A Unified Representation and Deep Learning Architecture for Argumentation Mining of Students' Persuasive Essays

Muhammad Tawsif Sazid[1],*, Robert E. Mercer[1]

[1]Department of Computer Science, The University of Western Ontario, London, Ontario, Canada

Abstract

We develop a novel unified representation for the argumentation mining task facilitating the extracting from text and the labelling of the non-argumentative units and argumentation components—premises, claims, and major claims—and the argumentative relations—premise to claim or premise in a support or attack relation, and claim to major-claim in a for or against relation—in an end-to-end machine learning pipeline. This tightly integrated representation combines the component and relation identification sub-problems and enables a unitary solution for detecting argumentation structures. This new representation together with a new deep learning architecture composed of a mixed embedding method, a multi-head attention layer, two biLSTM layers, and a final linear layer obtain state-of-the-art accuracy on the Persuasive Essays dataset. An augmentation of the corpus (Paragraph version) by including copies of major claims has further increased the performance.

Keywords

deep learning model, unified representation, data augmentation, word embeddings, argumentation mining sub-tasks, natural language processing

1. Introduction

Arguments consist of claims and premises and the relationships among them. Argumentation mining, a research topic in the field of Natural Language Processing (NLP) aims to identify the arguments in a text document and the internal structure of each argument. There are four subtasks of the problem. Since we are using the Persuasive Essay (PE) dataset [1] these subtasks can be made more precise: 1) Segmenting the argument components: separate the argumentative text from the non-argumentative text, 2) Labelling each argument component: whether the argumentative text is a Major-Claim, Claim, or Premise, 3) Determining which argumentation components are in a relationship: this is represented as the text distance (the number of sentences before or after) between a premise and its related argument component (in the PE corpus, which major-claim is related to a claim is not annotated), and 4) Classifying the stance of the relations

Workshop on Computational Models of Natural Argument (CMNA 22), 12 September 2022, Cardiff, Wales

*Corresponding author.

msazid@uwo.ca (M. T. Sazid); mercer@csd.uwo.ca (R. E. Mercer)
0000-0002-2355-508X (M. T. Sazid); 0000-0002-0080-715X (R. E. Mercer)

between argument components: as “support” or “attack” between premises and claims or other premises; and between claims and major claims as “for” or “against”.

Previous research has approached the development of a computational argumentation mining method from two distinct viewpoints. The first approach views the input as text and searches for a method to solve all four of the subtasks mentioned above. Stab and Gurevych [1] and Eger et al. [2] are noteworthy examples of this strategy. Eger et al. [2] produce the state-of-the-art method to which we compare our new method. Both of these works approach argumentation mining as a sequence tagging problem. They first detect entities and then predict the argument structure on top of that.

The second view of the argumentation mining problem assumes the first subtask, the segmenting of the text into argumentative and non-argumentative components has been done, and the input to the method are the argumentative components. We compare some of our results with some of these works.

The method proposed here takes the first approach, solving all four subtasks. As there are subtasks, previous argumentation mining works have decoupled various subtasks, solved them separately, and then combined the solutions. The end-to-end learning method proposed here differentiates itself from these previous works by approaching the problem in a unified manner. Our research contributions are summarized as follows:

1. Argumentation mining is formulated as a single problem by integrating all of its subtasks: separating the non-argumentative tokens from the argumentative tokens, labelling the argument components, identifying the related components, and classifying the stance of the relation. We show that combining all the subtasks results in improved performance.
2. By constructing this novel dense representation of the problem, we are able to achieve a better than previous performance using a stacked embedding model comprising two biLSTM layers, a multi-head attention layer, and a linear layer.[1]
3. We have developed an augmentation technique (this experiment has been done on the paragraph version of the PE corpus) based on the n-gram tokens that indicate the start of a major claim which has further improved the results on the PE corpus.

With the new formulation of the problem, our model, Unified-AM, reaches state-of-the-art argument mining performance on detecting and labelling argument components and relations for the paragraph version of the PE corpus.

