relation label r. However, we omit stgt and the relation label r if the span ssrc is independent of others, indicating it as "[ssrc | c]". An example showing how to apply TANL's approach to AM is provided below:

Input: As a result, numerous marine species are now in danger of extinction, and in extreme cases, a portion of the reef is no longer habitable by these marine species. Therefore, it is clear that tourism has put natural environments in danger.

Output: As a result, [much marine life has been endangered, in the extremes a portion of the reef has become uninhabitable for these marine species | premise | support = tourism has threatened the nature environments]. The fact that [tourism has threatened the wild environments | claim for] is so evident.

Elimination of Unnecssary Text Spans

TANL makes an effort to annotate textual structure while preserving the original input text's integrity. However, unlike the original TANL, which required sentence-level inputs, our attention shifts to tasks that require documents as input. For effective computing, the encoder and decoder models must minimise the maximum number of tokens. Therefore, nonargumentative spans are not included in TANL's annotation scheme. Examples of annotations with and without nonargumentative spans are displayed in Table 2.

Input Text	Advantages and disadvantages of the prevalent of English With the development of global- ization , 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 global- ization , 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.

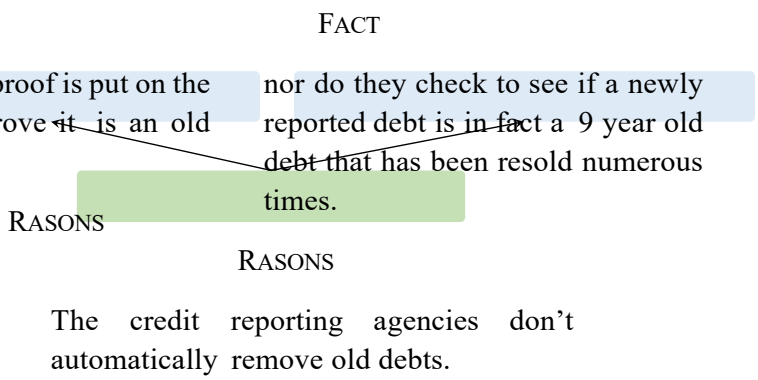
Argumentative Structure	
repeated representation	[The credit reporting agencies don't automatically remove old debts . | value | reasons = nor do they check to see if a newly reported debt is in fact a 9 year old debt that has been resold numerous times .] [The credit reporting agencies don't automatically 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.

Representation of Components with Multiple Parents

There are elements in the AM dataset that rely on several parents. Since TANL's joint entity and relation extraction operation simply adds annotations to the input text without repeating or deleting anything, the output format is unable to represent such structures in text. Two representations are used to handle components with many parents: serial representation and repeating representation3. Table 3 provides examples of these. For example, a VALUE component that reads, "The credit reporting agencies don't automatically remove old debts," depends on two REASONS components. The serial representation shows the first relation first, then the second, but the repeated representation treats these as two distinct relations and shows them successively.

Experiments

Datasets

The Argument-annotated Essay Corpus (AAEC) (Stab and Gurevych, 2017), AbstRCT (Mayer et al., 2020), and the Cornell eRulemaking Corpus (CDCP) (Park and Cardie, 2018) were the three main benchmark datasets that we used. Annotations on components and relationships for student writings are included in AAEC. Both essay-level and paragraph-level statistics are included. While AM is conducted on the predetermined paragraphs at the paragraph level, it is performed on the full essay as input at the essay level. AAEC offers four relation labels (R = FOR, AGAINST, SUPPORT, ATTACK) and three component labels (C = MAJORCLAIM, CLAIM, PREMISE}). A CLAIM is always dependant on a MAJORCLAIM, under the AAEC annotation requirements.Consequently, we modified the labels in our trials to include two relation labels R = {SUPPORT, ATTACK} and four component labels C = {MAJORCLAIM, CLAIMFOR, CLAIMAGAINST, PREMISE}. We evaluate CLAIMFOR and CLAIMAGAINST in the same way as in the earlier experiments, treating them as identical to CLAIM. Table 1 indicates that 28.09% of the total words are in the nonargumentative span. Note that a component does not have more than one parent. Randomised Controlled Trials (RCT) for a variety of illnesses, including neoplasm, glaucoma, hepatitis, diabetes, and hypertension, are the source of AbstRCT from the MEDLINE.It has three relation labels R = {SUPPORT, ATTACK, PARTIAL-ATTACK} and three component labels C = {MAJORCLAIM, CLAIM, EVIDENCE}. Nonargumentative spans make up a sizable percentage of this dataset, as shown in Table 1. Nonargumentative span words make up 49.30 percent of the total word count.

