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

Muhammad Tawsif Sazid and **Robert E. Mercer**
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. Also, we have introduced a decoupled solution to identify the entities and relations first, and on top of that a second model is used to detect distance between the detected related components. An augmentation of the corpus (paragraph version) by including copies of major claims has further increased the performance.

1 Introduction

Arguments are composed of statements, called claims, that take a position on a controversial subject and other statements, referred to as premises, that support or rebut the claims. When arguments are presented in text form, these argument components are realized as contiguous text spans. The writing also contains non-argumentative text spans. The argument and non-argumentative text spans are collectively referred to as argumentative discourse units (ADUs). Argumentation mining is usually viewed as the identification of argumentative structures: separating the argumentative ADUs from the non-argumentative ADUs, classifying the argumentative ADUs as premises and claims, and finding the relationships among the argumentative ADUs. Since we are using the Persuasive Essay (PE) dataset (Stab and Gurevych, 2017) these subtasks can be made more precise: 1) segment the argument components from the non-argumentative text, 2) label each argument component as a Major-Claim, Claim, or Premise, 3) determine which premises are in a relationship with claims or premises using a text distance measure, and 4) classify the stance of the relations between argument components.

Since we are using the Persuasive Essay (PE) dataset (Stab and Gurevych, 2017) we will use the description of these tasks as given by Eger et al. (2017): 1) segmenting the ADUs: separate the argumentative text spans from the non-argumentative text, 2) labeling each argument component as a Major-Claim, Claim, or Premise, 3) determining which premises are in a relationship with claims or premises and representing this relation 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 using the text distance method), and 4) classifying the stance of the relations between argument components ('for' and 'against' for the relationship between claims and major-claims; 'support' and 'attack' between premises and claims or other premises).

Previous research has approached the development of a computational argumentation mining method from two distinct viewpoints. Input for the first approach is plain text and this approach solves all four of the subtasks mentioned above. Stab and Gurevych (2017) provide the PE dataset, which we use in the development of our method. Eger et al. (2017) produce the state-of-the-art method to which we compare our new method. Recently, Persing and Ng (2020), using the PE dataset, have developed an unsupervised machine learning method that provides all but the stance information for the

Proceedings of the 9th Workshop on Argument Mining, co-located with the 29th International Conference on Computational Linguistics, pages 74–83
October 17, 2022.

relations.

The second view of argumentation mining assumes the first subtask has been done (Peldszus, 2014; Peldszus and Stede, 2015; Stab and Gurevych, 2017; Niculae et al., 2017; Potash et al., 2017; Kuribayashi et al., 2019; Bao et al., 2021).

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 with a unified representation. Our research contributions are summarized as follows:

1. Each token in the natural language text is encoded as a binary vector that captures all aspects of the argumentation mining task: the ADU type, the position in the argument component text span, the stance of the argument component, and the distance to the related argument component. The deep learning model computes a vector for each token which, when properly interpreted, provides the information required to assemble the text spans and relations thereby identifying the argument structure for the argument mining task. By combining all aspects of the argumentation mining task in this representation, a model is learned that has improved performance.

2. By constructing a 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, 3 linear layers with ReLU activation and 1 final linear layer (Unified-AM)[1].

3. We introduce a joint model (Decoupled-AM) approach[2]. We train both Unified-AM and a second model composed of a normalization layer, two biLSTM layers, three linear layers with Dropout and ReLU activations, and one final linear layer. While Unified-AM is detecting components and relations, the second model detects distances between the related components using different layer outputs provided by Unified-AM. In this setting, we have trained both models together from scratch.

4. Our previous work (Sazid and Mercer, 2022) only worked with the paragraph version of the PE dataset. Here we also test our novel representation and model on the essay version.

5. We develop an augmentation technique (paragraph version) based on the n-gram tokens that indicate the starting of the major claim tokens[3]. This further improves the results.

With the new formulation of the problem, our original Unified-AM and the Decoupled-AM reach state-of-the-art argument mining performance on detecting and labelling argument components and relations for the PE corpus.

