8th International Symposium on
Applied Engineering and Sciences (SAES2020)
Virtual Conference / 12th–19th December 2020

# Relation Identification Using Dialogical Features in Multi-Party Conversation

Takumi Himeno[1] * and Kazutaka Shimada[1]
[1] Department of Artificial Intelligence, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, JAPAN
* Corresponding author's e-mail: t_himeno@pluto.ai.kyutech.ac.jp
**Keywords:** Multi-party Conversation, Argument Mining, Argument Structure

**Introduction:** To understand the content of a discussion in meetings, a summary is important for people who did not attend the discussion. If the summary is illustrated as a discussion structure, it is helpful to grasp the essentials of the discussion immediately. In this paper, we construct models that predict a link between nodes that consist of some utterances in multi-party conversation. Various neural network models have been proposed for many tasks. However, those neural network models can usually capture only the linguistic feature. On the other hand, dialogical features are useful for understanding conversations. Therefore, we introduce some dialogical features to our method.

**Approach:** We utilize three models based on machine learning for the prediction of a link between nodes, namely SVMs, Bi-LSTM and BERT. The basic feature for the three models is based on linguistic features; BOW for SVMs and Word2Vec for the three models. We assume that a model with dialogical features leads to the improvement of the accuracy as compared with the models with only linguistic features, for the multi-party conversation settings. Therefore, we create eight types of dialogical features such as time information, for the three models. In this paper, we use 219 discussion from 92 dialogs of the AMI corpus [1]. We divide the AMI corpus into 201 discussions from 84 dialogs for the training data, 13 discussions from 4 dialogs for the development data, and 12 discussions from 4 dialogs for the evaluation data. However, the experimental data are imbalanced. The linked pairs are just 3850 as against 38530 not-linked pairs. Models generated from imbalanced data usually become a weak classifier. Therefore, we randomly select 3850 not-linked pairs from the original training data for all models. Then, we generate each model from the downsized and balanced training data.

**Results and Discussion:** We compared the effectiveness of our proposed features. Table 1 and Table 2 show the experimental results with our features and that without our features, respectively. All the F-scores of the models with our features were improved as compared with those without our features. This result shows the effectiveness of our dialogical features. By applying our dialogical features, our model dramatically boosted all scores about not-linked pairs. Next, we analyzed errors in the experimental results. One typical error was nodes with back-channel feedback. The back-channel nodes were often incorrectly connected with nodes about questions. Nodes consisting of some words or short phrases, such as back-channel feedback, are ambiguous, and they tended to connect with all sorts of nodes. Creating new dialogical features to capture this problem is the most important future work.

| | Linked pair | | | Not-linked pair | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| $SVM_{W2V}$ | **0.41** | 0.84 | **0.56** | **0.98** | 0.85 | 0.91 |
| $SVM_{BOW}$ | 0.39 | 0.89 | 0.55 | 0.98 | 0.82 | 0.90 |
| *Bi-LSTM* | 0.42 | 0.46 | 0.44 | 0.93 | **0.92** | **0.92** |
| *BERT* | 0.27 | **0.90** | 0.41 | 0.98 | 0.69 | 0.81 |

**Table 1:** Result of methods with our proposed feature.

| | Linked pair | | | Not-linked pair | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| $SVM_{W2V}$ | 0.12 | 0.55 | 0.20 | 0.90 | 0.49 | 0.64 |
| $SVM_{BOW}$ | 0.13 | 0.53 | 0.22 | 0.91 | 0.57 | 0.70 |
| *Bi-LSTM* | 0.15 | 0.55 | 0.24 | 0.91 | **0.61** | **0.73** |
| *BERT* | **0.16** | **0.60** | **0.25** | **0.92** | 0.60 | **0.73** |

**Table 2:** Result of methods without proposed feature.

[1] Carletta, Jean, et al. "The AMI meeting corpus: A pre-announcement." *International workshop on machine learning for multimodal interaction*. Springer, Berlin, Heidelberg, 2005.