The CDCP is annotated with links and components to allow for citizen feedback. It offers two relation labels, R = {REASONS, EVIDENCE}, and five component labels, C = {FACT, TESTIMONY, VALUE, POLICY, REFERENCE}. No nonargumen-tative spans are present in the CDCP, as Table 1 illustrates. Additionally, we found that compared to the other two datasets, CDCP has a higher number of components that depend on several other components.

Fine-tuning T5 with QLoRA

In earlier research, TANL carried out thorough fine-tuning—a procedure that involves updating every parameter—on the T5-Base. However, the purpose of this study is to investigate the impact of additional parameters. We do this by using QLoRA (Dettmers et al., 2023) adjustment to minimise GPU memory usage when training big parameter models like T5-XL (3B) and T5-XXL (11B). By quantising the model and applying Low-Rank Adapters (LoRA) (Hu et al., 2022), QLoRA is an adaptor that helps lower the number of parameters that need to be trained while preserving performance levels that are on par with full fine-tuning.

Settings

We use the train/dev/test split recommended by Eger et al. (2017) for the AAEC. For every dataset, there are 286, 36, and 80 articles, and 1587, 199, and 449 paragraphs, respectively. We followed the split suggested in the original paper (Park and Cardie, 2018) and utilised the full test set for the AbstRCT, allocating 300 for training, 50 for development, and

100 for testing. Out of the 731 comments in the CDCP, we kept 150 as a test set in accordance with (Niculae et al., 2017). 15% of the training data was taken out as a development set for the CDCP.

We looked at two pretrained encoder-decoder models that were utilised in the TANL framework: T5 (Raffel et al., 2020) and FLAN-T5 (Wei et al., 2022). Four distinct parameter models were used in our experiments: Base (220M), Large (770M), XL (3B), and XXL (11B). We present the average scores from three runs with various seeds in each experimental setup.

Compared Models

We contrasted our approach with the state-of-the-art model and the following models:
• Integer Linear Programming (ILP) (Stab and Gurevych, 2017): This feature-based technique uses ILP to pipeline the parsing of each subtask.
• BLCC: An approach that views Argument Mining (AM) as a sequence tagging problem (Eger et al., 2017).
• LSTM-ER (Eger et al., 2017): This end-to-end relation extraction model combines tree structure with sequential LSTM models and employs LSTM-ER (Miwa and Bansal, 2016).
• BiPAM-syn: A model that uses BERT as a language model for end-to-end dependency parsing, integrating syntactic information and biaffine operations (Huang et al., 2021).

• BART-CPM (Bao et al., 2022): A model that is similar to our encoder-decoder but uses the Constrained Pointer Mechanism (CPM) and BART (Lewis et al., 2020) as the language model.

• Single Task (ST) model (Morio et al., 2022): A cutting-edge AM model that uses Longformer (Beltagy et al., 2020) as the language model and a biaffine neural method similar to BiPAM-syn.

		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.

Evaluation Measures

The de-facto standard assessment metrics in the area, the Component-F1 score and the Relation-F1 score4, were used to assess our techniques (Stab and Gurevych, 2017; Eger et al., 2017; Ye and Teufel, 2021; Morio et al., 2022; Bao et al., 2022). The component classification and relation classification tasks of AbstRCT and CDCP, given an oracle span, have been the subject of numerous investigations, in contrast to AAEC (Kuribayashi et al., 2019; Morio et al., 2020; Mayer et al., 2020). We compared our results to those of previous studies that also examined Component-F1 and Relation-F1 scores in an end-to-end manner in order to assess AbstRCT and CDCP.

