Discourse Self-Attention for Discourse Element Identification in Argumentative Student Essays

Wei Song[1], **Ziyao Song**[1], **Ruiji Fu**[2,3], **Lizhen Liu**[1], **Miaomiao Cheng**[1], **Ting Liu**[4]
[1]College of Information Engineering and Academy for Multidisciplinary Studies,
Capital Normal University, Beijing, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
[3]iFLYTEK AI Research (Hebei), Langfang, China
[4]Research Center for Social Computing and Information Retrieval,
Harbin Institute of Technology, Harbin, China
{wsong, zysong, liz_liu7480, B365}@cnu.edu.cn,
rjfu@iflytek.com, tliu@ir.hit.edu.cn

Abstract

This paper proposes to adapt self-attention to discourse level for modeling discourse elements in argumentative student essays. Specifically, we focus on two issues. First, we propose structural sentence positional encodings to explicitly represent sentence positions. Second, we propose to use inter-sentence attentions to capture sentence interactions and enhance sentence representation. We conduct experiments on two datasets: a Chinese dataset and an English dataset. We find that (i) sentence positional encodings can lead to a large improvement for identifying discourse elements; (ii) a structural relative positional encoding of sentences shows to be most effective; (iii) inter-sentence attention vectors are useful as a kind of sentence representation for identifying discourse elements.

1 Introduction

Discourse describes how a document is organized. This paper focuses on the task of discourse element identification (DEI) in argumentative student essays. Discourse elements represent the function and contribution of every discourse unit to the discourse. Burstein et al. (2003) formulate discourse elements as 5 categories: *introduction*, *thesis*, *main idea*, *supporting* and *conclusion*, while argument components such as *major claim*, *claim* and *premise* are used as discourse elements in argumentation structure parsing in persuasive essays (Stab and Gurevych, 2014). DEI can benefit automated essay scoring in many aspects: modeling organization, inferring topics and opinions or used as features for scoring systems (Attali and Burstein, 2006; Burstein et al., 2001; Persing et al., 2010; Song et al., 2020).

Despite its importance, DEI is challenging.

First, the ambiguity of sentences makes learning models difficult to distinguish some discourse elements. For example, the *thesis* is defined as expressing the central claim of the author and the *main ideas* support the *thesis* from specific aspects. However, it is hard to distinguish them from their content and style.

Second, the discourse element of a specific sentence depends on context. As a result, considering individual sentences only would have difficulties in identifying discourse elements. The relations and relatedness among multiple sentences should be explored.

Third, the data imbalance problem is serious, e.g., the number of *elaboration* sentences could be 10 times more than the number of *thesis* sentences. The minority discourse elements (such as *thesis*, *main ideas* or *major claim*) are harder to be recalled although they have important roles in many scenarios, e.g., evaluating the organization of essays (Attali and Burstein, 2006).

In this paper, we propose a method to explicitly model sentence positions and relations to improve discourse element identification in argumentative student essays. Our idea is partially motivated by the self-attention mechanism such as (Vaswani et al., 2017). Self-attention is usually applied to capture dependencies between words. We aim to apply self-attention mechanism to describe relations between sentences.

On one hand, position information is important for DEI to give clues on discourse elements beyond content and style, because authors usually hold some conventions to organize content. Position is one of the most useful feature classes in feature-based DEI (Burstein et al., 2003; Stab and Gurevych, 2014). Previous neural network models usually cast DEI as a classification or sequence labeling task and do not explicitly model position information. Motivated by the positional encoding of words, we propose a simple structural positional encoding strategy for a sentence by considering its

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2820–2830,
November 16–20, 2020.

relative position in its essay, relative position of its paragraph in its essay, and its relative position within its paragraph.

On the other hand, relatedness among sentences may also indicate properties of discourse elements. For example, *thesis* sentences should have close relations to the whole essay; main ideas usually locate in similar positions and have high relatedness. Relatedness between discourse elements has shown to be an important indicator of essay coherence (Higgins et al., 2004). We compute inter-sentence attention vectors to represent either element-wise or content-wise relations to other sentences, which bring in additional information beyond individual sentences and enhance sentence representation without extra information.

Experiments show that the proposed approach can get considerable improvements compared with feature-based and neural network based baselines on a Chinese dataset and obtain competitive results compared with the state-of-the-art method on an English dataset. The structural positional encodings of sentences show effectiveness to achieve obvious overall improvements. The inter-sentence attention vectors enhance sentence representation helping identify discourse elements as well.

