End-to-End Argument Mining for Discussion Threads Based on Parallel Constrained Pointer Architecture

Gaku Morio and **Katsuhide Fujita**
Tokyo University of Agriculture and Technology
2-24-16, Koganei, Tokyo, Japan
morio@katfuji.lab.tuat.ac.jp, katfuji@cc.tuat.ac.jp

Abstract

Argument Mining (AM) is a relatively recent discipline, which concentrates on extracting claims or premises from discourses, and inferring their structures. However, many existing works do not consider micro-level AM studies on discussion threads sufficiently. In this paper, we tackle AM for discussion threads. Our main contributions are follows: (1) A novel combination scheme focusing on micro-level inner- and inter- post schemes for a discussion thread. (2) Annotation of large-scale civic discussion threads with the scheme. (3) *Parallel constrained pointer architecture* (PCPA), a novel end-to-end technique to discriminate sentence types, inner-post relations, and inter-post interactions simultaneously.[1] The experimental results demonstrate that our proposed model shows better accuracy in terms of relations extraction, in comparison to existing state-of-the-art models.

1 Introduction

Argument Mining (AM) is a discipline which concentrates on extracting claims or premises, and inferring their structures from a discourse. In (Palau and Moens, 2009; Stab and Gurevych, 2014; Peldszus and Stede, 2013), they construed an argument as the pairing of a single claim and a (possibly empty) set of premises, which justifies the claim.

Generally, identifying structures for argument components (i.e., premises and claims) is categorized as a micro-level approach, and among complete arguments as a macro-level approach. There are some micro-level approaches (Palau and Moens, 2009; Stab and Gurevych, 2014, 2017), however, few AM studies aggressively consider a scheme of micro-level reply-to interactions in a thread. Though Hidey et al. (2017) provided a micro-level thread structured dataset, they considered an entire thread as a discourse. Thus, they allowed a premise that links to a claim in another post, while a post should be considered as a stand-alone discourse because a writer for each post is different. Also, we need to consider post-to-post interactions with the stand-alone assumption as a backdrop. Moreover, the dataset of (Hidey et al., 2017) with only 78 threads is too small to apply state-of-the-art neural discrimination models.

In addition to the shortage of micro-level anotations for discussion threads, no empirical study on end-to-end discrimination models which tackle discussion threads exist, to the best of our knowledge.

Motivated by the weaknesses above, this paper commits to the empirical study for discussion threads. Our main three contributions are as follows: (1) A novel combination scheme to apply AM to discussion threads. We introduce *inner-post* and *inter-post* schemes in combination. This combination enables us to discriminate arguments per post, rather than per thread as in (Hidey et al., 2017). In the former scheme, a post is assumed as a stand-alone discourse and a micro-level annotation is provided. In the second scheme, we introduce inter-post micro-level interactions. The introduction of the interactions allows us to capture informative argumentative relations between posts. (2) Large-scale online civic discussions are annotated by the proposed scheme. Specifically, we provide two phase annotation, and evaluate inter-annotator agreements. (3) A **parallel constrained pointer architecture** (PCPA) is proposed, which is a novel end-to-end neural model. The model can discriminate types of sentences (e.g., claim or premise), inner-post relations and inter-post interactions, simultaneously. In particu-

[1] Available at:
https://github.com/EdoFrank/EMNLP2018-ArgMining-Morio
including source codes.

Proceedings of the 5th Workshop on Argument Mining, pages 11–21
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

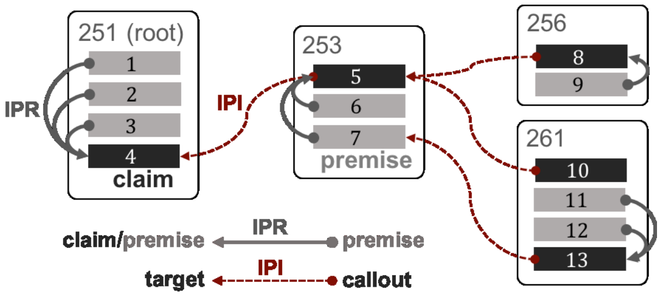


Figure 1: Example of our scheme for a thread.

lar, our PCPA achieved a significant improvement on challenging relation extractions in comparison to the existing state-of-the-art models (Eger et al., 2017; Potash et al., 2017). An advantage of our model is that the constraints of a thread structure are considered. The constraints make our architectures effective at learning and inferring, unlike existing pointer models.

While our dataset of discussion threads will make further advances in AM, the proposed PCPA will make end-to-end AM studies going forward.

2 Related Works

Stab and Gurevych (2017) argue that the task of AM is divided into the following three subtasks:

- **Component identification** focuses on separation of argumentative and non-argumentative text units and identification of argument component boundaries.
- **Component classification** addresses the function of argument components. It aims at classifying argument components into different types, such as claims and premises.
- **Structure identification** focuses on linking arguments or argument components. Its objective is to recognize different types of argumentative relations, such as support or attack relations.