# Relation Identification Using Dialogical Features in Multi-Party Conversation

**Takumi Himeno and Kazutaka Shimada**
Department of Artificial Intelligence
Kyushu Institute of Technology
Fukuoka, Japan
{t_himeno, shimada}@pluto.ai.kyutech.ac.jp

## Abstract

To grasp the content of a discussion in meetings, a summary is important for people who could not attend the discussion. If the summary is illustrated as a discussion structure, it is helpful to grasp the essentials of the discussion immediately. In this paper, we construct models that predict a link between nodes that consist of some utterances in multi-party conversation. Various neural network models have been proposed for many tasks. However, those neural network models can usually capture only the linguistic feature. On the other hand, dialogical features are useful for models to predict a link between nodes. First, we explain the features that we design for the task. Next, we report the result in which we compared a machine learning method using the proposed feature with one without them. The result shows that time information and distance between nodes were effective.

## 1 Introduction

Meetings are often held in laboratories and companies to come up with new research ideas and management strategies. To grasp the content of a discussion, a summary is important for people who could not attend the discussion. A summary is suitable for understanding the main discussion points. Assume that a summary is illustrated as a discussion structure. The summary is more powerful and helpful to understand the main discussion points because users can immediately capture the flow of the discussion by using links between utterances For the purpose, we need to estimate the discussion structure from each discussion.

Argument mining is one of the tasks to construct a structure of sentences (Stab and Gurevych, 2017a). It automatically derives the structure of argumentation from unstructured documents such as essays. It consists of the four subtasks as follows: component identification, component classification, relation identification, and relation classification. Component identification is the task that extracts argument components from a given document. Argument components denote sentences and paragraphs that related to the discussion structure. Component classification is the task that assigns a label, e.g., claim, to each argument component. Relation identification is the task that predicts whether each pair of argument components is related or not. Relation classification is the task that assigns a label, such as "attack" and "support", to the related pairs of argument components. In this paper, we focus on relation identification for constructing the discussion structure in a multi-party conversation. In other words, we construct a prediction model of links between nodes consisting of some utterances.

In recent years, neural network models are used in many tasks of natural language processing. In argument mining, some researchers have proposed models based on neural networks (Deguchi and Yamaguchi, 2019; Qin et al., 2017). We also apply neural models to our task. However, these neural network models tend to use only linguistic features. On the other hand, dialogical features, such as time information, are useful for models to predict a link between nodes. Therefore, we introduce some dialogical features to our method.

The contributions of this paper are as follows: (1) we show the effectiveness of our dialogical features in the relation identification task with the AMI corpus and (2) we compare SVMs, Bi-LSTM, and BERT models with/without the dialogical features.

## 2 Related Work

Automatic summary generation is one of the most important studies to help people that want to understand the main discussion points easily. Mehdad et al. (2013) have proposed a method that automatically generates a summary of the content of a discussion by extracting important utterances in the discussion. Although a summary is suitable for grasping the content of the whole discussion, it is not always suitable to understand the structure of discussion points immediately. In this paper, we focus on a prediction task of links between utterances for grasping the structure of the discussion.

In recent years, argument mining is attracting attention in natural language processing. Argument mining is a task to construct the structure of a document. It is applied to many natural language processing tasks such as document summarization (Barker and Gaizauskas, 2016; Peldszus, 2014), the automatic scoring of essays (Ghosh et al., 2016), the paper writing support (Stab and Gurevych, 2017b; Nguyen and Litman, 2016), the information retrieval (Stab et al., 2018) and so on. Stab and Grevych (2014) have tackled the relation identification for essays written by students. They created some features that capture the characteristics of the essay and predicted the link between argument components. The essay is usually formalized, such as the form of a claim followed by premises. However, the multi-party conversation is not formalized because many people freely speak to assert their opinions. Therefore, we introduce some features that capture the dialogical characteristics, such as time information, for the multi-party conversation task.