2. Related Work

Computational argumentation mining deals with finding argumentation structures in text. Palau and Moens [3] established that argument mining would need to detect claims and premises and their relationships.

Stab and Gurevych [1, 4] provided the PE dataset, a corpus annotated with a scheme that includes claims, premises, and also attack or support relations. Stab and Gurevych [1] addressed the argumentation problem by training independent models for each of

[1] We will make our code publicly available once our paper is published.

the subtasks and then combining them with an Integer Linear Programming Model for the end-to-end task. Eger et al. [2] achieved state-of-the-art performance on the PE corpus by addressing the problem as a sequence tagging problem. They have the best accuracy of 61.67% by using a modified version of the LSTM-ER model which had been introduced by Miwa and Bansal [5]. The LSTM-ER model uses a stacked architecture of Sequence and Tree LSTMs.

Persing and Ng [6] presented the first findings on end-to-end argument mining in student essays using a pipeline approach by performing joint inference using an Integer Linear Programming (ILP) framework. Ferrara et al. [7] introduced an unsupervised approach, topic modeling, to detect claims and premises. Persing and Ng [8] have also developed an unsupervised machine learning method that provides all but the stance information for the relations.

A number of works have investigated approaches for subtasks 2, 3, and 4. Early work is epitomized by Peldszus [9] and Pelszus and Stede [10]. Potash et al. [11], Kuribayashi et al. [12], and Bao et al. [13] are more recent. Niculae et al. [14] jointly approach unit type detections and relation predictions on their new CDCP dataset and the PE dataset.

We investigated some neural architectures and how additional handcrafted features can help boost the accuracy on certain sequence tagging tasks. Ahmed et al. [15] achieved state-of-the-art performance in Part of Speech, Named Entity Recognition, and Chunking tasks by combining different learned vectors with word-level embeddings. Kuribayashi et al. [12] and Persing and Ng [8] also noted the importance of discourse connectives in the argumentation mining task.

3. Research Methodology

Here we present our method to generate the argumentation structure for the PE data set. First, the data set is described. Then, we introduce the multi-label representation that considers argumentation mining as a single unified problem. Lastly, instead of presenting the final model with an ablation study, we present our method in a bottom-up style, starting with a base architecture to which we add, providing in Table 1 the performance increase given by each addition since we want to discuss the motivation for these additions. We compare the final model's performance with that achieved by Eger et al. [2].

3.1. Data Set Preparation

The PE dataset that we are using in this paper was created by Stab and Gurevych [1] and was used in Eger et al. [2]. The essays are written on controversial topics so that the authors can make their opinions and take their stances. The corpus has been tagged with the BIO scheme. There are essay and paragraph versions of the data set. We have worked with the paragraph version of the corpus. The data set contains 1,587 paragraphs totaling 105,988 tokens in the train-set and 449 paragraphs, 29,537 tokens in the test-set[2]. The development set has 12,657 tokens available in 199 paragraphs.

[2] This differs slightly from what is detailed in Eger et al. [2]

The argumentation structure can be viewed as a forest with each tree rooted by the author's major claim. The claims are connected to all of the major claims with either 'for' or 'against' relations. Premises are related to exactly one claim or premise. Premises either 'support' or 'attack' the claims or premises. One important piece of information is that the argumentation structure is completely contained in the paragraph except for some relations from claims to major claims which are not in the same paragraph. We extracted the dataset at the paragraph level for this paper. The corpus is imbalanced as Eger et al. [2] has mentioned.