The structure identification can also be divided to macro- and micro-level approaches. The macro-level approach as in (Boltužić and Šnajder, 2014; Ghosh et al., 2014; Murakami and Raymond, 2010) addresses relations between complete arguments and ignores the micro-structure of arguments (Stab and Gurevych, 2017). In (Ghosh et al., 2014), the authors introduced a scheme to represent relations between two posts by *target* and *callout*; however, their study discards micro-level structures in arguments because of their macro-level annotation. The micro-level approach as in (Palau and Moens, 2009; Stab and Gurevych, 2014, 2017) focuses on the relations between argument components. In (Palau and Moens, 2009), arguments are considered as trees. In (Stab and Gurevych, 2017), the authors also represented relations of argument components in essays as tree structures. However, they addressed discourses of a single writer (i.e., an essay writer) rather than multiple authors in a discussion thread. Therefore, we can't simply apply their scheme to our study.

Recently, the advances of automatic detection of argument structures have been seen in the discipline of AM. Some recent papers (Lippi and Torroni, 2015; Eckle-Kohler et al., 2015) propose argument component identification to extract argumentative components in the entire discourse. These works (Persing and Ng, 2016; Eger et al., 2017; Potash et al., 2017) showed link extraction task to find argumentative relations between argument components.

End-to-end discrimination models are also highlighted in AM. The reason is low error propagation compared with the other ends (pipeline). The pipeline models have to discriminate argument component identification and link extraction subtasks independently, and thus cause the error propagation (Eger et al., 2017). The authors propose manners to apply multi-task learning (Søgaard and Goldberg, 2016; Martínez Alonso and Plank, 2017) and LSTM-ER (Miwa and Bansal, 2016) to the end-to-end AM. Another end-to-end work for AM, Potash et al. (2017) argues that Pointer Networks (Vinyals et al., 2015; Katiyar and Cardie, 2017) which incorporate a sequence-to-sequence model in their classifier is a state-of-the-art model for argument component type prediction and link extraction tasks.

3 Argument Mining for Discussion Thread

3.1 Scheme

In this work, we present a novel scheme combining *inner-post* scheme of a stand-alone post with *inter-post* scheme that considers a reply-to argumentative relation. In the inner-post scheme (e.g., claim/premise types and inner-post relations), "one-claim" approach from (Stab and Gurevych, 2017) is adopted. In the inter-post scheme, the micro-level interaction in the spirit of (Ghosh et al., 2014) is employed. The definitions of inner-post relation and inter-post interaction are follows:

- **Inner-post relation (IPR)** is a directed argumentative relation in a post. Each IPR:($target \leftarrow source$) indicates that the $source$ component is either a justification for or a refutation of the $target$ component. Thus, a $source$ should be a premise, and each premise has a single outgoing link to another premise or claim (Eger et al., 2017).
- **Target** is a head of IPI that has been called out by a subsequent claim in another post that replies to the post of the target.
- **Callout** is a tail of IPI that refers back to a prior target. In addition to referring back to the target, a callout must be a claim.[2]
- **Inter-post interaction (IPI)** is the micro-level relationship of two posts: parent post and child post that replies to the parent post. A relation ($parent \leftarrow child$) represents the $child$ is a **callout** and $parent$ is a **target**.

Figure 1 shows our combination scheme for a discussion thread.

3.2 Dataset

To develop a sufficient AM corpus for discussion threads, we have annotated an original large-scale online civic discussion (Morio and Fujita, 2018a). The civic discussion data is obtained by an online civic engagement on the *COLLAGREE* (Ito et al., 2014; Morio and Fujita, 2018b) including a thread structure. The discussion was held from the end of 2016 to the beginning of 2017, and co-hosted by the government of Nagoya City, Japan. The accumulated data includes 204 citizens, 399 threads, 1327 posts, 5559 sentences and 120241 tokens spelled in Japanese.[3] To the best of our knowledge, this work is the first approach which annotates large-scale civic discussions for AM.[4]

3.3 Annotation Design

In (Peldszus and Stede, 2013), the authors argue that the annotation task for AM contains the following three subtasks: (1) segmentation, (2) segment classification and (3) relationship identification. The segmentation requires extensive human resources, time, and cost. Therefore, we apply a rule-based technique for the segmentation. Then, we consider each sentence as an argument component candidate (ACC). For classifying the argument component, the ACC types (claim, premise or non-argumentative (NonArg)) for each ACC are annotated. Finally, the relationship identification needs to annotate IPRs and IPIs.

Using multiple processes for multiple annotation subtasks is common (Meyers and Brashers, 2010; Stab and Gurevych, 2014, 2017). To annotate our data, we provide two phases. In the first phase, we concentrate on annotating ACC type and IPR, and create a temporal gold standard. In the second phase, IPI is annotated using the temporal gold standard.

We employed a majority vote to create the gold standard. All three annotators independently annotated in this work. The procedure of the first phase for compiling the temporal gold standard is as follows.

