Towards Exploiting Argumentative Context for Argumentative Relation Identification

Tatsuki Kuribayashi† Paul Reisert‡ Naoya Inoue† Kentaro Inui† ‡

†Tohoku University ‡RIKEN Center for Advanced Intelligence Project (AIP)
{kuribayashi,naoya-i,inui}@ecei.tohoku.ac.jp
paul.reisert@riken.jp

1 Introduction

Argument mining is the task of identifying argument structures in argumentative texts. This task is useful for many applications such as document summarization, opinion mining, and automated essay scoring [10, 13]. In the literature, several subtasks for argument mining have been extensively studied, such as *argument component type classification*, *stance classification*, and *argumentative relation identification* [7, 11, 4, 8, 9, 5]. This paper addresses the task of argumentative relation identification due to its recent popularity in argument mining.

Consider the argumentative text[1] in Figure 1, where argument components (ACs), basic units of arguments, are already identified. Argument component type classification aims at classifying ACs into a premise or claim (e.g. classifying AC_1 into a claim and AC_2 into a premise). Stance classification aims at classifying the stance of ACs towards a claim as either *proponent* or *opponent* (e.g. classifying AC_1 into a *proponent* stance and AC_2 into a *opponent* stance). Argumentative relation identification aims to identify an argumentative link between two ACs, and if it exists, classify it into two classes: *attack* or *support* (e.g. identifying the *attack* relation from AC_2 to AC_1).

Conventional approaches have focused on creating features using the local input ACs rather than using macro-level information such as the overall structure of an argument [7, 11, 1]. However, argumentative relations are closely related to each other and they form argument diagrams [6]. Thus, we speculate that the information of surrounding argumentative context (e.g. other argumentative relations) can be useful for predicting a relation.

For example, in Figure 1, AC_2 attacks AC_1 and AC_3 attacks AC_2. If we were to predict the *attack* relation from AC_3 to AC_2, knowing whether AC_2 is attacking another AC would be useful information, because a writer frequently uses such macro structure as a tactic for strengthening their argument. For example, in Figure 1, the writer gives a possible counter-argument to their claim (AC_2 attacks AC_1) and then attacks it immediately (AC_3 attacks AC_2), which makes it difficult for others to attack.

[1] Slightly modified version of the text (micro_b006) taken from Peldszus and Stede (2015) [7]

Topic : *Should Germany introduce the death penalty?*
AC_1 : *The death penalty is a legal means that as such is not practicable in Germany.*
AC_2 : *Even if many people think that a murderer has already decided on the life or death of another person,*
AC_3 : *this is precisely the crime that we should not repay with the same.*

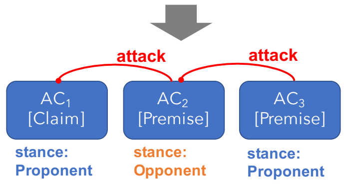

Figure 1: An example of argument structure with three ACs

A corpus-based analysis in previous research [3] revealed that some macro-level structures are frequently observed across different corpora. In this paper, we extend the previous work by investigating the effectiveness of argumentative context (i.e. macro-level information such as other surrounding argumentative relations) as a clue of argumentative relation identification. Our experiments demonstrate that such macro-level information is helpful for predicting argumentative relations.

2 Related work

Previous work has focused on two types of approaches for solving the task of argumentative relation identification. The first approach is to formalize argumentative relation identification as a structured prediction problem (i.e. predicting a graph consisting of argumentative relations from an argumentative text) [7, 9, 11, 8]. To predict a graph, Peldszus et al. [7] use the Maximum Spanning Tree (MST) algorithm and Stab et al. [11] use Integer Linear Programming (ILP). However, these works do not exploit information from other predicted argumentative relations when predicting a relation. Niculae et al. [5] use factor graphs for structured prediction. They report that higher-order features (e.g. combination features of argumentative links) increase the precision of AC type classification and link identification (i.e. whether argumentative relation exists or not between two ACs). Potash et al. [9] use Pointer Networks [12], which considers the previous prediction in the decoding steps, for AC type classifica-

Table 1: Macro-level structures found in the MT and PE corpus (excerpt from [3]). The highest percentage among three relation types is in bold and the lowest one is underlined. In PE, 9,000 *neither* relation pairs were sampled (about 10% of all *neither* pairs in the corpus).