2 Related Work

2.1 Discourse Element Identification

DEI could be seen as a subtask in discourse structure analysis. It aims to identify discourse elements, determine their functions and establish relationships among them in an argumentative text.

Typical tasks in this line include argumentative zoning (Teufel et al., 1999) , argumentation mining (Mochales and Moens, 2011; Lippi and Torroni, 2016) and analyzing argumentative student essays (Burstein et al., 2003; Stab and Gurevych, 2014). Argumentative zoning identifies arguments in scientific articles (Teufel et al., 1999; Guo et al., 2010). Argumentation mining aims to identify argument components and relations from legal texts (Palau and Moens, 2009; Mochales and Moens, 2011) or argumentative texts (Stab and Gurevych, 2014; Daxenberger et al., 2017).

The solutions to these tasks usually adopt similar machine learning methods but use domain related features. The methods could be roughly classified into the following categories.

Classification based methods cast DEI as a classification problem. Various classifiers have been tested, such as SVM (Stab and Gurevych, 2014), decision trees (Burstein et al., 2003, 2001) and naive Bayes, maximum entropy model (Moens et al., 2007; Palau and Moens, 2009).

Sequence labeling based methods exploit contextual information for DEI with conditional random fields (Hirohata et al., 2008; Song et al., 2015) or recurrent neural networks (Daxenberger et al., 2017).

Establishing relations between sentences is often viewed as a classification tasks as well (Stab and Gurevych, 2014). **Parsing based methods** are also adopted to build more complex structures with techniques like ILP (Stab and Gurevych, 2017) or RST style parsing (Peldszus and Stede, 2015).

Feature engineering. Some common features are shared across these tasks, including syntactic, lexical, semantic and discourse relations. There are also domain related features to further boost the performance. Mochales and Moens (2011) designed special features for argumentation mining in legal texts. Nguyen and Litman (2015) identified claims based on domain words. Lippi and Torroni (2015) modeled syntactic structures for content independent claim detection based on tree kernels.

Our work is mostly related to DEI in argumentative student essays (Burstein et al., 2003; Stab and Gurevych, 2014), which is useful for qualifying essay organization (Persing et al., 2010), argumentation (Persing and Ng, 2016; Wachsmuth et al., 2016) and general writing (Burstein et al., 2003; Ong et al., 2014; Song et al., 2014). The major feature classes proposed by Burstein et al. (2003) and Stab and Gurevych (2014) are used to build a baseline. The features include: position, cue words, lexical features (main verbs, adverbs and connectives) and structural features (such as number of clauses). Some of these features are based on manually collected lexicons.

Deep Learning Methods have achieved great success in many NLP tasks. Eger et al. (2017) proposed neural argumentation mining models based on sequence tagging or dependency parsing. It exploits inter-sentence relations but needs sophisticated language processing. Daxenberger et al. (2017) exploited CNN and LSTM for classifying sentences to identify claims from different domains. It mainly depends on the content of components but does not sufficiently model positions and exploit inter-sentence relatedness.

2.2 Attention Mechanism for Discourse Representation

Attention mechanism was first introduced by (Bahdanau et al., 2015) in the encoder-decoder framework. Attention has the ability to learn important regions within a context and has been widely adopted in deep learning. Liu and Lapata (2018) proposed a structured attention mechanism to derive a tree over a text, akin to an RST discourse tree. Ferracane et al. (2019) evaluated the model, however, found multiple negative results. Attention mechanism has also been applied for RST parsing and its applications (Li et al., 2016; Ji and Smith, 2017; Huber and Carenini, 2019) but it is mostly used for capturing local semantic interactions.

2.3 Self-Attention Mechanism

Vaswani et al. (2017) proposed the self-attention mechanism and achieved state of the art results in many NLP tasks. Since then, self-attention has drawn increasing interests due to flexibility in modeling long range interactions.

Self-attention ignores word order in a sentence. As a result, position representations are developed to cooperate with self-attention. In addition to the sinusoidal position representation proposed by Vaswani et al. (2017), there are also other variations to bias the selection of attentive regions (Shen et al., 2018; Shaw et al., 2018; Yang et al., 2019). In NLP, self-attention is mostly applied to sequential structures such as a sequence of words. Mihaylov and Frank (2019) proposed a discourse-aware self-attention encoder for reading comprehension on narrative texts, where event chains, discourse relations and coreference relations are used for connecting sentences. Self-attention can be also extended to 2d-dimensions for image processing (Parmar et al., 2018) and lattice inputs (Sperber et al., 2019).