A1: Each ACC type is decided on a majority vote. When the ACC type of the sentence cannot be decided by majority vote, NonArg is assigned to them.
A2: Each IPR (link existence) is decided on a majority vote.
A3: Merging the results from A1 and A2, and obtaining trees where root is a claim. Thus, we have trees to the number of claims in a post.
A4: Eliminating premise tags that do not belong to any trees, assigning them to NonArg, and eliminating their IPR.

3.4 Annotation Result

Inter-annotator agreement for ACC type, IPR and IPI annotations are calculated using Fleiss's κ (Fleiss, 1971). First, we attempt to evaluate the agreement of the first phase annotations, however, the κ of IPR is relatively low: 0.420. The annotators are less likely to agree on serial arguments (Stab and Gurevych, 2017) like ($premise \leftarrow premise$) relations.[5]

[2] To restrict a callout to a claim makes our problem more simple because the number of outgoing links from a claim becomes one at a maximum. Thus, we introduced the restriction.

[3] The average of the number of posts per thread is 3.33 (standard deviation is 3.29), the depth of threads is 1.09 (standard deviation is 1.19), the number of sentences per post is 4.19 (standard deviation is 3.33) and the number of words per sentence is 21.63 (standard deviation is 19.92).

[4] Recently, Park and Cardie (2018) provide a similar dataset of civic engagement, while their dataset doesn't consider post-to-post relations sufficiently.

[5] Unlike with Persuasive Essays (Stab and Gurevych, 2017), citizen's documents for civic discussions are seldom

Corpus	Type	Size	κ
COLLAGREE	Claim	1449	.531
	Premise	2762	.554
	NonArg	1348	.529
	IPR w/ A0	2762	.466
	IPI	745	.430
Persuasive Essays	Claim	1506	.635
	Premise	3832	.833
	Inner-essay rel	3832	.708-.737

Table 1: Inter-annotator agreement scores for the two corpora.

Therefore, we introduce an initial process A0, transforming $(premise1 \leftarrow premise2)$ into $(\textit{root claim of premise2} \leftarrow premise2)$, before A1.[6]

Table 1 summarizes the number of each type of relation and inter-annotator agreement.[7] For comparison, we also mention the annotation results of Persuasive Essays (Stab and Gurevych, 2017). Unlike the essay dataset, our datasets contain badly-structured writings, resulting in low agreement. However, classification tasks can be applied as (Landis and Koch, 1977) refers to the κ value from 0.41 to 0.61 as "moderate agreement". Moreover, the agreement of IPR is improved by providing the process A0.

4 Discriminating ACC Type, Inner-Post Relation and Inter-Post Interaction

This section describes the study on our end-to-end discrimination model, which identifies ACC type, IPR and IPI for our annotated dataset.

4.1 Thread Representation as a Sequence

If the thread itself contains flow of its argument, only the thread itself is considered as the desirable input for a discrimination model. Thus, we describe a way of representing a thread with an input sequence.

In this work, we extend the sequence representation of (Eger et al., 2017; Potash et al., 2017). The creation of thread representation as an input sequence consists of the following two steps. First, we assume each element of the input sequence for recurrent neural network is a sentence representation, rather than a word representation. Second, we sort the sentence representations by the thread depth order. In addition, for each thread depth, we in turn order them according to the timestamp of their post, and insert separator representations. The first one makes it possible to input a short sequence to LSTM units (Hochreiter and Schmidhuber, 1997). The second makes a classifier easy to discriminate considering the hierarchy of a thread and reply relations. Figure 2 shows an example of a thread representation as sequence.

4.2 Parallel Constrained Pointer Architecture

One of the main technical contributions of our approach is to provide a discrimination model that classifies ACC type, IPR and IPI simultaneously via end-to-end learning. A Pointer Network (PN) for end-to-end AM achieves state-of-the-art results (Potash et al., 2017), which leads to applying a PN based technique to our scheme. Unfortunately, the naive PN did not achieve the result expected (the quantitative results are shown in Section 5), because the simple PN is unable to constrain its search space for thread structures. For instance, an inner-post relation classifier could discriminate with no need to search out of its post, or an inter-post interaction classifier could classify with no need to search out of the parent post and child post. Therefore, we propose a novel neural model named *parallel constrained pointer architecture* (PCPA). PCPA provides two parallel pointer architectures: IPR and IPI discrimination architectures that adopt the apparent constrains of threads.

Sentence Representation as Input

First, we introduce the input representation. Given N threads $(T_1, \ldots, T_N)$, we denote T_i's posts which are sorted in thread depth order, and then timestamp order as described in Section 4.1 as $(P_1^{(i)}, \ldots, P_{N_i}^{(i)})$, where N_i represents the number of posts in T_i. In addition to the thread and post representations, write $(S_1^{(i,j)}, \ldots, S_{N_{i,j}}^{(i,j)})$ for sentences in post $P_j^{(i)}$, where $N_{i,j}$ represents the number of sentences in $P_j^{(i)}$. Note that separator rep-

well-structured. Thus, we don't see the point in providing a more complex scheme (i.e., allowing $(premise \leftarrow premise)$ relations).