As a method that predicts the link between argument components, Potash et al. (2017) have proposed a method based on Pointer Network (Vinyals et al., 2015). They applied some features, namely BOW and embedding, for the task. However, as mentioned above, not only linguistic features but also dialogical features are important for the multi-party conversation task. In this paper, we prepare some features to capture the characteristics of the multi-party conversation: for example, speaker ID, time information, and distance between nodes.

## 3 Dataset and Task

### 3.1 Dataset

In this paper, we use the AMI corpus, a multi-party conversation corpus (Carletta et al., 2005). It contains useful various annotations, such as the argument structure and time information, to predict a link between nodes. Each node consists of one or more utterances. We use scenario meetings that are held with the discussion points given in advance. In the discussion setting, four employees in different roles in a company discuss developing a new TV remote control that replaces an old-style TV remote control for consumers on the market. This discussion is held four times. Each utterance in the AMI corpus contains speaker ID, time information, and a dialog act. A dialog act indicates what intention each utterance represents. The details of the dialog act are shown in Table 1. This corpus is annotated with a total of 15 types of dialog act tags such as "Inform" representing an utterance providing some information and "Backchannel" giving responses in a word. In this paper, we introduce dialogical features by using these tags.

In this paper, we use the annotated data[1] based on the Twente argument schema (TAS) to contain the link between nodes (Rienks et al., 2005). TAS is an annotation schema created to clarify the discussion structure which arises from the scenario meeting of the AMI corpus. The discussion structure in TAS consists of two elements. One is the node and the other is the edge. The node in TAS contains parts of, or even complete, speaker turns. The edge in TAS represents the type of relation between the nodes. In TAS, the Unit Label that represents the role of the node is also annotated The details of the Unit Label are shown in Table 2. In addition, TAS defines "discussion" as segments in the meeting ("Dialog"). One dialog consists of one or more discussions, one discussion consists of some nodes and one unit label is assigned to each node.

[1] http://groups.inf.ed.ac.uk/ami/download/

| Tag Type | Detail |
|---|---|
| Backchannel | Utterances such as giving the response |
| Stall | Utterances playing as the filled pauses |
| Fragment | Utterances which do not convey a speaker intention |
| Inform | Utterances in which the speaker gives information to the listener |
| Elicit-Inform | Utterances in which the speaker asks the listener for information |
| Suggest | Utterances in which the speaker expresses an intention about actions |
| Offer | Utterances offering the speaker's own action |
| Elicit-Offer-or-suggestion | Utterances in which the speaker expresses a desire for someone to make an offer or suggestion. |
| Assess | Utterances about any comment that expresses an evaluation |
| Comment-about-Understand | Utterances showing the speaker's own understanding |
| Elicit-Assessment | Utterances in which the speaker attempts to elicit an assessment |
| Elicit-Comment-about-Understanding | Utterances in which the speaker attempts to elicit a comment |
| Be-Positive | Utterances which are intended to make an individual or the group happier |
| Be-Negative | Utterances which express negative feelings towards an individual or the group |
| Other | Utterances which don't fit any of the other classes |

Table 1: Detail of dialog act tags in the AMI corpus.

### 3.2 Task

Figure 1 shows an example of the relation identification process. In Figure 1, the dialog contains two discussions: discussion1 and discussion2. The two discussions contain three nodes and two nodes respectively. ♦ in each discussion denotes true links. For example, the pairs of $node1$-$node2$ and $node1$-$node3$ contain the link that we want to predict.

First, we extract all combinations of two nodes in each discussion. In Figure 1, three pairs are extracted from discussion1 and one pair of $node4$ and $node5$ is extracted from discussion2. Next, our model classifies each pair into linked or not-linked pairs. We evaluate whether the result corresponds to the ground truth.