3 Baseline

We use Hierarchical BiLSTM (HBiLSTM), which is similar to (Yang et al., 2016), as the base model to model sentence and discourse level representations.

The task is to assign discourse element labels $\mathbf{y} = (y_1, ..., y_n)$ to sentences $(x_1, ..., x_n)$ in a text, where $x_i, 1 \leq i \leq n$, is a sentence of a sequence of words and $y_i \in \mathcal{Y}$, $\mathcal{Y}$ is a set of pre-defined discourse elements.

3.1 Sentence Representation Layer

A sequence of words $x = \{w_1, ..., w_N\}$ is modeled with a RNN encoder and is converted into a sequence of hidden states $\mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_N\}$. The hidden state at the i-th step is

$$\mathbf{h}_i = f\left(\mathbf{e}\left(w_i\right), \mathbf{h}_{i-1}\right), \quad (1)$$

where f is a RNN unit, $\mathbf{e}(w_i) \in \mathbb{R}^d$ is the embedding of a word, and $\mathbf{h}_{i-1}$ is the hidden state of the previous step. The whole sequence could be represented as a fixed length vector $\mathbf{c} = \phi(\{\mathbf{h}_1, \cdots, \mathbf{h}_N\})$ to represent the semantic of a sentence, where $\phi(\cdot)$ is a function to summarize hidden states.

In this work, Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is used as the RNN unit and the sequence is encoded in a Bi-directional way that a hidden state $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$ is the concatenation of the corresponding hidden states from both directions. The summarization function $\phi(\cdot)$ could be based on the attention mechanism.

3.2 Discourse Representation Layer

In the discourse element layer, we feed sentence representations $\mathbf{C} = (\mathbf{c}_1, ..., \mathbf{c}_n) \in \mathbb{R}^{d \times n}$ to a BiLSTM and use a nonlinear layer to map semantic representations to discourse element representations,

$$\mathbf{D} = \tanh(\text{BiLSTM}\left(\mathbf{C}\right)). \quad (2)$$

3.3 Inference Layer

Finally, we use a linear and a softmax layer to predict the discourse element of every sentence,

$$\mathbf{Y} = \text{softmax}\left(\text{linear}(\mathbf{D})\right), \quad (3)$$

where $\mathbf{Y} \in \mathbb{R}^{|\mathcal{Y}| \times n}$ refers to the probabilities of every sentence over discourse element categories.

The baseline mainly exploits interactions between adjacent sentences, but long distance interactions and sentence positions are not explicitly considered, which may be also important to determine the function of sentences in argumentative discourse.

4 Discourse Self-Attention

We propose the Discourse Self-Attention (DiSA) layer to improve the baseline by explicitly modeling sentence positions and inter-sentence interactions. The architecture is illustrated in Figure 1.

The sentences in an essay are converted to sentence embeddings $\mathbf{C}$ through the BiLSTM encoder introduced in Section 3.1, which are used as the input of DiSA. DiSA explicitly represents sentence positions, which are integrated with the content representations of sentences to get element representations. DiSA also has an inter-sentence attention module to get both element-wise and content-wise attention vectors of sentences to capture sentence interactions. The attention vectors and element representations are concatenated and fed to a linear layer and a softmax layer for prediction.

4.1 Sentence Positional Encodings (SPE)

Discourse elements in argumentative essays are sensitive to their positions. For example, *introduction* mostly comes before *thesis* or *main ideas* and *main ideas* may occur more often at the beginnings or endings of paragraphs.

Figure 2 shows an essay with 7 sentences and 4 paragraphs as an example. We consider three types of sentence positions for positional encoding.

- **Global position**: The index of a sentence is used to describe its position where we assume sentences in an essay form a sequence.
- **Paragraph position**: An essay has multiple paragraphs. The position of the paragraph that contains the sentence is also important.
- **Local position**: The position of the sentence in its paragraph is informative as well.

We adopt a relative positional encoding approach. We compute the relative positions for the above three position types. For example, the relative global position of the i-th ($i \geq 1$) sentence in an essay E is

$$pos_{global}(i) = \frac{i}{|E|}, \quad (4)$$

where $|E|$ is the number of sentences.

To integrate with sentence representations, we expand $pos_{global}(i)$ to a vector of the same dimension d of the distributed sentence representations by duplicating its value to every dimension, noted as $\mathbf{pos}_{global}(i) \in \mathbb{R}^d$. The relative paragraph position representation $\mathbf{pos}_{para}(i)$ and relative local position representation $\mathbf{pos}_{local}(i)$ are computed in the same way.