[6] For example, two IPRs $\{(claim1 \leftarrow premise1), (premise1 \leftarrow premise2)\}$ are transformed to $\{(claim1 \leftarrow premise1), (claim1 \leftarrow premise2)\}$.

[7] Outgoing IPI links are composed of 574 claims, 109 premises, and 62 NonArgs. Considering that a callout should be a claim, the $(claim \leftarrow claim)$ interaction accounts for 77% of the total. The results indicate that IPIs are pretty argumentative. In addition, we annotated support/attack relations (Cocarascu and Toni, 2017). The results show support accounts for 86% and attacks for 7% of the total IPIs.

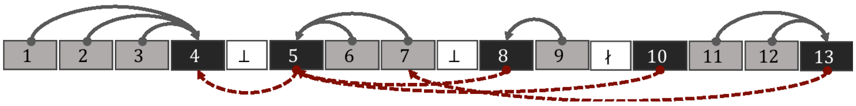


Figure 2: Sequence representation from Figure 1. ⊥ and ∤ denotes a separator representation of thread depth and posts.

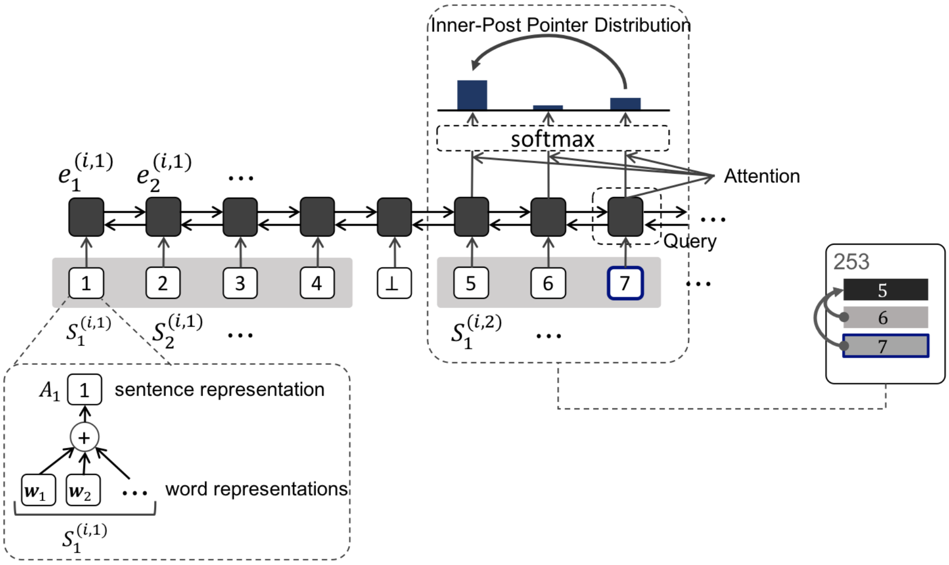


Figure 3: Example of the constrained pointer architecture of inner-post relation (IPR) identification, discriminating the IPR target from the ACC "7".

resentations are not considered in the notation.

Then, $\boldsymbol{w}_n$ is given initially, an embedding vector of nth word in a sentence $S_k^{(i,j)}$, a sentence representation for an input of LSTM is represented as: $A_k = \sum_n \boldsymbol{w}_n$, where $\boldsymbol{w}_n$ is gained from bag-of-words (BoW) or word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Stab and Gurevych, 2017). In our study, we employed BoW and a fully connected layer with a trainable parameter to learn word embeddings. Subsequently, we provide Bidirectional LSTM (BiLSTM) (Graves and Schmidhuber, 2005) because PN requires encoding steps. At each time step of the encoder BiLSTM, PCPA considers a representation of an ACC. Thus, the hidden representation e_i of BiLSTM becomes the concatenation of forward and backward hidden representations. To simplify the explanation, we denote the hidden representations of $(S_1^{(i,j)}, \ldots, S_{N_{i,j}}^{(i,j)})$ as $(e_1^{(i,j)}, \ldots, e_{N_{i,j}}^{(i,j)})$. For better understanding, we show notations in Figure 3.

Discriminating Inner-Post Relation

The general PN of (Potash et al., 2017) uses all hidden states e_i. Alternately, PCPA can limit the states to improve the accuracies, since each premise has a single outgoing link to another sentence in its post. Hence, we provide an approach to discriminate IPR using only inner-post hidden states of the BiLSTM.

Figure 3 shows the example IPR discrimination in thread T_i; for example, we assume that the inner-post relation of the sentence written as "7" in the 3rd ACC of post $P_2^{(i)}$ is classified. The general PN needs to consider all e_i, therefore, the search space is large. On the other hands, our proposed PCPA can consider $(e_1^{(i,2)}, e_2^{(i,2)}, e_3^{(i,2)})$, which needs to use the hidden states of its post only. Therefore, our constrained architecture can reduce the search space significantly.