2 Related Work

Computational argumentation mining deals with finding argumentation structures in text. Palau and Moens (2009) established that argument mining would need to detect claims and premises and their relationships. Stab and Gurevych (2014, 2017) provided the PE dataset, a corpus annotated with a scheme that includes claims, premises, and also attack or support relations. Stab and Gurevych (2017) addressed the argumentation problem by training independent models for each of the subtasks and then combining them with an Integer Linear Programming Model for the end-to-end task. Eger et al. (2017) 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, introduced by Miwa and Bansal (2016), which uses a stacked architecture of Sequence and Tree LSTMs.

Persing and Ng (2016) 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. (2017) introduced an unsupervised approach, topic modeling, to detect claims and premises. Persing and Ng (2020) have also developed an unsupervised machine learning method that provides all but the stance information for the relations.

[1] Unified-AM code is available at https://github.com/tawsifsazid/Unified-Representation-for-Argumentation-Mining.

[2] Decoupled-AM code is also available at https://github.com/tawsifsazid/Unified-Representation-for-Argumentation-Mining.

[3] The augmented dataset is available at https://github.com/tawsifsazid/Unified-Representation-for-Argumentation-Mining.

A number of works have investigated approaches for subtasks 2, 3, and 4. Early work is epitomized by Peldszus (2014) and Peldszus and Stede (2015) where they develop a novel methodology for predicting argument structure by dividing it into different sub-tasks (relation, central claim, role, and function classification). Potash et al. (2017) presented the first neural network-based approach to argumentation mining, focusing on extracting links between argument components and classifying types of argument components as a secondary goal. Niculae et al. (2017) jointly approach unit type detections and relation predictions on their new CDCP dataset and the PE dataset. Kuribayashi et al. (2019) focuses on Argumentation Structure Parsing (ASP). Their analysis of other works regarding the span representation led them to the development of a simple task-dependent addition for the ASP. Bao et al. (2021) avoid previous inefficient enumeration operations for detecting relational attributes. For that, they introduce a transition-based methodology that follows an incremental procedure for building graphs based on argumentation.

We note from Ahmed et al. (2018) how additional handcrafted features can boost the accuracy on certain sequence tagging tasks. Kuribayashi et al. (2019) and Persing and Ng (Persing and Ng, 2020) also noted the importance of discourse connectives in the argumentation mining task.

3 Research Methodology

Here we present the method that we have developed to generate the argumentation structure for the PE data set. First, the data set is described. Then, we introduce the multi-label representation that allows us to consider 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 2 the performance increase given by that 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. (2017).

3.1 Data Set Description

The PE dataset that we are using in this paper was created by Stab and Gurevych (2017) and was used in Eger et al. (2017). 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, the type of components, stances, and distances from premise to claim or premise (Eger et al., 2017; Stab and Gurevych, 2017). There are essay and paragraph versions of the data set. We have worked with both versions 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[4]. The development set has 12,657 tokens available in 199 paragraphs. In the essay version of the corpus, there are 285 essays in the train-set. The development and the test set have 35 and 79 essays, respectively.

The argumentation structure can be viewed as a forest with each tree rooted by one of the author's major claims. 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. The corpus is imbalanced as Eger et al. (2017) have 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 (first described in Sazid and Mercer (2022)). 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. The maximum and minimum distances from premise to claim suggested in Eger et al. (2017) are +11 and -11, respectively. Thus, we have constructed a dense unified representation of the argumentation mining problem. Table 1 provides the novel representation.

By formulating the argumentation mining task as a multi-label problem, we have enabled the options to solve the argumentation problem in a unified or in a decoupled way. We have tried both strategies

[4] Differs slightly from that reported in Eger et al. (2017).