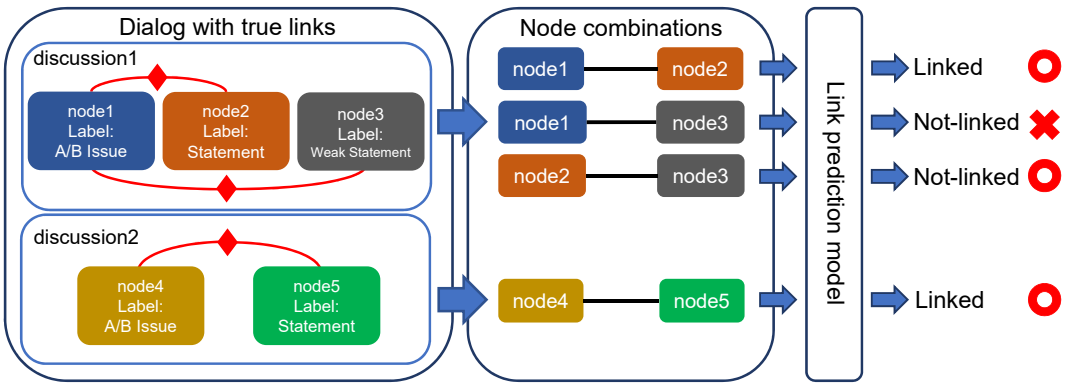


Figure 1: Relation identification. We handle all combinations of node pairs in each discussion for the relation identification task. The task is to classify each pair into "linked" or "not-linked".

| Tag Type | Detail |
|---|---|
| Statement | A claim without a weakening qualifier |
| Weak Statement | A claim with a weakening qualifier |
| Open Issue | An issue that are raised where every possible response could be a solution |
| A/B Issue | An issue that are raised where the possible responses are explicitly enumerated |
| Yes/No Issue | An issue that are raised where the possible responses are Yes and No |
| Other | Not fitting any of the other Unit Labels |

Table 2: Detail of the unit labels in the TAS.

## 4 Proposed Method

In this section, we explain three models based on machine learning for the prediction. First, in Section 4.1, we explain the models as follows; SVM, Bi-LSTM, and BERT. The basic feature for the three models is based on linguistic features; BOW for SVMs and Word2Vec, namely word embedding, for the three models. Then, in Section 4.2, we explain our proposed features based on dialogical characteristics.

### 4.1 Basic models with linguistic features

#### 4.1.1 SVM

We apply Support Vector Machines (SVMs) to the prediction task. Here SVMs handle two types of feature spaces; BOW features and features based on word embedding. As the BOW features, we use all words without the stopword list by NLTK. As the word embedding, we use word2vec (W2V) [2] published by Google. We generate the vector space as follows:

$$V_{node_n} = \sum_{x=1}^{m} v_x \tag{1}$$

where $v_x$ denotes the word vectors of $node_n$ and $m$ denotes the size of the $node$. For example, assume that $node1$ and $node2$ consist of embedding ($v_x$) of words in each node. We obtain two summed word embedding vectors, namely $V_{node1}$ and $V_{node2}$. Finally, SVMs learn and predict the relation by using concatenated $V_{node1}$ and $V_{node2j}$.

#### 4.1.2 Bi-LSTM

The second model is based on the Bi-LSTM model to predict the relation (Hochreiter and Schmidhuber, 1997). Figure 2 shows an overview of the Bi-LSTM model. We use W2V in each node as the input for the model. The W2Vs in $node_i$ and $node_j$ are learned by Bi-LSTM, respectively. The model concatenates $V_{node_i}$ and $V_{node_j}$ that are obtained from the Bi-LSTM. Finally, the softmax function produces the estimated probability of the relation from the dense layer[3].