In general, given W_1, W_2, and v_1, parameters of attention model (Luong et al., 2015) for PN,

$$u_l^{(i,j,k)} = v_1^\top \tanh\left(W_1 e_l^{(i,j)} + W_2 e_k^{(i,j)}\right) \quad (1)$$

represents a degree that kth ACC in post $P_j^{(i)}$ has an outgoing link to lth ACC. Moreover, we can assume $e_k^{(i,j)}$ as a query vector. Supposing the ACC has no outgoing link, we can consider the ACC learned to point to itself. Although equation (1) is real-value, a distribution over the IPR input is

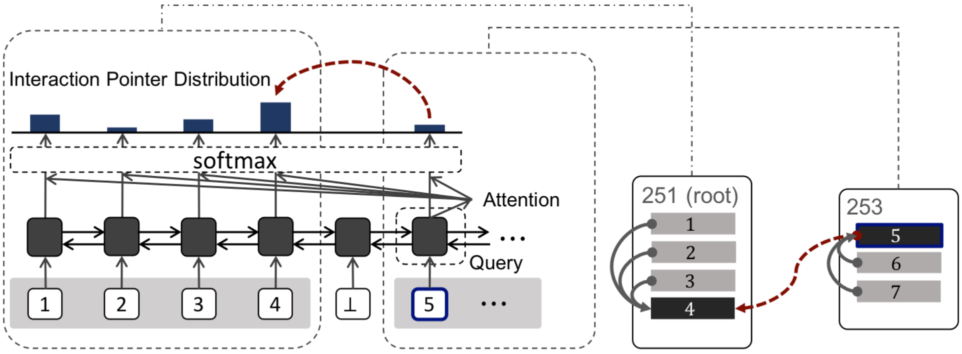


Figure 4: Example of the constrained pointer architecture of inter-post interaction (IPI) identification, discriminating the IPI target that is called out from the ACC "5".

considered by taking softmax function, i.e.,

$$p(y_k^{ipr} \mid P_j^{(i)}) = \mathrm{softmax}(u^{(i,j,k)}) \tag{2}$$

representing the probability that kth ACC in post $P_j^{(i)}$ has an outgoing link to lth ACC in $P_j^{(i)}$. Therefore, the objective for IPR in thread T_i is calculated by taking the sum of log-likelihoods for all posts:

$$L_i^{ipr} = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i,j}} \log p(y_k^{ipr} \mid P_j^{(i)}) \tag{3}$$

Discriminating Inter-Post Interaction

As the definition of target and callout in our scheme, IPI exists between a parent post and child post that replies to the parent. Thus, PCPA can discriminate IPI with no need to use all of the hidden representations of the LSTM. In other words, it can discriminate IPI without searching outside of the two posts.

Hence, we design an output layer that requires only a set of reply pairs in thread T_i. Specifically, we assume that $R^{(i)} = \{(j_1, j_2), \cdots\}$ where $j_1 \neq j_2 \wedge j_1 < j_2$ for a set of parent-child pairs in thread T_i. Supposing j_1 is the index of a parent post and j_2 represents the index of the child post that replies to the j_1. Note that when thread T_i does not have any reply pairs, $R^{(i)} = \varnothing$. Considering the above, a technique that is similar to the IPR's technique is introduced.

Figure 4 shows the example IPI discrimination in thread T_i; supposing that we are going to discriminate a target that is called out from ACC "5" in the figure. In this case, the search space is limited by the parent post $(e_1^{(i,1)}, \ldots, e_4^{(i,1)})$. Moreover, we add an element $e_1^{(i,2)}$ so that a callout can point itself if there's no target in its parent post. The left four outputs in the "Interaction Pointer Distribution" indicate a discrete probabilistic distribution that the callout ACC "5" links to target sentences in its parent post, and an output on the far right represents a probability that the callout links to itself.

The equation (1) uses a query in the PN, so we in turn concentrate on using a query vector for the callout in IPI. Herein, we introduce an additional PN for IPI using new attention parameters, W_3,W_4 and v_2, as:

$$q_l^{(i,j,k)} = v_2^\top \tanh\left(W_3 e_l^{(i,j)} + W_4 e_k^{(i,j)}\right) \tag{4}$$

where $e_k^{(i,j)}$ is the query from the callout. Supposing that the reply pair is (j_1, j_2), a target of kth ACC of the child post $P_{j_2}^{(i)}$ is searched. The expanded vector $\left[q^{(i,j_1,k)}; q_k^{(i,j_2,k)}\right]$ is obtained by concatenating the attention vectors $q^{(i,j_1,k)}$ from the parent post and a vector $q_k^{(i,j_2,k)}$ from the callout. This expansion process is the same as the process of (Merity et al., 2016). Finally, given all reply pairs of thread T_i, the log-likelihood is calculated as follows:

$$p(y_k^{ipi} \mid P_{j_1}^{(i)}, P_{j_2}^{(i)}) = \mathrm{softmax}([q^{(i,j_1,k)}; q_k^{(i,j_2,k)}])$$

$$L_i^{ipi} = \sum_{(j_1,j_2) \in R^{(i)}} \sum_{k=1}^{N_{i,j_2}} \log p(y_k^{ipi} \mid P_{j_1}^{(i)}, P_{j_2}^{(i)}) \tag{5}$$