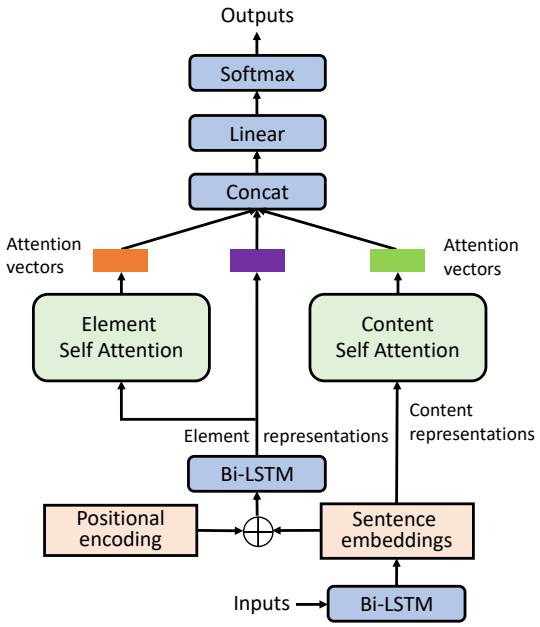

Figure 1: The architecture of Discourse Self-Attention.

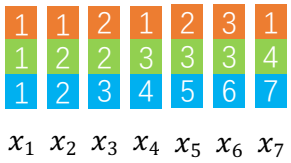

Figure 2: Three types of sentence positions for positional encoding.

The final position representation $\mathbf{pos}(i)$ is formulated as a liner combination of the three relative position representations, i.e.,

$$\mathbf{pos}(i) = \sum_{t \in \{global, local, para\}} \beta_t \mathbf{pos}_t(i), \quad (5)$$

where $\{\beta_t\}$ are parameters to be learnt in training. The element representation of the i-th sentence is

$$\mathbf{e}_i = \tanh(\text{BiLSTM}\,(\mathbf{C}_i + \mathbf{pos}(i))). \quad (6)$$

4.2 Inter-Sentence Attention (ISA)

Self-Attention relates elements at different positions by computing attention between every pair of elements. An attention function is to map a query and a set of key-value pairs to an output. The queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$ are vectors. We define $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{d_k \times n}$ and d_k is the dimension. The attention is computed as

$$\boldsymbol{\alpha} = \text{Attn}(\mathbf{Q}, \mathbf{K}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}). \quad (7)$$

The output is computed as a weighted sum of the values, i.e., $\boldsymbol{\alpha}\mathbf{V}$. Here, we are interested in the attention vectors rather than the weighted output, because an attention vector reflects the relatedness of a given sentence to every other sentence. We propose the inter-sentence attention (ISA) by applying self-attention to sentence semantic representations $\mathbf{C}$ and discourse element representations $\mathbf{E} = \{\mathbf{e}_i\}$.

- **Element Self-Attention (ElemSA)**: ElemSA models relations among discourse elements. We use $\mathbf{E}$ to get $\mathbf{Q}$ and $\mathbf{K}$, $\mathbf{Q} = \mathbf{E}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{E}\mathbf{W}^K$, where $\mathbf{W}^Q, \mathbf{W}^K \in \mathbb{R}^{d \times d_k}$. We do not use the normalized attention vectors as shown in Equation 7 to capture relative relatedness. Instead, we use $\boldsymbol{\alpha}_e = \tanh(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}})$ as attention vectors.

- **Content Self-Attention (ContSA)**: ContSA explores content relatedness to model sentence interactions. Similarly to ElemSA, we use the sentence semantic representations $\mathbf{C}$ to compute the ContSA vector $\boldsymbol{\alpha}_c$. The parameters are independent from ElemSA.

Adaptive Maxpooling Different essays have different number of sentences. To have a fixed-length attention vector, we borrow the idea of spatial pyramid pooling from image processing (He et al., 2015). It can maintain relatedness information by maxpooling $\boldsymbol{\alpha}_e$ and $\boldsymbol{\alpha}_c$ in local bins. These bins have sizes proportional to the number of an essay's sentences so that the number of bins is fixed regardless of the essay length. We set the number of bins to 1, 2, 4 and 8, respectively. The resulted representations can be seen as descriptions of the relatedness of a sentence to different zones of its essay. These representations are concatenated so that the dimension of the pooled attention vectors $\boldsymbol{\alpha}'_c$, $\boldsymbol{\alpha}'_e$ is 1+2+4+8=15.