#### 4.1.3 BERT

The third model is based on BERT (Devlin et al., 2018). The BERT model transforms the input word vector based on W2V in two nodes into feature vectors. We extract the 11th layer from the BERT since the last layer tends to be too close to the target functions during pretraining. It is often biased toward the target. Finally, we compute the softmax function from the dense layer by the 11th layer[4].

[2] http://code.google.com/archive/p/word2vec/

[3] As mentioned in Figure 2, we concatenate the proposed features described in Section 4.2 to the dense layer if we provide the option

[4] In a similar to Bi-LSTM, We also concatenate the proposed features in the next section to the dense layer if we provide the proposed features.

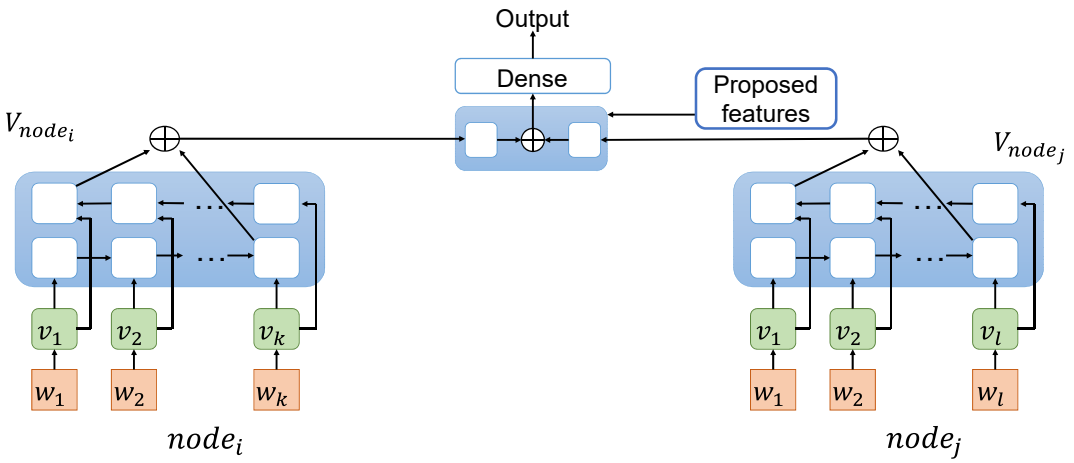


Figure 2: The Bi-LSTM and dense network. We concatenate our proposed features with the outputs of LSTMs.

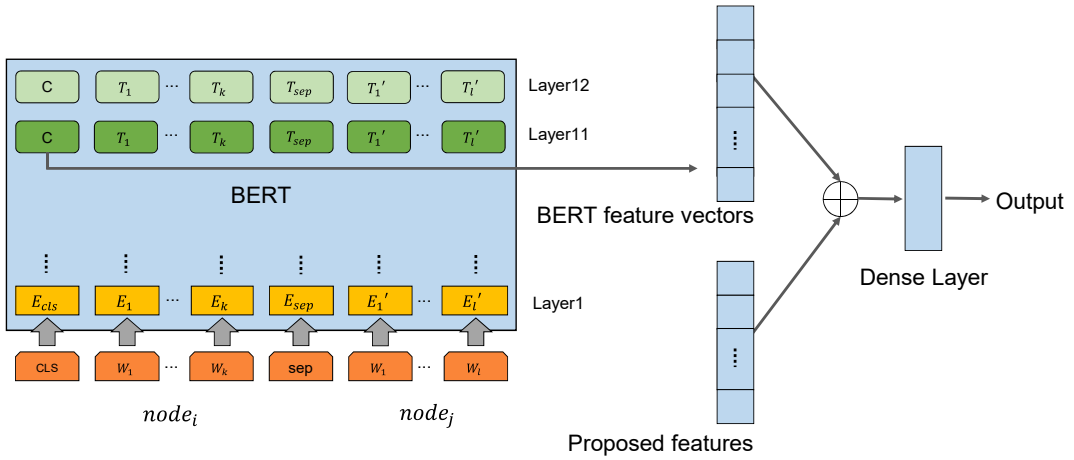


Figure 3: The BERT model. We use the 11th layer for integration with our features.