Discriminating ACC Type

At each time step of the BiLSTM, the type classification task predicts whether it is claim, premise, or NonArg. The ACC type of sentence $S_k^{(i,j)}$ can be classified by taking softmax of $z_k^{(i,j)} = W_{type} e_k^{(i,j)} + b_{type}$, where W_{type} and b_{type} are parameters. An objective for the type classifier can also be described by taking the sum of log-likelihoods for all posts as:

$$p(y_k^{type} \mid P_j^{(i)}) = \mathrm{softmax}(z_k^{(i,j)})$$

$$L_i^{type} = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i,j}} \log p(y_k^{type} \mid P_j^{(i)}) \qquad (6)$$

Joint Learning

Combining objectives of IPR (equation (3)), IPI (equation (5)) and the ACC type (equation (6)), the training objective of PCPA is shown as follows:

$$\begin{aligned} Loss = & \frac{1}{N} \sum_i (-\alpha L_i^{ipr} - \beta L_i^{ipi} \\ & - (1 - \alpha - \beta) L_i^{type}) \end{aligned} \qquad (7)$$

where α and β are hyperparameters which adjust the weight of tasks in our cost function. Note that $\alpha, \beta \in [0, 1] \wedge \alpha + \beta < 1$.

5 Experiments

5.1 Experimental Settings

Evaluation Metric

For the evaluation of ACC types, IPR and IPI discrimination, we adopt precision, recall and F1 scores. To obtain the precision and recall, we introduce a way to compute positive and negative cases by creating relations (Stab and Gurevych, 2017), excluding self-pointers. [8] [9]

Baselines

First, we employ state-of-the-art PN techniques from (Potash et al., 2017) as baselines. The use of these baselines was decided because our model PCPA (**Our Model**) employs pointer architectures. As the authors proposed two techniques, sequence-to-sequence model (**PN with Seq2Seq**) and w/o sequence-to-sequence model (**PN without Seq2Seq**), we have the two models for comparison.

To analyze how a *non* PN model works, multi-task learning is employed to the baseline (Søgaard and Goldberg, 2016) (**STagBLSTM**) by (Eger et al., 2017). STagBLSTM is composed of shared BiLSTM layers for subtasks, and output layers for each subtask. In (Eger et al., 2017), the authors provided a BIO tagging task, however, the task is not required in our work because BiLSTM handles an input as sentence representation rather than as word representation. In this paper, we use one BiLSTM.[10]

To show end-to-end learning models are effective for AM on thread structures, we provide the following three task specific baselines. First, feature-based SVM (Stab and Gurevych, 2017) (**SVM - T**) is introduced. T indicates each subtask of the claim classifier, premise classifier, IPR classifier, and IPI classifier. In addition, random forest (**RF - T**) and the logistic regression technique (Peldszus and Stede, 2015) (**Simple - T**) are also introduced. For each task specific model, BoW features the top 500 most frequent words [11].

We assume that each output of PN with Seq2Seq, PN without Seq2Seq or STagBLSTM does not satisfy the constraints as a self-pointer. This is because inappropriate outputs with constraint violations of IPR and IPI by these approaches will happen, i.e., they can predict IPI out of parent and child posts. The assumption maintains the false positive (FP) of baselines, since a self-pointer which results from a chance is not counted as FP. This condition gives the baselines the advantage of precision over our models. Therefore, this assumption is convincing.

The following describes our implementation details. The implementation of neural models are by Chainer (Tokui et al., 2015). The hyperparameters are the same as (Potash et al., 2017) for the PN baselines and our models[12]. In the interest of time,

[8] For example, supposing there is a post which contains three sentences, (S_1, S_2, S_3), and two gold standard IPRs, $(S_1 \leftarrow S_2)$ and $(S_1 \leftarrow S_3)$. This is exactly the case that positive cases of IPR are $\{(S_1 \leftarrow S_2), (S_1 \leftarrow S_3)\}$, and negative cases are all sentence pairs excluding self-pointers. That is, negatives are $\{(S_2 \leftarrow S_1), (S_2 \leftarrow S_3), (S_3 \leftarrow S_1), (S_3 \leftarrow S_2)\}$. In this case, self-pointer cases are $\{(S_1 \leftarrow S_1), (S_2 \leftarrow S_2), (S_3 \leftarrow S_3)\}$.

[9] For IPI, we are also able to create sentence pairs. For instance, suppose there is a parent post which contains three sentences (S_1, S_2, S_3), a child post that contains two sentences (S_4, S_5), and a gold standard IPI, $(S_2 \leftarrow S_5)$. The positive case of IPI is exactly $\{(S_2 \leftarrow S_5)\}$, and negative cases are all sentence pairs excluding self-pointers, that is, $\{(S_1 \leftarrow S_4), (S_1 \leftarrow S_5), (S_2 \leftarrow S_4), (S_3 \leftarrow S_4), (S_3 \leftarrow S_5)\}$.