Finally, the prediction is made according to

$$\mathbf{Y} = \mathrm{softmax}\left(\mathrm{linear}([\boldsymbol{\alpha}'_e; \boldsymbol{\alpha}'_c; \mathbf{E}])\right), \quad (8)$$

where $\boldsymbol{\alpha}'_c$, $\boldsymbol{\alpha}'_e$ and E are concatenated.

5 Datasets

5.1 The Chinese Dataset

The construction of the Chinese Dataset mainly follows the definition and taxonomy of discourse elements proposed by Burstein et al. (2003). Specifically, we consider the following discourse elements:

Element	Train	Test	Total	%
Introduction	2,859	285	3,144	9.5
Thesis	881	151	1,032	3.1
Main Idea	4,443	578	5,021	15.2
Evidence	5,972	679	6,651	20.1
Elaboration	12,405	1,127	13,532	41.0
Conclusion	3,086	333	3,419	10.3
Other	170	20	190	0.6
Total	29,816	3,173	32,989	
# essays	1,112	118	1,230	
Avg. #Chinese chars per essay			843	
Avg. #sentences per essay			27	
Avg. #words per sentence			21	

Table 1: Basic statistics of the Chinese dataset.

- **Introduction** The role of introduction is to introduce background or attract readers' attention before making claims.

- **Thesis** The thesis express the central claim of an author with respect to the essay's topic.

- **Main Idea** The ideas establish foundational ideas or aspects that are related to the thesis.

- **Evidence** The evidence elements provide examples or other evidence that are used to support main ideas and thesis.

- **Elaboration** The elaboration elements further explain main ideas or provide reasons, but contain no examples or other evidence.

- **Conclusion** The conclusion sentence is the extension of the central argument, summarizes the full text, and echos the thesis of the essay.

- **Other** Other elements refer to the ones that do not match the above classes.

The dataset has 1,230 argumentative essays written by high school students, covering diverse topics. These essays were collected from a website LeleKetang.[1] We asked two annotators from the literal arts college to assign discourse elements to sentences from these essays according to a manual. The annotators discussed to reach a consensus and refined the manual for several rounds. We use one annotator's annotation as the gold answer, and the other's annotation as the prediction, and compute the F1 scores to measure the agreement, which is shown in Figure 3.

Table 1 shows the basic statistics of the dataset. The distribution of discourse elements is imbalanced. *Elaboration* and *evidence* sentences are

[1] http://www.leleketang.com/zuowen/.

[To conclude, **art could play an active role in improving the quality of people's lives**, $]_{s_1}$ [but I think that **governments should attach heavier weight to other social issues such as education and housing needs**$]_{s_2}$ [because **those are the most essential ways enable to make people a decent life.**$]_{s_3}$

Table 2: A sentence from the dataset of (Stab and Gurevych, 2017), with clause level component annotations (in bold), are split into three individual sentences s_1, s_2 and s_3.

Element	Train	Test	Total	%
Major Claim	598	153	751	10.3
Claim	1,202	304	1,506	20.6
Premise	3,023	809	3,832	52.3
Other	999	232	1,231	16.8
Total	5,822	1,498	7,320	
# essays	322	80	402	
Avg. #sentences per essay			19	
Avg. #words per sentence			20	

Table 3: Basic statistics of the English dataset converted from (Stab and Gurevych, 2017).

many more than *thesis* and *main idea* sentences. The type of *other* sentence accounts for a very small percentage of the dataset. The test dataset is 10% of the whole dataset.

5.2 The English Dataset

We also use the English student essay dataset released by Stab and Gurevych (2017). This dataset marks argument components, i.e., *major claim, claim*, and *premise*, at clause level. Table 2 shows an example sentence. The consecutive words in bold form three components, corresponding to *claim, major claim* and *premise*, respectively.

Because our model is at sentence level, we convert the original annotations to sentence level. First, an essay is split into sentences by NLTK. Then if a sentence contains only one argument component, we annotate this sentence as the type of this component; if a sentence contains more than one argument component, we further separate it into multiple sentences to ensure that each sentence has only one argument. The beginning of a new sentence is from the end of the last component. The end of a new sentence is the end of the component it contains. As shown in Table 2, three sentences s_1, s_2 and s_3 are generated from the original example sentence. If a sentence does not have any argument component, its label is *other*. Table 3 shows the basic statistics of the converted dataset.