MT

relation	source		target			
	attacked	supported	attacked by another	supported by another	attacking	supporting
attack	**62/174 (36%)**	16/174 (9%)	27/174 (16%)	90/174 (52%)	**64/174 (37%)**	23/174 (13%)
support	23/290 (8%)	51/290 (18%)	**139/290 (48%)**	**202/290 (70%)**	21/290 (7%)	57/290 (20%)
neither	582/2000 (29%)	**686/2000 (34%)**	354/2000 (18%)	373/2000 (19%)	638/2000 (32%)	**1197/2000 (60%)**

PE

relation	source		target			
	attacked	supported	attacked by another	supported by another	attacking	supporting
attack	**36/219 (16%)**	32/219 (15%)	**68/219 (31%)**	106/219 (48%)	**38/219 (17%)**	19/219 (9%)
support	18/3613 (0%)	499/3613 (14%)	175/3613 (5%)	**2978/3613 (82%)**	43/3613 (1%)	624/3613 (17%)
neither	224/9000 (2%)	**2368/9000 (26%)**	195/9000 (2%)	1993/9000 (22%)	243/9000 (3%)	**5616/9000 (62%)**

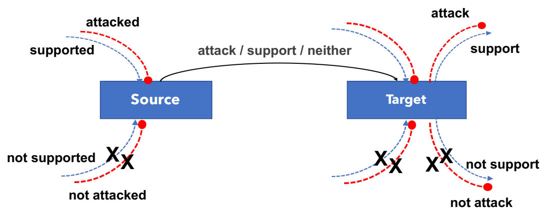

Figure 2: Macro-level structures examined in Kuribayashi et al. [3], shown as dotted lines. Red, oval arrow depicts an *attack* relation and blue, open arrow depicts a *support* relation.

tion and link identification. However, it still remains an open issue as to whether such higher-order features are useful for argumentative relation identification (classifying a relation into *attack*, *support* and *neither*).

The second approach for solving the task of argumentative relation identification is to formalize the task as a pairwise multi-class classification problem [4, 1]. Cocarascu et al. [1] use a Siamese Neural Network-based classifier with a Long Short-Term Memory [2], where the input feature vector is constructed from the information from input ACs only. Nguyen et al. [4] exploit discourse structure features for argumentative relation identification. Their work is closest to our work in the sense of using macro-level information. However, we focus on macro-level information constructed using argumentative relations in a document.

Kuribayashi et al. [3] show the potential effectiveness of argumentative context in argumentative relation identification. However, their experiments are preliminary because they used gold-information for extracting macro-level features. In this paper, we show the effectiveness of macro-level features without using gold-information and examine other types of argumentative context (i.e. macro-level features such as argumentative flow).

3 Data

This study uses the arg-microtexts (henceforth, *MT*) corpus [7], which contains 112 argumentative short texts (one paragraph each, 5.1 ACs on average). Each text consists of an argumentative topic (e.g. *Should Germany introduce the death penalty?*) and a monologue text discussing this topic (e.g. Figure 1). Each text is composed of segmented ACs, which have AC type (*claim* or *premise*), stance (*proponent* or *opponent*), and argumentative relations between ACs (*support, attack, rebut, undercut, normal, example*, or *add*). Following [7], we reduced *rebut* and *undercut* to *attack*, and *normal* and *example* to *support*. For *add* relations from AC_i to AC_j, we first create a link from AC_i to AC_k, the grandparent of AC_i, with the same relation from AC_j to AC_k. In our experiment, we use 174 attack relations, 290 support relations, and 2,000 neither relations (neither *attack* nor *support* relation) obtained by this conversion process.