3.2. New Problem Formulation

To integrate all of the sub-problems (argumentative and non-argumentative unit classification; major-claim, claim, and premise component classification; relation identification, and distance between 2 entities) into a single problem, we construct a binary vector of size 33 for our target labels. This novel unified representation has 33 indexes representing different components related to argumentative units. We are addressing the argumentation mining problem as a sequence tagging problem and classifying each word or token as beginning argumentative/continuation argumentative/non-argumentative; premise/claim/major-claim; support/attack; for/against; relative distance between the current component and the component it relates to. One of these distance indexes will represent the related component. The maximum distances from a premise to a claim suggested in Eger et al. [2] are $+11$ and -11 (the number of sentences after or before, respectively). Thus, we have constructed a dense unified representation of the argumentation mining problem. In this representation, the value '1' signifies that the index belongs to that particular (non-)argumentative unit, or in the case of argumentative components, the index is the continuation or beginning of that component; '0' indicates otherwise. By formulating the argumentation mining task as a multi-label problem, we don't have to think of separated and decoupled solutions for each of the subtasks. As an example, the word "children" of the sentence "For instance, children immigrated to a new country will . . . " would be represented by the vector [0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0] which indicates this is the beginning of a premise and that it is supporting a claim that is three sentences after it. After getting the model's logit values during training and evaluation phases, we choose the index for each of the categories that has the highest logit value (see [16] for details) in that specific category (components, stances, and distance).

4. Description of the Deep Learning Model and the Hyper-Parameters

Our deep learning model architecture (Unified-AM) includes: stacked embedding, axial positional embedding, a multi-head attention layer, a 2-layered biLSTM, and the final linear layer. The output of the model is optimized with BCEWithLogitsLoss.

For its capability of retaining long-distance information from sequential texts, we use biLSTM for the paragraph level for the argumentation mining task. Before adding

Table 1
Error Analysis and Comparison Between the Three Models (False Positives + False Negatives)

	Number of Wrong Predictions				
	Major Claim	**Claim**	**Premise**	**Relations**	**Non Argumentative**
Trained Embedding	1306	4011	4787	10004	3255
Stacked Embedding	1176	3215	3122	7653	2120
Unified-AM	1111	3082	2953	7301	2202

the axial positional embedding and the multi-head attention layer, our preliminary experimentation determined the number of biLSTM layers by using a trial and error methodology, i.e., we have tried two layers of biLSTM with one linear layer, one biLSTM layer with one linear layer, and so on. We have found one linear layer and two biLSTM layers achieve the best accuracy.

The final architecture includes mixed embedding but in the model design we first experimented with a plain embedding layer instead. Lample et al. [17], have shown that a combination of different embeddings may work better than using only one embedding class. For the pre-trained mixed embedding, we use the memory-efficient stacked embedding class that Akbik et al. [18] introduced in their Flair framework for combining the FastText and Byte-pair embeddings. As our corpus contains unknown words in the test set and the whole corpus contains many suffix and prefix dependent words, we used these two types of embedding together. The final design decision was to include the multi-head attention [19] and the axial positional embedding for the positional information [20, 21].

To show the effects of each of these design decisions, we compare the number of wrong-predictions between our non-pre-trained embedding model, the pre-trained stacked embedding model, both without multi-head attention, and the final Unified-AM model. Table 1 shows the error analysis of these three stages of architecture design for the non-argumentative units, argumentative components, and relations. For each of the mentioned argumentative units we present the total number of errors (false negatives + false positives). For relations (support, attack, for, and against), we have combined the errors from each class and report this combined value. There are somewhat fewer wrong predictions when the stacked embedding is incorporated into the model. Without stacked embedding, the total number of wrong predictions for all of the classes on the paragraph level is 23,363. With the addition of stacked embedding the total number of wrong predictions becomes 17,286. After using this pre-trained embedding, the error rate is reduced by 26.01%. The total number of errors for the Unified-AM model is 16,649. The Unified-AM model further reduces the error rate by 3.69%.

We have formulated the argumentation problem in a unified way. As a result, it has become a multi-class, multi-label problem. As it becomes a multi-label problem when we create a unified representation, we just want to choose the index for each of the categories that has the highest logit value in that specific category (components, stances, and distance). For this, we have created a function to interpret our multi-label outputs from the Unified-AM model.