6 Experiment

6.1 Experimental Settings

The max length of sentences is set to 40 words. Sentences are padded or truncated according to this length. The Tencent pre-trained word embeddings (Song et al., 2018) were used for experiments on the Chinese dataset. The dimension of word embeddings is 200. The Bert tokenizer and embeddings were used for experiments on the English dataset. The dimension of all the BiLSTM hidden layers is 256 on Chinese dataset, and 128 on English dataset. So is the dimension of d_k. The dimension of the attention vectors is 15. The optimizer is stochastic gradient descent (SGD) with a learning rate 0.1. The best models were selected for all settings based on the results on the validation data, which is 10% of the training data.

We use accuracy (Acc.) and macro-F1 as evaluation metrics.

6.2 Comparisons

We compare with the following systems.

- Feature-based. We adapt features from previous feature-based methods (Burstein et al., 2003; Stab and Gurevych, 2014; Song et al., 2015) to build a feature-based CRF model.
- HBiLSTM. The baseline described in Section 3 uses two BiLSTM layers to encode word sequences and sentences.
- BERT. We fine-tune BERT on training data to train a sentence classifier, because the lengths of many Chinese essays exceed the length constraint of BERT and it is expensive to train BERT-like models at discourse level.

6.3 Results on the Chinese Dataset

6.3.1 System Comparisons

Table 4 shows the performance of the baselines and DiSA. We can see that HBiLSTM performs even worse than feature-based approach. HBiLSTM has a low macro-F1 score, indicating that it has difficulties in identifying particular discourse elements. The two end-to-end models do not consider position information and interactions among sentences. The performance of BERT is worse than HBiLSTM. This verifies that sequence modeling is more proper than single sentence classification for this task. DiSA achieves the best performance

Model	Acc.	macro-F1
Feature-based	0.623	0.581
BERT	0.569	0.507
HBiLSTM	0.592	0.540
DiSA	**0.681**	**0.657**

Table 4: System comparisons.

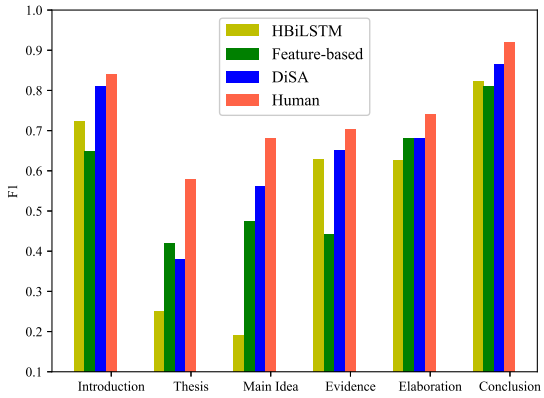

Figure 3: F1 scores on identifying specific discourse elements.

on all metrics, with a large improvement compared with the baselines.

Figure 3 further illustrates system performance on identifying specific discourse elements. The human performance is also measured by considering one annotator's annotation as the answer, and the other one's as the prediction.

The discourse elements that HBiLSTM is unable to accurately identify are *thesis* and *main idea*. Despite their importance for understanding a text, their scale is obviously smaller than other discourse elements, which may bring in obstacles for data-driven approaches.

Feature-based method performs better than HBiLSTM on identifying *thesis* and *main idea*. But it heavily relies on feature-engineering such as manually collected discourse markers and cue words. It does not perform well on identifying *evidence* due to the difficulties in designing related features.

DiSA is also an end-to-end model the same as HBiLSTM but performs much better. We will discuss the impacts of positional encoding and inter-sentence attention in Section 6.3.2 and 6.3.3.

Compared with the feature-based method, DiSA has comparable performance on identifying *thesis* but has superior results on identifying *main idea* (9% higher in F1 score) and *evidence* (21% higher in F1 score).

SPE Type	Acc.	macro-F1
Sinusoidal	0.674	0.638
PosEmbedding	0.657	0.628
RelativeSPE	**0.681**	**0.657**
No SPE	0.595	0.540
$+\mathbf{pos}_{global}$	0.591	0.540
$+\mathbf{pos}_{para}+\mathbf{pos}_{local}$	0.676	0.655

Table 5: The effects of different SPEs.

6.3.2 Analysis of Positional Encodings

This part investigates the effect of sentence positional encodings. We compare our relative sentence positional encoding (**relativeSPE**) with two other encoding strategies which are previously used for word sequences. **Sinusoidal** indicates the sinusoidal positional encoding which is used in Transformer (Vaswani et al., 2017). **PosEmbedding** uses a distributed vector to represent an absolute position. The position embeddings are learned during training. Each of the above three strategies is applied for modeling global position, local position and paragraph position, which are then combined according to Equation 5.