4 Macro-level argumentative structure analysis

Kuribayashi et al. [3] analyzed typical macro-level structures on several corpora. In addition to the MT corpus, they analyzed the persuasive essay (henceforth, *PE*) corpus [11], which consists of 402 essays (5 paragraphs, 15 ACs on average) posted in online forums. They extracted all *support*, *attack*, and *neither* relations from each corpora. For notational convenience, we call the starting point of each relation a *source AC* (i.e. an AC which supports/attacks something) and the end point of each relation a *target AC* (i.e. an AC which is supported/attacked by the source AC). In addition, Kuribayashi et al. [3] defined a macro-level structure as *the combination of a relation type and the state of the source AC and target AC*, as illustrated in Figure 2. For the state of an AC, they considered the following properties:

- whether the source AC is attacked/supported by another AC
- whether the target AC is attacked/supported by another AC
- whether the target AC attacks/supports another AC

The results on MT and PE corpora are shown in Table 1. The skewed distribution indicates typical macro-level argumentative structures frequently used in each corpus. Kuribayashi et al. [3] also discovered that both corpora have similar tendencies on macro-level structures and provided a detailed discussion about the results.

5 Identifying argumentative relations with macro-level information

To evaluate the effectiveness of macro-level features in argumentative relation identification, we add macro-level features to a baseline model and compare the performance of the models in argumentative relation identification. To consider macro-level information, we use output (meta-features) of the subtask-level classifiers (introduced in Sec. 5.1) for other ACs. The main classifier considers the meta-features (e.g. an AC seems to be attacking something) of ACs over macro-range to predict local argumentative relations.

One merit of this approach for considering other argumentative relations is that the model does not become complex because it does not need dynamic features such as history of prediction of other argumentative relations.

5.1 Baseline models

Simple model (SM): As a baseline model, we use a simple logistic regression classifier. This classifier takes a pair of ACs as input and outputs the relation type between the pair. We represent the pair of ACs (input) as a binary-valued feature vector, following Peldszus and Stede [7]. We extract surface features such as lemma, part-of-speech tags, and segment length from the source, target, and their adjacent ACs. See the original paper [7] for further details.

Subtask-stacked model (SSM): Following Peldszus and Stede [7], we use a *subtask-stacked model* (slightly modified version of their work) as a baseline model. There are four subtasks which are closely related to argumentative relation identification: central claim identification (**cc**), role identification (**ro**), function classification (**fu**), and attachment classification (**at**). At first, they pre-train the following subtask-level classifiers:

- **cc** classifier: predicts whether an AC is *claim* or a *premise.*
- **fu** classifier: predicts whether an AC attacks something, supports something, or no function (if an AC is *claim*, it has no outgoing edge).
- **ro** classifier: predicts whether an AC's stance is *proponent* or *opponent* towards a central claim.
- **at** classifier: predicts whether two ACs have an argumentative relation or not.

Using the predicted probabilities $p(\cdot)$ of each subtask as meta-features, we train a logistic regression classifier (main classifier), which predicts an argumentative relation (*attack*, *support*, or *neither*). We define the meta-feature set for $w_{i,j}$ (relation from AC_i to AC_j) as the following:

$$\begin{aligned} sf_{i,j} = \Big\{ & p(\mathrm{cc}_i = \mathit{premise}),\, p(\mathrm{fu}_i = \mathit{attack}), \\ & p(\mathrm{fu}_i = \mathit{support}),\, p(\mathrm{ro}_i = \mathit{opp}) \times p(\mathrm{ro}_j = \mathit{pro}), \\ & p(\mathrm{ro}_i = \mathit{pro}) \times p(\mathrm{ro}_j = \mathit{opp}),\, p(\mathrm{ro}_i = \mathit{opp}) \times p(\mathrm{ro}_j = \mathit{opp}), \\ & p(\mathrm{ro}_i = \mathit{pro}) \times p(\mathrm{ro}_j = \mathit{pro}),\, p(\mathrm{at}_{i,j} = \mathit{yes}) \Big\} \end{aligned}$$

This model uses $sf_{i,j}$ as features for the main classifier.