After trying several hyperparameter values for each of the different components we

Table 2
Experiment on the Paragraph Level with Unified-AM Compared to LSTM-ER [2]

Paragraph Level							
Model	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	C-F1	R-F1
Unified-AM (Ours)	**66.79%**	68.88	**78.22**	**51.14**	**56.41**	**60.00**	**67.32**
LSTM-ER [2]	61.67%	**70.83**	77.19	45.52	50.05	55.42	60.72

have chosen the final values. We use dropout values of 0.5 for the linear layer, and 0.65 for the biLSTM layer of our architecture. We use the default dropout value (0.0) for the multi-head attention layer. We do not use any type of activation function in-between the layers. A learning rate of 0.001 has been used in all of the experimental design stages. The Adam optimizer is used throughout. During training, we have used random shuffling for all of the final experiments. We have trained our model around 1000-1100 epochs for all of the experiments except the data augmentation experiment (see Table 4). For determining the default training epochs (1000-1100) we have closely observed the development set accuracy value after every 5 epochs. If after 1100 epochs the development set accuracy stops increasing or starts fluctuating somewhat between a small range of accuracy values, we have stopped the training procedure. We also observe the training loss and find that when it reaches around 0.0005 loss value, the model has the highest development set accuracy. If we further train and decrease the loss value, it does not help to improve the accuracy value of the development set. As we have also increased the original PE corpus by augmenting the data in our augmentation experiments (see Section 5.2), we also increase the training epochs to reach around the 0.0005 training loss which has given us improvements regarding the C-F1, R-F1 and F1 scores.

5. Experiments and Results

5.1. Experiments on the original version of the PE corpus

We have used the multi-head attention module and have fed the stacked embedding representation for all tokens to the query, key and value matrices that we are using for solving the argumentation mining task. For our 400-dimension embedding class we use four heads for the multi-head attention layer for this experiment. This new approach achieves the highest token level accuracy of 66.79% in our argumentation mining task. Table 2 summarizes the result for this experiment, including the F1 measure for the component and relation tasks, and a global F1 score. The results from Eger et al. [2] have been included for comparison. Now, compared to the Eger et al. [2] decoupled method for computing the relation identification, this task in Unified-AM is coupled with the component identification task due to the unified representation of the problem, which has led to the better performance. We have used the distance values from -11 to +11 that were observed by Eger et al. [2] in the PE data set. Regarding this problem, our target label vector is: Y = [Non-Argumentative, Beginning, Continuation, Major-Claim,

Table 3
Precision, Recall and F1-score for the Argumentation Mining Classes for Unified-AM

Class	Precision	Recall	F1 Score	Token Percentage
Non-Argumentative	88.38	88.27	88.33	32.20
Major-Claim	73.87	74.18	74.02	7.41
Claim	65.37	58.05	61.48	15.41
Premise	88.01	90.87	89.42	44.99
Support	86.79	89.69	88.22	42.61
For	60.96	57.05	58.94	12.77
Attack	32.52	26.77	29.37	2.38
Against	60.81	29.97	40.15	2.64

Claim, Premise, Support, For, Attack, Against, (-11 to +11)] (33 labels). For this task we have used our final model architecture. Table 2 shows the results.

In Table 3, we present individual precision, recall and F1 score for the 8-label representation of the components and relations that are available in the PE corpus. We observe low precision and recall score for the claim tokens even though the class is not the least frequent one in the PE corpus. We observe similarity as the lowest agreement score among the human annotators is also for the claim [1]. Unified-AM also finds it difficult to predict the claim tokens in the corpus.

5.2. Data Augmentation Experiment on the Paragraph Version of the PE Corpus

We now turn to the final argumentation model performance improvement. Adding linguistic information to a model has been successful for low level NLP tasks [15]. We have observed (as did Kuribayashi et al. [12], and Persing and Ng [8]) that many major claims are prefaced by a reasonably small set of n-grams. An n-gram is a continuous sequence of *n* words. Some examples of the n-grams that are found in the PE corpus are: ‘I firmly believe that’, ‘In conclusion ,’, ‘Hence ,’, and ‘Firstly ,’. We consider augmenting the corpus by using these n-grams to increase the frequency of the Major Claim component type which is the least frequent component available in the PE corpus.

In this experimental setup, we have augmented the PE training dataset (which consists of paragraphs). Below, we describe the augmentation technique that we have used to augment the PE corpus. We also compare the performance of Unified-AM on both of the augmented and original corpora.