[10] Though there are some variation models other than the single BiLSTM model, our preliminary experiments show a non-significant improvement.

[11] In fact (Stab and Gurevych, 2017) and employs rich features such as structural features. We only use BoW for comparison because the properties of COLLAGREE corpus substantially differ from their corpus.

[12] Hidden input dimension size 512, hidden layer size 256 for the BiLSTMs, hidden layer size 512 for the LSTM decoder of PN without Seq2Seq, and high dropout rate of 0.9 (Srivastava et al., 2014; Zarrella and Marsh, 2016). All models are trained with the Adam optimizer (Kingma and Ba,

Model type	Model name	Type classification			Link extraction			
		Claim F1	Premise F1	NonArg F1	IPR Precision	IPR F1	IPI Precision	IPI F1
Joint learning	Our Model	58.5	68.7	36.0	33.8	***40.8**	19.6	***24.8**
	Our Model - Hyp	58.1	**71.5**	**58.8**	***45.8**	***44.3**	***30.4**	***26.9**
	STagBLSTM	54.2	65.6	56.9	14.3	14.9	21.0	12.6
	PN with Seq2Seq	58.3	70.8	48.6	35.7	27.2	13.0	19.4
	PN without Seq2Seq	**60.1**	71.3	53.1	36.6	35.0	26.5	20.8
Task specific	SVM - T	53.3	64.4	52.3	13.8	22.4	6.4	11.5
	RF - T	41.0	66.8	38.3	0	0	100	1.4
	Simple - T	41.1	66.1	38.3	0	0	0	0
Joint learning w/o separator	Our Model w/o separator	43.1	66.3	29.6	30.0	36.1	9.9	13.7
	STagBLSTM w/o separator	51.8	66.1	55.2	13.9	14.5	16.1	10.8
	PN with Seq2Seq w/o separator	40.7	67.8	52.7	30.4	23.2	10.8	14.6
	PN without Seq2Seq w/o separator	43.4	67.6	53.7	29.5	21.1	19.0	6.0

Table 2: **Top**: Our models vs. joint baselines (%). * indicates significant. at $p < 0.01$, two-sided Wilcoxon signed rank test (Derryberry et al., 2010), compared with each baseline. **Middle**: Performances of task specific baselines. **Bottom**: Performances of joint models w/o separator representations.

we ran 50 epochs, and used the trained model for testing. The COLLAGREE dataset is divided into training threads and testing threads at $8 : 2$. In addition, we use the following hyperparameters in equation (7): $\alpha = \beta = 1/3$. However, total loss of L^{ipr} and L^{ipi} tends to enlarge since they have to calculate a sum of the sentence pairs. Hence, we provide a model with tuned hyperparameters $\alpha = \beta = 0.15$ (**Our Model - Hyp**) for comparison.

5.2 Experimental Results

Table 2 summarizes the results of our models and baselines. For each model, we showed the best F1 score in the table. Due to limitations of space, we omitted recalls and some precisions. Surprisingly, all models performed as well as we expected in our dataset, in spite of low agreements (see Table 1). Although the basis of the ACC type classifier of PCPA is the same as the PN model, our model with tuned hyperparameters is better at NonArg identification than the baseline PN models.

Both of our models significantly outperform all baselines for the IPR and IPI discrimination tasks. "Our Model - Hyp" achieves F1 $+9.3\%$ in IPR identification in comparison with the best baseline PN without Seq2Seq. This is the most important result because it indicates that incorporating constrains of thread structures with the PNs makes relation classifiers easy to learn and discriminate.

STagBLSTM shows lower scores in terms of both IPR and IPI identification, implying the difficulty of the use of the multi-task learning of BiLSTM. In addition, Table 2 (Middle) also illustrates that most neural models yield better F1 scores in comparison with the task specific models. In addition, the logistic regression and RF are overfitted, despite that cross validations are employed. Thus, end-to-end learning assumes an important role for AM, even in thread structures.

Model	IPR - F1	IPI - F1
Our Model	±0.7	±1.8
PN with Seq2Seq	±2.3	±1.2
PN without Seq2Seq	±2.7	±3.9

Table 3: Standard deviations of F1 scores (%)

Effectiveness of Separator Representation

To demonstrate the effectiveness of the separator representations, we conducted an experiment. In Table 2 (Bottom), the models without the separator input representations are indicated as "w/o separator". It shows that separator representations dramatically improve scores of PN based models. This remarkable result is from the ability to learn the structural information of a thread by encoding separators in the BiLSTM.

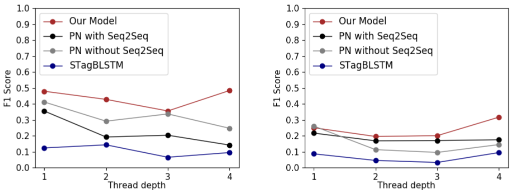
Stability

To analyze the stability of our models, we compare standard deviations among three selected models. Table 3 shows standard deviations for the three models. These results indicate that our model has lower standard deviations for IPR than baseline PN models. The reason for this is the size of search space: our models can effectively limit the search space based on thread structures.