Implementation Details

In order to train the model, we used a batch size of 32 for paragraph-level AAEC and 8 for essay-level AAEC, in addition to AbstRCT and CDCP. The maximum token length is 1,024 for the other datasets and 512 for AAEC at the paragraph level. The Base and Large models had a learning rate of 0.0005, whilst the XL (3B) and XXL (11B) models had a learning rate of 0.0002. With checkpoints every 200 steps, all training was conducted on a single A100 (80GB) GPU across 10,000 steps.

The input text may not always be accurately replicated by encoder-decoder models, leading to identified text spans that are different from those in the original text. To mitigate this issue, TANL uses the Needleman-Wunsch alignment algorithm (Needle- man and Wunsch, 1970) to establish alignment be- tween the output and input text spans. This procedure, which we also used, establishes the placement of words in the input text within the output text.

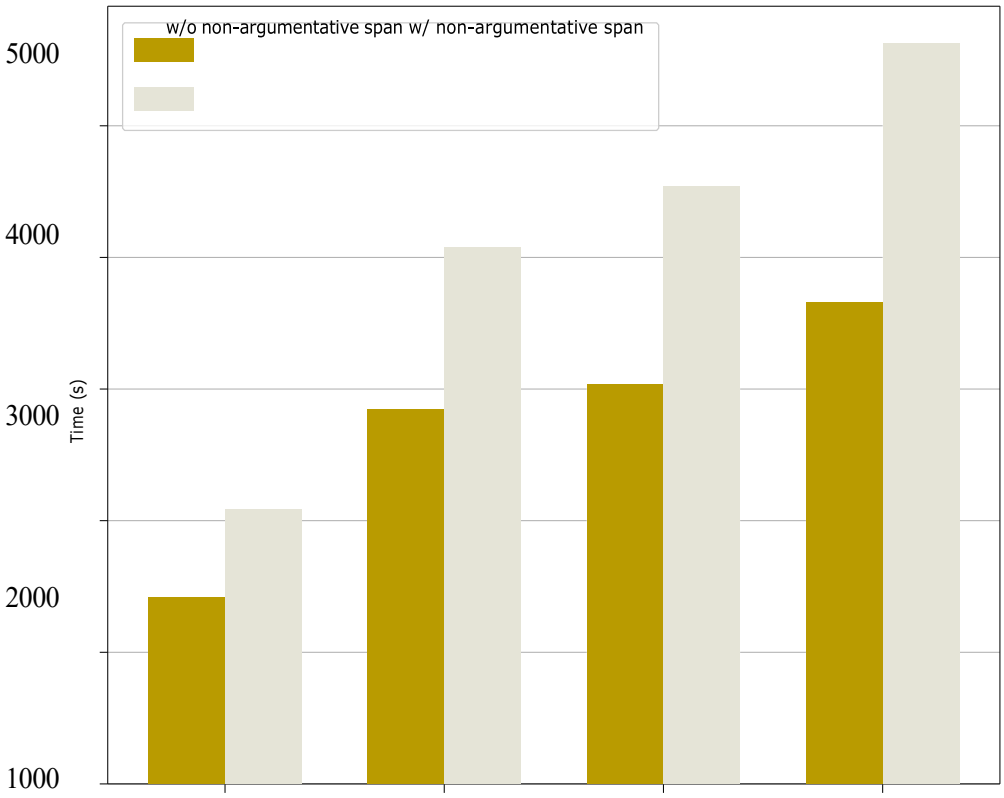
		AAEC (Essay)		AAEC (Paragraph)		AbstRCT	
Output Format	**Model**	**Component**	**Relation**	**Component**	**Relation**	**Component**	**Relation**
FLAN-T5 Base		74.99	50.87	74.97	57.54	65.30	34.55
w/ nonargumentative span	FLAN T5-Large	77.76	55.62	76.53	59.09	69.47	39.66
	FLAN T5-XL	78.73	57.21	77.17	61.03	73.13	42.39
FLAN T5-XXL		**80.59**	60.37	**79.06**	**62.38**	72.78	47.11
FLAN-T5 Base		75.17	51.99	75.55	58.51	68.76	38.31
w/o nonargumentative span	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 with- out nonargumentative spans in AbstRCT. We set the batch size to 2 for all models during inference and mea- sured the time required to complete the process on the entire test dataset.