Table 5 lists the results of using different SPEs and modeling different positions. **RelativeSPE** performs best with improvements of 2-3% macro-F1 score compared with **Sinusoidal** and **PosEmbedding**. Without SPE, the metrics drop at least 6.2% compared with using any SPE strategy, and 8.6% compared with **relativeSPE**. If we explicitly add only $\mathbf{pos}_{global}$, the results even decrease. Perhaps recurrent neural networks such as LSTM naturally capture sequential positional information. However, encoding paragraph position ($\mathbf{pos}_{para}$) and local position ($\mathbf{pos}_{local}$) largely improves the performance. This indicates that proper structural positional encodings can exploit richer discourse structures than sequential structures.

6.3.3 Analysis of Inter-Sentence Attention

Table 6 shows the effects of removing inter-sentence attention (ISA) components from DiSA. We can see that both ElemSA and ContSA can make contributions, and ElemSA seems to have a larger effect on macro-F1 score. Removing ISA, the accuracy and the macro-F1 score decreases 1.8% and 2.2%.

Remind that ISA uses attention vectors as representations rather than the final output $\boldsymbol{\alpha}\mathbf{V}$ in the self-attention module. Table 6 also lists the performance that $\boldsymbol{\alpha}\mathbf{V}$ is used to replace attention vectors.

ISA Type	Acc.	macro-F1
DiSA	0.681	0.657
– ContSA	0.675	0.646
– ElemSA	0.677	0.647
– ISA	0.663	0.635
ISA with $\boldsymbol{\alpha}\mathbf{V}$	0.618	0.600

Table 6: The effects of inter-sentence attention (ISA).

Model	DiSA	– ISA	Δ
Introduction	**0.796**	0.792	–0.4%
Thesis	**0.383**	0.338	**–4.5%**
Main Idea	**0.577**	0.573	–0.4%
Evidence	**0.627**	0.578	**–4.9%**
Elaboration	**0.689**	0.677	–1.2%
Conclusion	**0.868**	0.850	– 1.8%

Table 7: Macro-F1 scores on identifying specific discourse elements.

The result is not good. This indicates that semantic relation among sentences is more important for DEI than the specific meaning of sentences.

We further analyze ISA's impact on specific discourse elements. As shown in Table 7, ISA affects the identification of the minority discourse element *thesis* most. It also benefits identifying *evidence* which is not a minority discourse element. Thesis sentences often relate to other sentences from different essay zones, while evidence sentences mainly provide facts or examples so they often relate to local context in content. ISA helps capture such patterns. The performance on other types also increases with different degrees.

Anyway, ISA provides a way to build useful representations by exploiting relations between sentences in the same text without any extra burden.

6.4 Results on the English Dataset

Table 8 and Table 9 show main experimental results on the English dataset.

The second column of Table 8 shows the results on distinguishing four component types (i.e., *major claim, claim, premise, other*). DiSA outperforms the baselines with a large margin on both accuracy and macro-F1. Again, removing SPE leads to a large performance decrease.

Stab and Gurevych (2017) conducted argument component classification experiments (classifying a component into *major claim, claim* and *premise*) by assuming that argument components have been correctly distinguished from *other* parts. To compare with their results, during training, the *other* type is removed from the label set and only the losses over *non-other* sentences are accumulated.

	4 classes		3 classes	
Model	Acc.	macro-F1	Acc.	macro-F1
BERT	0.673	0.596	-	-
HBiLSTM	0.680	0.501	-	-
DiSA	**0.772**	**0.699**	0.806	0.742
DiSA - SPE	0.687	0.529	0.710	0.534
DiSA+Feature	-	-	0.839	0.807
Eger et al. (2017)	-	-	-	0.730
Single-Best	-	-	-	0.773
Joint-Best	-	-	**0.850**	**0.826**

Table 8: Comparisons on the English dataset. Single-BEST and Joint-Best indicate the best results reported in (Stab and Gurevych, 2017), where Joint-Best incorporates relation identification as an auxiliary task.

The third column of Table 8 shows the comparison to the best results from (Stab and Gurevych, 2017). DiSA does not perform competitively based on the distributed representation only, because the baseline uses some strong hand-crafted features, especially the component position features, which rely on the correct argument component information. Thus we build a feature vector by incorporating the indicator features and a component position feature: number of preceding and following components in paragraph, out of 8 categories of features introduced in (Stab and Gurevych, 2017). The vector is concatenated with the distributed representation. This combination obtains improvements, outperforms Single-Best results, and achieves close performance compared with Joint-Best, which considers argumentative relation identification as an auxiliary task. We also attempt to apply the same strategy for the Chinese task. But the improvement is negligible. The reason may be that the indicator phrases used in Chinese essays is much less than in English essays. The English dataset heavily relies on phrases signaling beliefs or argumentative discourse connectors (Daxenberger et al., 2017).