We have augmented the paragraph-level corpus with new paragraphs. These new paragraphs are copies of those paragraphs that contain one of the 108 n-gram tokens that occur immediately before the major claim tokens but have had the n-gram randomly swapped with a same size n-gram token. This augmentation increases the number of major claim tokens in the whole corpus but with different introductory n-grams. We have hypothesized that if we increase the root element, i.e., the major claim components of the corpus, by swapping frequently occurring n-gram tokens that appear immediately before the component, it would help the model to accurately detect this type of component and differentiate between the three types of components that are available in the PE corpus. We have shown below an example of the original paragraph and the augmented paragraph after applying the described augmentation method:

Original Paragraph: "It is always said that competition can effectively promote the development of economy . In order to survive in the competition , companies continue to improve their products and service , and as a result , the whole society prospers . However , when we discuss the issue of competition or cooperation , what we are concerned about is not the whole society , but the development of an individuals whole life . I firmly believe that we should attach more importance to cooperation during primary education."

Augmented Paragraph: "It is always said that competition can effectively promote the development of economy . In order to survive in the competition , companies continue to improve their products and service , and as a result , the whole society prospers . However , when we discuss the issue of competition or cooperation , what we are concerned about is not the whole society , but the development of an individuals whole life . I truly believe that we should attach more importance to cooperation during primary education."

Description of the Augmentation Process: In this particular example we have substituted the n-gram "I firmly believe that" with an equal size randomly chosen n-gram "I truly believe that" from our collected n-gram list. The words following in that particular sentence are major claim tokens. Here, the n-grams consist of 4 words.

By using the augmentation technique, we have increased the number of Major Claim tokens by approximately 4000. Also, because Claim, Premise, and Non-argumentative components occur in these paragraphs, the number of Claim, Premise, and Non-argumentative tokens have increased by around 2000, 1000, and 8000, respectively.

After creating the augmented corpus, we have trained our Unified-AM model on the corpus. We have achieved the highest token level accuracy on the paragraph-level argumentation corpus. Previously, without augmentation, we have achieved 66.79% token level accuracy on the PE dataset (see Table 2) and after applying the augmentation methodology we have achieved the highest token level accuracy of 68.02%. Also, all other performance measures have been improved significantly. We further worked on the paragraph-level augmentation and trained it more (1500-1600 epochs). Table 4 shows the results related to the augmented datasets. If we compare Unified-AM's performance between the augmented corpus and the original corpus (see Table 4), the model has much higher token level accuracy, C-F1, R-F1, and F1 scores when we apply augmentation techniques on the training corpus. We have reached the highest component C-F1(100%) score of 71.35% where Eger et al. [2] has obtained 70.83%.

We present the token level improvements below and compare them with the original PE corpus results. In the test set, we have in total 2,134 major claim tokens, 4,238 claim tokens, 13,728 premise tokens, and 9,437 non-argumentative tokens. Our goal is to increase the major claim tokens which can be considered as the root of the argumentation structure. The results provided below show the overall token level improvements that we get compared to the original paragraph version of the PE corpus. These results indicate that the augmentation technique has significantly improved the previous predictions regarding the major claim, claim, and premise tokens. For the original corpus (Paragraph level) we see Correct Major Claim Tokens: 1542; Correct Claim Tokens (with Stance:

Table 4
Experiment on the Augmented Corpus with Unified-AM

Original Paragraph Corpus							
Model	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	F1 (100%)	F1 (50%)
Unified-AM	66.79%	68.88	78.22	51.14	56.41	60.00	67.32
Augmented Corpus (Addition of New Paragraphs)							
Model	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	F1 (100%)	F1 (50%)
Unified-AM	67.02%	71.03	79.82	52.50	58.25	61.77	69.04
Augmented Corpus (Addition of New Paragraphs) Training Epochs: 1500-1600							
Model	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	F1 (100%)	F1 (50%)
Unified-AM	**68.03%**	**71.35**	**80.21**	**54.27**	**59.46**	**62.81**	**69.83**

For, Against): 2057; Correct Premise Tokens (with Stance: Support, Attack and Distance -11 to + 11): 7329; and Correct Non-Argumentative Tokens: 8217. For the Augmented Corpus (Addition of New Paragraphs) these numbers increase to 1633, 2287, 7612, 8266, respectively. Given extra training (Training Epochs: 1500-1600) the numbers for Claim and Premise Tokens increase and for Major Claim and Non-Argumentative Tokens decrease: 1597, 2344, 7956, 8196, respectively.