Results and Discussion

Main Results

The AAEC results are displayed in Table 4. The findings imply that considerable performance improvements are typically obtained by increasing the parameters. Our models' performance is significantly improved when utilising models with billion-scale parameters, even though they do not outperform state-of-the-art models when using base or large models. While models with 11B parameters outperformed the current top F1 scores, our model with 3B parameters (T5-XL and FLAN-T5-XL) achieved F1 scores that were on par with the state-of-the-art models. In particular, our FLAN T5-XXL model produced Relation-F1 scores of 61.19 and 61.87, and Component-F1 values of 80.15 and 78.40 at the essay and paragraph levels, respectively.

Our simple model design using QLoRA works well even with a higher number of parameters. The data also shows that FLAN T5 outperforms T5, suggesting that the AM task benefits from instruction-tuning on other tasks using FLAN (Wei et al., 2022).

Additionally, we display the AbstRCT and CDCP results in Tables 5 and 6, respectively. Our FLAN T5-XXL obtained state-of-the-art performance for AbstRCT, with a Relation-F1 score of 47.66 and a Component-F1 score of 72.86. With Component-F1 and Relation-F1 scores of 72.68 and 33.96, respectively, the CDCP findings likewise show state-of-the-art performance. In all datasets, our model performs noticeably better than current models, demonstrating the efficacy of our method: text-to-text creation using the TANL framework.

Differences of Hyperparameter Tuning

Our model's simplicity gives our approach a competitive edge over earlier approaches. For subtasks like span identification, component classification, and relation classification, earlier approaches had separate hyperparameters. To identify the ideal hyperparameters, Morio et al. (2022) used Optuna (Akiba et al., 2019). However, because the three subtasks are interdependent, hyperparameter adjustment might become difficult. When utilising

large models, this complexity increases and could make training more difficult.

In contrast, our approach avoids the challenges of hyperparameter tuning that come with earlier approaches because it just uses the learning rate as its hyperparameter. This ease of use is especially beneficial when using large-scale language models.

Eliminating NonArgumentative Spans

Unrelated text spans that are not arguments are not produced by our model. We evaluated the model without removing the following three tasks5: AbstRCT, AAEC at the paragraph level, and AAEC at the essay level in order to gauge the effect of this removal.

The results are displayed in Table 7. It is clear from all three tasks that the eliminations do not impair performance. Additionally, removing unnecessary spans can shorten inference time; this effect is especially noticeable in AbstRCT because of the large number of nonargumentative spans. The run time of the inference on AbstRCT is shown in Figure 2. The effectiveness of the eliminations is seen in the figure. For Base, Large, XL, and XXL models, our approach cuts the inference time by about 30%.

Comparison of Component Representations with Multiple Parents

A comparison of two distinct representations—repeated and serial—across the whole test dataset from the CDCP (Full Dataset) is shown in Table 8. The Component-F1 and Relation-F1 scores, which are only computed for components with multiple parents (Multiple Parents), are also included in the table. The data shows that the two representations' performance in the component categorisation does not differ much. Nonetheless, across all models, the serial representation consistently performs better than the other models in relation classification. The findings imply that repeated representation might not be the best option for tasks like relation categorisation that call for the extraction of long-term relationships. A comparison of two distinct representations—repeated and serial—across the whole test dataset from the CDCP (Full Dataset) is shown in Table 8. The Component-F1 and Relation-F1 scores, which are only computed for components with multiple parents (Multiple Parents), are also included in the table. The data shows that the two representations' performance in the component categorisation does not differ much. Nonetheless, across all models, the serial representation consistently performs better than the other models in relation classification. The findings imply that repeated representation might not be the best option for tasks like relation categorisation that call for the extraction of long-term relationships.