Token	O	B	I	MC	C	P	Sup	For	At	Ag	Distance Value -11	...	Distance Value 3	...	Distance Value +11
For	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
instance	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
,	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
children	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0
immigrated	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
to	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
a	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
new	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
country	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0

Table 1: Example of the Novel Compact Representation of the Argumentation Problem. O: Non-Argumentative Token, B: Beginning of Argument Component, I: Continuation of Argument Component, MC: Major Claim Component, Cl: Claim Component, P: Premise Component, Sup: Support Relation Identifier, For: For Relation Identifier, At: Attack Relation Identifier, Ag: Against Relation Identifier, Distance Values: -11 to +11. The sentence being encoded "For instance, children immigrated to a new country . . . " has three introductory non-argumentative tokens, the premise starting with "children" supports an argument component three sentences later in the paragraph.

and compare our results related to the experiments.

3.3 Interpretation Function for the Multi-label Outputs of the Model

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 an interpretation function.

For each token, this function first decides whether the token is to be considered non-argumentative or part of an argument component. If it is to be considered argumentative, the beginning and continuation designations are determined. Then, depending on the argument component type, the stance is determined, and if it is a premise, the distance is as well.

3.4 Description of the Deep Learning Model and the Hyper-Parameters

Figure 1 represents our final argumentation model architecture (Unified-AM) which we have created for detecting argumentation structures and solve all the subtasks jointly.

We have also developed a decoupled model (Decoupled-AM) (research contribution 3) where we first predict the components and relations. Then we detect the distance between the predicted components based on it. For this methodology, we have used two models. The first model is identical to the Unified-AM and we introduce a second model which predicts the distances. In this particular experiment, we have trained both models from scratch. In this experiment, Unified-AM is used to predict the first 10 labels and the loss is calculated for those 10 labels only. The second model predicts the last 23 labels (distances) and the loss is calculated for only these last 23 labels. After that, the two loss values are summed and then this summed loss value is used to calculate the gradients to initialize the back propagation for both models simultaneously. The components and stances (first 10 labels) predicted by Unified-AM and the distances (last 23 labels) predicted by the second model are concatenated to finally produce the 33-labelled output.

Our deep learning model architecture includes: stacked embedding, axial positional embedding, a multi-head attention layer, a 2-layered biLSTM, 3 linear layers with dropouts and ReLU activations 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 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 two biLSTM