5.2 SSM + macro model

Given our observation in Section 4, information about incoming and outgoing relations on a source and target AC will be helpful for predicting a relation. To consider incoming relations, we pre-train two additional classifiers. One classifier predicts whether a segment is attacked or not (**attacked** classifier), and the other predicts whether a segment is supported or not (**supported** classifier). Then, we use the output of the two classifiers for a source and target segment as a part of the macro-features. To consider outgoing relations of a target AC, we use the output of a function classifier on the target segment as additional macro-features. To summarize, we use the following as macro-level features:

$$\begin{aligned} mf_{i,j} = \Big\{ & p(\mathrm{fu}_j = \mathit{attack}),\, p(\mathrm{fu}_j = \mathit{support}), \\ & p(\mathrm{attacked}_i = \mathit{yes}),\, p(\mathrm{supported}_i = \mathit{yes}),\, p(\mathrm{attacked}_j = \mathit{yes}), \\ & p(\mathrm{supported}_i = \mathit{yes}) \Big\} \end{aligned}$$

This model uses $sf_{i,j} \cup mf_{i,j}$ as features for the main classifier.

5.3 SSM + macro + flow model

To consider additional argumentative context, we define the following features (probabilities) as argumentative flow:

- whether the adjacent ACs of source and target AC are attacked/supported by another AC
- whether the adjacent ACs of source and target AC attacks/supports another AC
- the argument component type (*claim* or *premise*) of the following; target AC, source-adjacent ACs, and target-adjacent ACs
- the stances of the following: source AC, target AC, source-adjacent ACs, and target-adjacent ACs

$$\begin{aligned} ff_{i,j} = \Big\{ & p(\mathrm{cc}_k = \mathit{claim}),\, p(\mathrm{cc}_j = \mathit{claim}), \\ & p(\mathrm{fu}_k = \mathit{attack}),\, p(\mathrm{fu}_k = \mathit{support}),\, p(\mathrm{attacked}_k = \mathit{yes}), \\ & p(\mathrm{supported}_k = \mathit{yes}),\, p(\mathrm{ro}_k = \mathit{opp}),\, p(\mathrm{ro}_k = \mathit{pro}),\, p(\mathrm{ro}_i = \mathit{opp}), \\ & p(\mathrm{ro}_i = \mathit{pro}),\, p(\mathrm{ro}_j = \mathit{opp}),\, p(\mathrm{ro}_j = \mathit{pro}) \mid k = i, j \pm 1 \Big\} \end{aligned}$$

This model use $sf_{i,j} \cup mf_{i,j} \cup ff_{i,j}$ as features for the main classifier.

6 Evaluation

6.1 Setting

We compare the performance of each model in the following two tasks:

(i) classifying whether a given segment pair has a *support* relation or not (*support* detection).
(ii) classifying whether a given segment pair has a *attack* relation or not (*attack* detection).

Table 2: Performance of all models in each task.

model	support vs non-support				attack vs non-attack			
	macro	support = yes			macro	attack = yes		
		f1	precision	recall		f1	precision	recall
Baseline	.601 ± .130	.355 ± .064	.312 ± .062	.457 ± .164	**.620 ± .031**	**.297 ± .059**	**.290 ± .059**	.315 ± .083
Baseline SS	.604 ± .030	.369 ± .063	.259 ± .045	.643 ± .113	.516 ± .118	.247 ± .049	.154 ± .034	.670 ± .129
SS + macro	**.638 ± .029**	**.419 ± .042**	**.301 ± .037**	**.698 ± .075**	.520 ± .119	.253 ± .047	.157 ± .033	.682 ± .123
SS + macro + flow	.627 ± .080	.400 ± .053	**.301 ± .048**	.615 ± .108	.536 ± .099	.266 ± .040	.166 ± .030	**.697 ± .108**

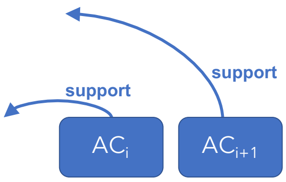

Figure 3: An AC (e.g. AC_{i+1}) which follows a supporting AC (e.g. AC_i) tends to support something.