6. Error Analysis

We have measured the distance prediction accuracy of the Unified-AM model and compare it with that of Eger et al. [2]. Also, we compare our results with the works [13] which do not consider subtask 1 while solving the other subtasks related to argumentation mining.

We observe a higher accuracy of predicting longer distance in the paragraphs. One of the key strategies that we have followed for all of these experimental setups: We ensure the models share all of their learned parameters while solving any particular subtask (component detection and labelling, relation classification, or accurate distance prediction) of the main Argumentation Mining problem. This denser representation of the whole argumentation task enables our neural models to share all of the parameters while making predictions for each of the subtasks which has led to a high performance. Eger et al. [2] showed that LSTM-ER model's probability of correctness given true distance is below 40% and it becomes below 20% when the distances are larger than 3. But in our case, our analysis shows above 50% accuracy for distances 1, 2, and 3 (for Unified-AM). Our final model has higher accuracy regarding smaller distances but its prediction accuracy declines as we observe larger distance values in the PE corpus.

Recalling that subtask 1 of the argumentation mining problem is the separation of the argumentative text from the non-argumentative text, we compare Unified-AM's performance with some of the recent works where the output of subtask 1 of the argumentation problem has already been obtained. We look for 100% accuracy of span detection (successful segmentation of the argumentative text from the non-argumentative text) by

Table 5
Comparison between Unified-AMs with other models (which do not consider subtask 1)

Method	**Argument Component Type Classification**			
	Macro	MC	Claim	Premise
Joint-ILP [1]	82.6	89.1	68.2	90.3
St-SVM-full [14]	77.6	78.2	64.5	90.2
Joint-PN [11]	84.9	89.4	73.2	92.1
Span-LSTM [12]	87.3	-	-	-
Span-LSTM-Trans [13]	87.5	**93.8**	76.4	92.2
BERT-Trans [13]	88.4	93.2	**78.8**	93.1
Unified-AM (Ours)	**89.18**	92.30	78.25	**96.98**

Unified-AM and only on those spans do we measure F1 values for individual component type identification to compare our results with the other models (which assume subtask 1 is given). Lastly we calculate the macro-F1 score. Table 5 contains the results. We obtain the highest macro-F1 score of 89.18% when we do not consider subtask 1. We have the highest individual F1 score for the premise tokens (96.98%) which boosts the macro F1 score considerably.

7. Conclusions and Future Work

In this work, we show that rather than using a complex stacked architecture for a problem having different subtasks (where all the subtasks are related), we can have a compact and unified representation of all the sub-problems and can tackle it as a single problem with less complicated architectures. We obtain an improved performance over Eger et al. [2] in recognizing the argument components and relations. We further improve this result by introducing the Flair stacked embedding [18] to represent the text input. We introduce a multi-head attention layer to the neural architecture which leads to the highest accuracy on the PE corpus. Observing that the imbalanced corpus may be creating problems for this model to learn certain underrepresented features of the corpus, we have used the standard technique of data augmentation to achieve further gains in performance. We have created one augmented version of the PE training corpus by using different combinations of the n-grams that occur immediately before approximately two-thirds of the major claim components (see Section 5.2) in the paragraph version of the corpus. By using the augmentation methodology, we further improve the Unified-AM model's performance on the test set. We have obtained the highest token level accuracy, C-F1, R-F1, and the global F1 score (which is the combination of both C-F1 and R-F1 scores) on the paragraph version of the PE corpus by applying the augmentation technique. Shared parameter values across different subtasks enhanced the accuracy score and also the model's capability for accurate detection of components, relations and distance.