2014) with a mini batch size of 16.

Model	IPR - F1	IPI - F1
Our Model	***39.6**	***22.6**
Our Model with Param Share	36.7	11.9

Table 4: The effect of parameter sharing of the two pointer architectures.



(a) IPR

(b) IPI

Figure 5: Performances on different thread depths.

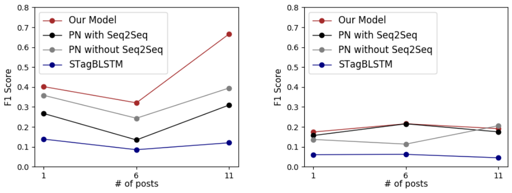
Analysis for Parallel Design

Next, we show how our models improve their performance by employing our parallel pointer architecture. Herein, we provide a new model of PCPA with a single PN (**Our Model with Param Share**), which shares v_1, W_1 and W_2 in equation (1) and v_2, W_3 and W_4 in equation (4), respectively. Table 4 demonstrates the mean of F1 scores for our model and Our Model with Param Share. Note that the average performances are lower than the best performances in Table 2. The scores indicate that sharing the two pointer architecture parameters is not effective in our proposed model. We estimate this is because poor association (Caruana, 1997) between the IPR and IPI identification tasks exists. Therefore, our approach of using two parallel pointer architectures is effective.

Performance Specialized in Threads

We examine how our models are specialized in thread structures. Specifically, we limit the threads in test datasets by specific thresholds, and then analyze performance transitions. We conduct two experiments as the thread depth is limited (Figure 5a and 5b). While the baselines performances decrease as the thread depth increases, our model keeps its F1 score because of the separators and the search space. The separator representations for an input increase according to the thread depth, and the baseline PN models need to use wider range of hidden states in comparison with the PCPA model. In other words, our models are extremely effective, even for deeper threads.

We also limit the threads that we can use in test data by the number of posts (Figure 6a, and 6b). For discriminating IPR, our model increasingly outperforms others in accordance with the number of posts. Figure 6b indicates that the difference between our model and baselines is minimal. This is because the number of posts does not affect the thread depth, necessarily. Most of *COLLAGREE*'s threads have a depth of at most 2. In other words, Figure 6b also implies the depth of threads affects the improvement of IPI identifications.



(a) IPR

(b) IPI

Figure 6: Performances on different number of posts. When the horizontal value is 1, we test using threads which contains [1-5] posts.

6 Conclusion

This paper presented an end-to-end study on discussion threads for argument mining (AM). We proposed an AM scheme that is composed of micro-level inner- and inter- post scheme for a discussion thread. The annotation result shows we acquire the valid and pretty argumentative corpus. To structuralize the discourses of threads automatically, we propose a neural end-to-end AM technique. Specifically, we presented a novel technique to utilize constraints of the thread structure for pointer networks. The experimental results demonstrated that our proposed model outperformed state-of-the-art baselines in terms of relation identifications.

Possible future work includes enhancing our scheme for less restricted conditions, i.e., multiple targets from one callout.

Acknowledgments

This work was supported by CREST, JST (JPMJCR15E1), Japan and JST AIP-PRISM Grant Number JPMJCR18ZL, Japan. We thank Takayuki Ito, Eizo Hideshima, Takanori Ito and Shun Shiramatsu for providing us with the COLLAGREE data.

References

Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379. Association for Computational Linguistics.

DeWayne R. Derryberry, Sue B. Schou, and W. J. Conover. 2010. Teaching rank-based tests by emphasizing structural similarities to corresponding parametric tests. *Journal of Statistics Education*, 18(1).

Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.

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.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics, ArgMining@ACL 2014, June 26, 2014, Baltimore, Maryland, USA*, pages 39–48.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

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

Takayuki Ito, Yuma Imi, Takanori Ito, and Eizo Hideshima. 2014. Collagree: A faciliator-mediated large-scale consensus support system. In *Proceedings of the 2nd International Conference of Collective Intelligence*.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1).

Marco Lippi and Paolo Torroni. 2015. Context-independent claim detection for argument mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 185–191. AAAI Press.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421. The Association for Computational Linguistics.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Renee A. Meyers and Dale Brashers. 2010. Extending the conversational argument coding scheme: Argument categories, units, and coding procedures. *Communication Methods and Measures*, 4(1-2):27–45.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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

Gaku Morio and Katsuhide Fujita. 2018a. Annotating online civic discussion threads for argument mining. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, page (to appear).

Gaku Morio and Katsuhide Fujita. 2018b. Predicting argumentative influence probabilities in large-scale online civic engagement. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1427–1434, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: Classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 869–875, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Joonsuk Park and Claire Cardie. 2018. A corpus of erulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *Int. J. Cogn. Inform. Nat. Intell.*, 7(1):1–31.

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, Lisbon, Portugal. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

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

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1375–1384. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 231–235. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1501–1510.

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

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS).*

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Guido Zarrella and Amy Marsh. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463, San Diego, California. Association for Computational Linguistics.