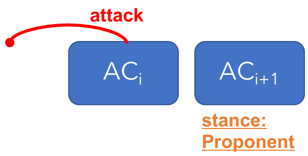

Figure 4: An AC (e.g. AC_{i+1}) following an attacking AC (e.g. AC_i) with a *proponent* stance.

We extract all AC pairs in MT[2] corpus and predict the relation type. All models are evaluated on 10 iterations of 5 × 3-fold nested cross validation. The reported results are average and standard deviation over the 10 folds. We tuned all hyperparameters using the inner 3-fold CV from the training data. We use macro F1 and F1 for each class as an evaluation metric. The subtask-level classifiers are also trained in the above regime.

6.2 Results

The results are shown in Table 2. For task (i), we found that our SSM+macro model outperforms the other models. This results indicate that macro-level information is useful for predicting *support* relations. For task (ii), all the models predict an *attack* relation with low f1 score. We attribute this to the fact that there are few *attack* relations in the MT corpus. On the other hand, we found that our model predicts an *attack* relation with high recall.

We speculate that the effectiveness of macro-level meta-features subsequently depend on the performance of the subtask-classifiers. Therefore, we speculate that our performance is hindered by mistakes in our sub-task classifiers. We examined the feature weights learned by our SSM+macro model and SSM+macro+flow model, we then found that they indicate a similar tendency to that of Kuribayashi et al. [3]'s analysis. In addition, we found flow-level tendencies such as the following:

1. An AC following a supporting AC tends to support. (see Figure 3)
2. An AC following an attacking AC has *for* stance. (i.e. even when a writer attacks with *against* stance, they switch their stance into *for* stance immediately.) (see Figure 4)

[2]In future work, we will also perform the experiments using the PE corpus.

7 Conclusions and future work

In this paper, we examined the usefulness of typical macro-level argumentative structures observed in argumentative texts and created a model for capturing macro-level information such as argumentative context. Our results showed that with the addition of macro-level features such as argumentative context, we can reasonably predict argumentative relations. In our future work, we will create a more sophisticated model capable of jointly learning the subtask-level classifiers and main classifier. Furthermore, we will exploit more corpora, such as the PE corpus, for evaluating our models.

Acknowledgement

This work was supported by JST CREST Grant Number JPMJCR1513 and JSPS KAKENHI Grant Number 16H06614 and 15H01702.

References

[1] Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. *ACL*, pages 1385–1390, 2017.

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

[3] Tatsuki Kuribayashi, Paul Reisert, Naoya Inoue, and Kentaro Inui. Examining macro-level argumentative structure features for argumentative relation identification. pages 1–6, 2017.

[4] Huy V Nguyen and Diane J Litman. Context-aware Argumentative Relation Mining. *ACL*, pages 1127–1137, 2016.

[5] Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument Mining with Structured SVMs and RNNs. *ACL*, pages 985–995, 2017.

[6] Andreas Peldszus and Manfred Stede. From Argument Diagrams to Argumentation Mining in Texts: a survey. *International Journal of Cognitive Informatics and Natural Intelligence*, 7(1):1–31, 2013.

[7] Andreas Peldszus and Manfred Stede. Joint prediction in MST-style discourse parsing for argumentation mining. *EMNLP*, pages 938–948, 2015.

[8] Isaac Persing and Vincent Ng. End-to-End Argumentation Mining in Student Essays. *NAACL*, pages 1384–1394, 2016.

[9] Peter Potash, Alexey Romanov, and Anna Rumshisky. Here's My Point: Joint Pointer Architecture for Argument Mining. *EMNLP*, pages 1375–1384, 2017.

[10] Yi Song, Michael Heilman, Beata Beigman Klebanov, and Paul Deane. Applying Argumentation Schemes for Essay Scoring. *Proceedings of the First Workshop on Argumentation Mining*, (2011):69–78, 2014.

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

[12] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700. 2015.

[13] Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. Using argument mining to assess the argumentation quality of essays. *COLING*, pages 1680–1691, 2016.