### 4.2 Proposed Features

A model with dialogical features leads to the improvement of the accuracy as compared with the models with only linguistic features, namely models in Section 4.1, for the multi-party conversation settings. In this section, we explain eight types of dialogical features for our model. For the model based on Bi-LSTM and BERT, Figure 2 and Figure 3 have already indicated the usage instructions of our proposed features in this section. For the SVMs, we concatenate the proposed features in this section to the vector space based on BOW or W2V.

- Number of words in node pair (NW)

  If the speaker supports and attacks the other speaker's claim, the size of the node tends to be larger[5]. In a similar way, the node also tends to be larger if the speaker wants to convey a lot of information to the other speaker. On the other hand, the size of the node becomes smaller if the node consists of short utterances, such as back-channel feedback. Thus, the size of each node is one of the important characteristics. To capture this feature, we use the number of words in each node.

- Number of common words in node pairs (NCW)

  If two nodes are related to a common topic, words in them frequently are overlapped. Therefore, we count the number of common words that appear in each node as the feature.

[5]The size denotes the number of utterances in a node in this context.

- Speaker information (SI)

  If a speaker claims own opinion, a speaker who gives a positive opinion and points out a problem about the opinion differs from him/her. Besides, the situation that the same speaker gives a positive opinion to own claim or point out a problem of his/her claim is very rare. Therefore, speaker information of each node has an important role in the relation between two nodes. We use the speaker ID of each node as the feature.

- Time information (TI)

  If the discussion is active, the time interval between nodes tends to become shorter. Links between a node in the early stage and a node in the last stage in a discussion is rare. In other words, far-flung nodes usually do not possess a link. To capture this feature, we focus on time information in the corpus. We compute the time information by using the end time of a node and the start time of another node as the feature.

- Distance between nodes (DN)

  Assume that the discussion is stagnant. In this situation, the distance between nodes becomes short because the number of nodes in the stagnant situation becomes small[6]. Thus, the distance, namely the number of nodes that appear between two nodes, is one important feature. Therefore, we sort the nodes in a discussion in terms of the timestamps and use the distance between nodes as the feature.

- Dialog act (DA)

  Dialog act tags are important information for the prediction model. For example, if a node contains the Inform tag, the node tends to connect with nodes that contain "Backchannel" and "Assess" because of the nature of discussions. On the other hand, a node with the Elicit-inform tag does not usually connect with the Inform tag because the Elicit-inform tag is used by a speaker to request that someone else give some information while the Inform tag is used by a speaker to give information. Therefore, we use the distribution of 15 types of dialog acts in each node as the feature.

- Unit label (UL)

  The unit labels described in Section 3 also have an important role in the prediction of the link between nodes. They contain three types of labels that are related to questions; "Open Issue", "A/B Issue", and "Yes/No Issue". In the situation that a node contains such tags, the node tends to connect with nodes that express positive/negative opinions. Besides, nodes with such tags do not generally connect with nodes about questions because it is a question-question pair. Therefore, we use the unit label of each node as the feature.

- Polarity of node pair (P)

  Emotional information is also one of the characteristics of conversations. For example, a speaker may emotionally argue while claiming his/her opinion in a discussion. In a similar way, when a speaker may emotionally argue when he/she agrees or disagrees with another speaker's question. To capture the information, we use Stanford CoreNLP (Manning et al., 2014). We compute the score (1 to 5) of each utterance by using CoreNLP. Then we compute the average score from the score of the utterances in each node. We use the average polarity score of each node as the feature.

## 5 Experiment and Analysis

In this section, we describe the experimental setup of the three models in Section 4.1. Next, we describe the experimental data. Finally, we report the experimental result and discuss the influence of the proposed features.