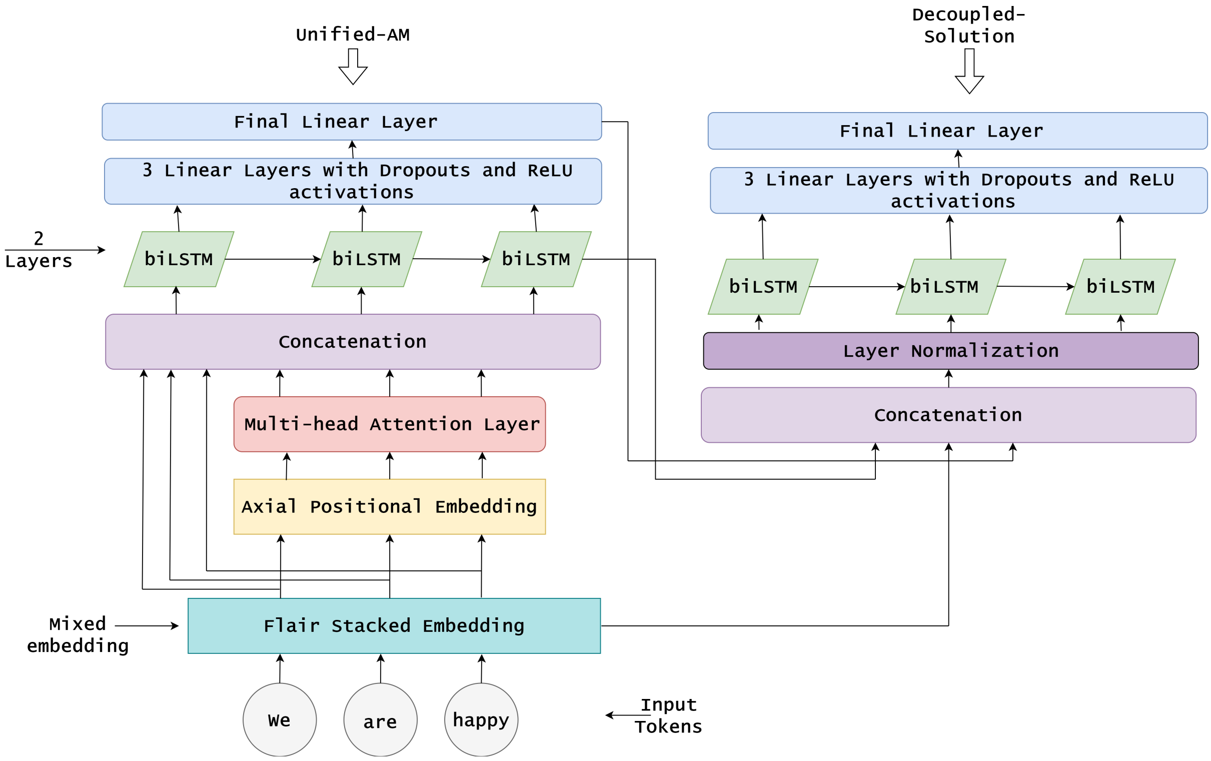

Figure 1: The Final Argumentation Mining Model Architecture with Decoupled Distance Prediction

layers, 3 linear layers with non-linear activation functions, and one final linear layer achieve the best accuracy.

Figure 1 includes a mixed embedding but in the model design we first experimented with a plain embedding layer instead. Lample et al. (2016), 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. (2019) 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 (Vaswani et al., 2017) and the axial positional embedding for the positional information (Ho et al., 2019; Kitaev et al., 2020). For our 400-dimension embedding class we use four heads for the multi-head attention layer for both of the experiments. This completes the description of the architecture.

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 2 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**. This model further reduces the error rate by **3.69%**.

After trying several hyperparameter values for each of the different components we have chosen the final values. We use dropout values of 0.5 for the linear layers, 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 use

Table 2: Error Analysis and Comparison of 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 ReLU activation function in-between the linear 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 Tables 4, 6). For determining the default training epochs (1000-1100) we have closely observed the development set accuracy value after every 5 epochs. If after some 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 (paragraph version) by augmenting the data in our augmentation experiments (see Section 4.2), 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.

4 Experiments and Results

4.1 Experiments on the original version of the PE corpus

We have experimented with the new unified-representation of all of the sub-tasks of argumentation mining and trained our final model architectures. In one of our experiments, we have only trained the original Unified-AM (Sazid and Mercer, 2022) to jointly solve all of the sub-tasks (all 33 labels) of argumentation mining. In Table 3, we present individual precision, recall and F1 score for the four ADUs and the four relations that are available in the PE corpus (both paragraph and essay versions) for the original Unified-AM model. We observe low precision and recall scores for the claim tokens even though the class is not the least frequent one in the PE corpus. This is similar to the observed low agreement score among the human annotators for the claim tokens (Stab and Gurevych, 2017). Unified-AM also finds it difficult to predict the claim tokens in the corpus.

And in the other experimental setup, we have used a second model to detect the distance values (the last 23 labels) separately by using information about the components and relations (the first 10 labels) from the Unified-AM.

With the original Unified-AM, we achieve a token level accuracy of 66.79% in our argumentation mining task. On the other hand, the Decoupled-AM achieves the highest token level accuracy of **67.50%**. Also, we have improved C-F1, R-F1 scores regarding the task with Decoupled-AM.

Table 4 summarizes the result for these experiments, including the F1 measure for the component and relation tasks, and a global F1 score. The results from Eger et al. (2017) have been included for comparison. Now, compared to the Eger et al. (2017) decoupled method for computing the relation identification, this task in our original Unified-AM and Decoupled-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. (2017) in the PE data set.

We have also experimented on the essay-level of the argumentation corpus and our original Unified-AM model has achieved the highest token level accuracy, C-F1, and R-F1 scores. The experiments on the essay version of the corpus show the robustness of the unified representation of all the subtasks with our model. When we experiment on the essay version of the corpus, our scores and results have not decreased like the LSTM-ER model and its decoupled solution. Our Decoupled-AM has not performed well on the essay version of the corpus compared to the original Unified-AM. Table 5 summarizes the results for the experiments related to the essay version of the argumentation corpus.