Future work includes applying Unified-AM on the essay level of the PE corpus, using contextual embeddings to enhance the representations of the argumentative texts, and testing an appropriately modified model on other datasets (e.g., the CDCP dataset [14]).

Acknowledgments

This research is partially funded by The Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant to Robert E. Mercer. We also acknowledge the helpful comments provided by the reviewers.

References

[1] C. Stab, I. Gurevych, Parsing argumentation structures in persuasive essays, Computational Linguistics 43 (2017) 619–659.

[2] S. Eger, J. Daxenberger, I. Gurevych, Neural end-to-end learning for computational argumentation mining, in: Proc. of 55th Ann. Meet. of Assoc. for Comp. Ling. (Vol. 1: Long Papers), 2017, pp. 11–22.

[3] R. M. Palau, M.-F. Moens, Argumentation mining: the detection, classification and structure of arguments in text, in: Proc. of the 12th International Conference on Artificial Intelligence and Law, 2009, pp. 98–107.

[4] C. Stab, I. Gurevych, Annotating argument components and relations in persuasive essays, in: Proc. of the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 1501–1510.

[5] M. Miwa, M. Bansal, End-to-end relation extraction using LSTMs on sequences and tree structures, in: Proc. of the 54th Ann. Meet. of the Assoc. for Comp. Ling. (Vol. 1: Long Papers), 2016, pp. 1105–1116.

[6] I. Persing, V. Ng, End-to-end argumentation mining in student essays, in: Proc. of the 2016 Conf. of the N. American Chap. of the Assoc. for Comp. Ling. Human Language Technologies, 2016, pp. 1384–1394.

[7] A. Ferrara, S. Montanelli, G. Petasis, Unsupervised detection of argumentative units though topic modeling techniques, in: Proceedings of the 4th Workshop on Argument Mining, 2017, pp. 97–107.

[8] I. Persing, V. Ng, Unsupervised argumentation mining in student essays, in: Proceedings of the 12th Language Resources and Evaluation Conference, 2020, pp. 6795–6803.

[9] A. Peldszus, Towards segment-based recognition of argumentation structure in short texts, in: Proceedings of the First Workshop on Argumentation Mining, 2014, pp. 88–97.

[10] A. Peldszus, M. Stede, Joint prediction in MST-style discourse parsing for argumentation mining, in: Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 938–948.

[11] P. Potash, A. Romanov, A. Rumshisky, Here's my point: Joint pointer architecture for argument mining, in: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing, 2017, pp. 1364–1373.

[12] T. Kuribayashi, H. Ouchi, N. Inoue, P. Reisert, T. Miyoshi, J. Suzuki, K. Inui, An empirical study of span representations in argumentation structure parsing,

in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4691–4698.

[13] J. Bao, C. Fan, J. Wu, Y. Dang, J. Du, R. Xu, A neural transition-based model for argumentation mining, in: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conf. on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6354–6364.

[14] V. Niculae, J. Park, C. Cardie, Argument mining with structured SVMs and RNNs, in: Proc. of the 55th Annual Meeting of the Assoc. for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 985–995.

[15] M. Ahmed, M. R. Samee, R. E. Mercer, Improving neural sequence labelling using additional linguistic information, in: 2018 17th IEEE Int. Conf. on Machine Learning and Applications, 2018, pp. 650–657.

[16] M. T. Sazid, A Unified Representation and Deep Learning Architecture for Persuasive Essays in English, MSc Thesis, The University of Western Ontario, London, Ontario, Canada, 2022. Electronic Thesis and Dissertation Repository. 8497. https://ir.lib.uwo.ca/etd/8497.

[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 260–270.

[18] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, Flair: An easy-to-use framework for state-of-the-art nlp, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), 2019, pp. 54–59.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[20] J. Ho, N. Kalchbrenner, D. Weissenborn, T. Salimans, Axial attention in multidimensional transformers, arXiv preprint arXiv:1912.12180 (2019).

[21] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations, 2020, p. 12pp.