Performance Improvement with Increasing Model Parameters

When compared to FLAN T5-XL, FLAN T5-XXL demonstrated a notable improvement in Relation-F1 performance in the essay-level AAEC test (Table 4). Surprisingly, the gain is about 4.4 points. We go over the F1 ratings for each Component and Relation label to further examine the findings. The results are displayed in Figure 3. We found that CLAIM significantly outperformed MA-JORCLAIM (79.34 vs. 82.96) and PREMISE (82.87 vs. 83.01) for the component label (66.12 vs. 71.57). The MA-JORCLAIM-CLAIM relation showed the biggest improvement when we looked at relation labels (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.

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.

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-penses lead to a higher product price and some of them express fake information , creating informa- tion 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.

Conclusion

Using the TANL framework for text-to-text generation, we presented a straightforward yet effective argument mining (AM) technique in this research. We removed unnecessary text spans from the reference texts in order to streamline and simplify annotations. Experimental findings from AAEC, AbstRCT, and CDCP showed that our strategy performed better on these datasets than the state-of-the-art methodology at the moment. Additionally, our study demonstrated the effectiveness of using the TANL framework to forecast document topologies at the document level. Additionally, we discovered that eliminating unnecessary text spans reduced the AbstRCT inference time by about 30%.

Limitations

Our method's inference time is a major obstacle to practical implementation, despite the fact that it achieves state-of-the-art Component-F1 and Relation-F1 scores across multiple datasets. The length of the input text has a significant impact on inference time.

Even though removing unnecessary spans can help cut down on this time, our strategy still requires more inference time than earlier approaches. These huge parameter models still need GPUs with a significant memory capacity, as the A100 (80GB), even if QLoRA lowers memory requirements during training.

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].

References

1. 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)*, pages 2623–2631, New York, NY, USA. Association for Computing Machinery.
2. 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.
3. Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020.**Longformer: The long-document transformer.** *arXiv:2004.05150.*
4. Hyung Won Chung et al. 2022. **Scaling instruction-finetuned language models.**
5. Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. **QLoRA: Efficient finetuning of quantized LLMs.** *arXiv preprint arXiv:2305.14314.*
6. 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.
7. 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.
8. Alexander Fabbri et al. 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.
9. Edward J. Hu et al. 2022.**LoRA: Low-rank adaptation of large language models.** In *International Conference on Learning Representations.*
10. 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.
11. 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.
12. 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.
13. Tatsuki Kuribayashi et al. 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.
14. John Lawrence and Chris Reed. 2019. **Argument mining: A survey.** *Computational Linguistics*, 45(4):765–818.

15. Mike Lewis et al. 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.
16. 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.
17. Yaojie Lu et al. 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.
18. Yaojie Lu et al. 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.
19. 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.
20. 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.
21. Gaku Morio et al. 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.
22. Gaku Morio et al. 2022.**End-to-end argument mining with cross-corpora multi-task learning.** *Transactions of the Association for Computational Linguistics*, 10:639–658.
23. 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.
24. 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.
25. 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.
26. 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.
27. OpenAI. 2023.**GPT-4 Technical Report.**
28. Giovanni Paolini et al. 2021.**Structured prediction as translation between augmented natural languages.** In *9th International Conference on Learning Representations (ICLR 2021).*
29. 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).
30. 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.
31. Colin Raffel et al. 2020.**Exploring the limits of transfer learning with a unified text-to-text transformer.***Journal of Machine Learning Research*, 21(140):1–67.

32. 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.
33. Christian Stab and Iryna Gurevych. 2017.**Parsing argumentation structures in persuasive essays.***Computational Linguistics*, 43(3):619–659.
34. Hugo Touvron et al. 2023.**LLaMA 2: Open foundation and fine-tuned chat models.**
35. Jason Wei et al. 2022.**Finetuned language models are zero-shot learners.** In *International Conference on Learning Representations.*
36. 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.