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

Class	**Paragraph Level**			**Essay Level**			**Token**
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Percentage**
Non-Argumentative	88.38	88.27	88.33	89.26	91.27	90.25	32.20
Major-Claim	73.87	74.18	74.02	70.34	72.05	71.19	7.41
Claim	65.37	58.05	61.48	56.11	49.34	52.51	15.41
Premise	88.01	90.87	89.42	87.18	88.32	87.74	44.99
Support	86.79	89.69	88.22	85.09	88.35	86.69	42.61
For	60.96	57.05	58.94	56.41	50.76	53.43	12.77
Attack	32.52	26.77	29.37	27.08	7.10	11.25	2.38
Against	60.81	29.97	40.15	21.01	10.23	13.76	2.64

Table 4: Comparison of LSTM-ER (Eger et al., 2017), Unified-AM, and Decoupled-AM on the Paragraph Level

Model	Corpus	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	F1 (100%)	F1 (50%)
LSTM-ER	Original	61.67%	70.83	77.19	45.52	50.05	55.42	60.72
Unified-AM	Original	66.79%	68.88	78.22	51.14	56.41	60.00	67.32
Decoupled-AM	Original	**67.50%**	**71.24**	**79.98**	**52.71**	**57.92**	**61.97**	**68.95**
Unified-AM	Augmented	**68.03%**	**71.35**	**80.21**	**54.27**	**59.46**	**62.81**	**69.83**
Decoupled-AM	Augmented	65.53%	68.59	77.94	50.22	56.24	59.41	67.10

4.2 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 (Ahmed et al., 2018). We have observed (as did Kuribayashi et al. (2019), and Persing and Ng (2020)) 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 paragraph version of 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 paragraph level PE dataset. Below, we describe the augmentation technique that we have used to augment the PE corpus. We also compare the performance between Unified-AM, Decoupled-AM on both 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 individual's 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 compe-

Table 5: Comparison of LSTM-ER (Eger et al., 2017), Unified-AM, and Decoupled-AM on the Essay Level

Model	Corpus	Token Accuracy	C-F1 (100%)	C-F1 (50%)	R-F1 (100%)	R-F1 (50%)	F1 (100%)	F1 (50%)
LSTM-ER	Original	54.17%	66.21	73.02	29.56	32.72	40.87	45.19
Unified-AM	Original	**62.88%**	**67.78**	**76.20**	**48.24**	**52.49**	**58.01**	**64.35**
Decoupled-AM	Original	57.89%	64.67	75.51	40.01	46.10	52.34	60.75

Table 6: Token level Comparison between the Original and the Augmented Datasets

Model	**Corpus (Paragraph Version)**	**Correct Major-Claim Tokens**	**Correct Claim Tokens (with Stance)**	**Correct Premise Tokens (with Stance)**	**Correct Non-Argumentative Tokens**
Unified-AM	Original	1542	2057	7329	8217
Unified-AM	Augmented	1597	2344	**7956**	**8196**
Decoupled-AM	Original	**1595**	**2397**	**7720**	**8226**
Decoupled-AM	Augmented	1594	2355	7334	8074

tition , 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 individual's 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 4-gram "*I firmly believe that*" with an equal size randomly chosen 4-gram "*I truly believe that*" from our collected n-gram list. The words following in that particular sentence are major claim tokens.

By using data augmentation, we have increased the number of Major Claim tokens by approximately 4000. Also, because claims, premises, 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 first (see Figure 1) on the corpus. We have achieved the highest token level accuracy on the paragraph-level argumentation corpus. Previously, without augmentation, we have achieved 67.50% token level accuracy on the PE dataset (see Table 4) 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. Table 4 shows the results related to the augmented datasets. Comparing 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. (2017) has obtained 70.83%. After training Unified-AM, we move on to the next experimental setup and train our Decoupled-AM on the augmented corpus. Decoupled-AM has not performed well on the augmented corpus compared to Unified-AM. The reason is: Decoupled-AM needs more training time compared to the Unified-AM when we are experimenting for the augmented corpus. Training both the models together from scratch on a larger corpus needs sufficient amount of time and resources.