Table 9 shows the macro-F1 scores of DiSA on identifying specific argument components. Without the ISA module, the identification of major claims and claims would decline by 3% and 1.4% absolute F1 score, respectively. This is consistent with the experimental results on the Chinese dataset. As a result, the effectiveness of the SPE and ISA can be verified on both the Chinese and the English datasets.

7 Conclusion

We presented a method DiSA to identify discourse elements in argumentative student essays by explicitly modeling structural positions and inter-

Model	DiSA	− ISA	Δ
Major Claim	**0.649**	0.619	−3.0%
Claim	**0.523**	0.509	−1.4%
Premise	**0.887**	0.882	−0.5%
Other	**0.737**	0.723	−1.4%

Table 9: Macro-F1 scores on identifying specific argument components on the English dataset.

sentence relations. The structural positional encoding considers relative positions of the sentence and its paragraph. Moreover, we use inter-sentence attention vectors to capture sentence relations in content and function. Experiments on a Chinese dataset and an English dataset show that (i) although it is simple, the positional encoding largely improves the performance. This indicates that modeling structural positions is feasible and important to analyze the role of sentences; (ii) discourse elements could be better identified with the help of inter-sentence attention vectors, especially the minority ones and the ones that have distinct relation patterns to other sentences. In future, we plan to evaluate DiSA on other discourse analysis tasks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61876113, 61876112), Beijing Natural Science Foundation (No. 4192017), Support Project of High-level Teachers in Beijing Municipal Universities in the Period of 13th Five-year Plan (CIT&TCD20170322) and Capital Building for Sci-Tech Innovation-Fundamental Scientific Research Funds. Lizhen Liu is the corresponding author.

References

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, and Martin Chodorow. 2001. Enriching automated essay scoring using discourse marking. *Technical Report*.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066.

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.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. Evaluating discourse in structured text representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 646–653.

Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins Karolinska, Lin Sun, and Ulla Stenius. 2010. Identifying the information structure of scientific abstracts: an investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 99–107. Association for Computational Linguistics.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1904–1916.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192.

Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, and Mitsuru Ishizuka. 2008. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Patrick Huber and Giuseppe Carenini. 2019. Predicting discourse structure using distant supervision from sentiment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2306–2316.

Yangfeng Ji and Noah A Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371.

Marco Lippi and Paolo Torroni. 2015. Context-independent claim detection for argument mining. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Marco Lippi and Paolo Torroni. 2016. Argument mining from speech: Detecting claims in political debates. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Todor Mihaylov and Anette Frank. 2019. Discourse-aware semantic self-attention for narrative reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2541–2552, Hong Kong, China. Association for Computational Linguistics.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM.

Huy Nguyen and Diane Litman. 2015. Extracting argument and domain words for identifying argument components in texts. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 22–28.

Nathan Ong, Diane Litman, and Alexandra Brusilovsky. 2014. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International Conference on Machine Learning*, pages 4052–4061.

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

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239. Association for Computational Linguistics.

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.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wei Song, Ruiji Fu, Lizhen Liu, and Ting Liu. 2015. Discourse element identification in student essays based on global and local cohesion. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2255–2261.

Wei Song, Ziyao Song, Lizhen Liu, and Ruiji Fu. 2020. Hierarchical multi-task learning for organization evaluation of argumentative student essays. In *IJCAI*, pages 3875–3881.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180, New Orleans, Louisiana. Association for Computational Linguistics.

Yi Song, Michael Heilman, Beata Beigman Klebanov, and Paul Deane. 2014. Applying argumentation schemes for essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 69–78.

Matthias Sperber, Graham Neubig, Ngoc-Quan Pham, and Alex Waibel. 2019. Self-attentional models for lattice inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1185–1197, Florence, Italy. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56.

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

Simone Teufel et al. 1999. *Argumentative zoning: Information extraction from scientific text*. Ph.D. thesis, Citeseer.

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.

Henning Wachsmuth, Khalid Al Khatib, and Benno Stein. 2016. Using argument mining to assess the argumentation quality of essays. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1680–1691.

Baosong Yang, Longyue Wang, Derek F Wong, Lidia S Chao, and Zhaopeng Tu. 2019. Convolutional self-attention networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4040–4045.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.