We present in Table 6, the token level improvements and compare them with the original PE corpus results. In the test set, we have 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 in Table 6 show the overall token level improvements that we get compared to the original paragraph version of the PE corpus for both Unified-AM and Decoupled-AM.

5 Error Analysis

We have done some error analysis and comparison between various neural architectures to see how dif-

Table 7: F1 scores on the BIO labeling task

	STag_BLCC	LSTM-ER	ILP	HUB	Unified AM	Unified-AM Augmented Paragraph	Decoupled-AM	Decoupled-AM Augmented Paragraph
Essay	90.04	**90.57**	-	-	90.52	-	89.99	-
Paragraph	88.32	**90.84**	86.67	88.60	89.69	89.88	90.30	89.11

ferently all of the models perform on the argumentation task. Also, we have measured the distance prediction accuracy of the Unified-AM model and compare it with that of Eger et al. (2017).

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. (2017) 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 (see Figure 1) has higher accuracy regarding smaller distances but its prediction accuracy declines as we observe larger distance values in the PE corpus.

For major-claim, premise, and claim, there are two different tags in the PE corpus, B: Beginning of a component and I: Continuation of a component. Non-Argumentative tokens are tagged as 'O' in the BIO scheme. We compare the component segmentation task (subtask 1) results with other works that have been mentioned in Eger et al. (2017). Table 7 shows the results for the models. We see that LSTM-ER has the highest macro-F1 score when we consider only the BIO labeling task.

6 Conclusions and Future Work

In this work, we show that rather than using a complex stacked architecture for a problem which has different subtasks (where all the subtasks are related to each other), 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. (2017) in recognizing the argument components and relations. We further improve this result by introducing the Flair stacked embedding (Akbik et al., 2019) to represent the text input. We introduce a multi-head attention layer to the neural architecture which leads us 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 4.2) 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. We have obtained better results on the original essay version of the corpus. Shared parameter values across different subtasks enhanced the accuracy score and also the model's capability for accurate detection of components, relations and distance. Our work has shown a robust method which jointly solves the component and relation identification tasks on the essay and paragraph levels of the Persuasive Essays corpus.

Future work includes a modified, yet unified, representation for other corpora and using contextual embeddings to enhance the representations of the argumentative texts.

Acknowledgements

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

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E. Mercer. 2018. Improving neural sequence labelling using additional linguistic information. In *2018 17th IEEE Int. Conf. on Machine Learning and Applications*, pages 650–657.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. 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)*, pages 54–59.

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. 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)*, pages 6354–6364.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proc. of 55th Ann. Meet. of Assoc. for Comp. Ling. (Vol. 1: Long Papers)*, pages 11–22.

Alfio Ferrara, Stefano Montanelli, and Georgios Petasis. 2017. Unsupervised detection of argumentative units though topic modeling techniques. In *Proceedings of the 4th Workshop on Argument Mining*, pages 97–107.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. 2019. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*, page 12pp.

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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. 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*, pages 260–270.

Makoto Miwa and Mohit Bansal. 2016. 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)*, pages 1105–1116.

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

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proc. of the 12th International Conference on Artificial Intelligence and Law*, pages 98–107.

Andreas Peldszus. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97.

Andreas Peldszus and Manfred Stede. 2015. Joint prediction in MST-style discourse parsing for argumentation mining. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948.

Isaac Persing and Vincent Ng. 2016. 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*, pages 1384–1394.

Isaac Persing and Vincent Ng. 2020. Unsupervised argumentation mining in student essays. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6795–6803.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, pages 1364–1373.

Muhammad Tawsif Sazid and Robert E. Mercer. 2022. A unified representation and deep learning architecture for argumentation mining of students' persuasive essays. In *to appear*, CEUR Workshop Proceedings.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proc. of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.