[6]Note that this distance denotes the number of lines when each utterance is transcribed by one line. This is essentially different from the time information feature.

| | Dialog | Discussion | Linked pair | Not-linked pair |
|---|---|---|---|---|
| Training data | 84 | 201 | 3850 | 38530 |
| Development data | 4 | 13 | 235 | 1822 |
| Evaluation data | 4 | 12 | 238 | 1875 |

Table 3: Distribution of the experimental data. For the training data, we select 3850 not-linked pairs randomly to generate balanced training data.

| | Linked pair | | | Not-linked pair | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-score | Precision | Recall | F-score |
| $SVM_{W2V}$ | **0.41**$^{\dagger}$ | 0.84$^{\dagger}$ | **0.56**$^{\dagger}$ | **0.98**$^{\dagger}$ | 0.85$^{\dagger}$ | 0.91$^{\dagger}$ |
| $SVM_{BOW}$ | 0.39$^{\dagger}$ | 0.89$^{\dagger}$ | 0.55$^{\dagger}$ | 0.98$^{\dagger}$ | 0.82$^{\dagger}$ | 0.90$^{\dagger}$ |
| $Bi\text{-}LSTM$ | 0.42$^{\dagger}$ | 0.46 | 0.44$^{\dagger}$ | 0.93$^{\dagger}$ | **0.92**$^{\dagger}$ | **0.92**$^{\dagger}$ |
| $BERT$ | 0.27$^{\dagger}$ | **0.90**$^{\dagger}$ | 0.41$^{\dagger}$ | 0.98$^{\dagger}$ | 0.69$^{\dagger}$ | 0.81$^{\dagger}$ |

Table 4: Result of methods with our proposed feature. Each † denotes that the score is better than that of the baseline in Table 5, namely the effectiveness of our method.

### 5.1 Experimental Setup

For the SVM model, we use LiBSVM (scikit-learn) for the implementation (Chang and Lin, 2011). The kernel function was the RBF function and the cost parameter was 100.

For the Bi-LSTM, the input layer dimensions were set to 300. The batch size was 16. NLLoss was used as the loss function. The optimizer was Adam (Kingma and Ba, 2015); the learning rate was 0.001 and the drop out was 0.2. When the Bi-LSTM model learned the training data, the Bi-LSTM model evaluated the development data for each epoch at the same time. The smallest value of the loss function in the development data appeared in 25 epochs. Therefore, we used the model as the final model.

For the BERT model, we used the BERT-Base as the pre-trained model. The text has been lowercased. The batch size was 16. We used cross-entropy as the loss function. The optimizer was Adam and the learning rate was 0.00002.

### 5.2 Experimental Data

We used 219 discussions from 92 dialogs of the AMI corpus. In this experiment, all nodes in each discussion were given and we used oracle unit labels in the corpus for the feature extraction. We divided the AMI corpus into 201 discussions from 84 dialogs for the training data, 13 discussions from 4 dialogs for the development data, and 12 discussions from 4 dialogs for the evaluation data. As explained in Section 3.2, we generated all combinations of two nodes in each discussion. The distribution, such as the number of linked pairs and not-linked pairs, was shown in Table 3.

Table 3 said that the experimental data were imbalanced; the linked pairs were just 3850 as against 38530 not-linked pairs. Models generated from imbalanced data usually become a weak classifier. Therefore, we reduced the imbalance of the training data. For all models, we randomly selected 3850 not-linked pairs from the original training data. Then, we generated each model, namely SVMs, Bi-LSTM,

| | Linked pair | | | Not-linked pair | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-score | Precision | Recall | F-score |
| $SVM_{W2V}$ | 0.12 | 0.55 | 0.20 | 0.90 | 0.49 | 0.64 |
| $SVM_{BOW}$ | 0.13 | 0.53 | 0.22 | 0.91 | 0.57 | 0.70 |
| $Bi\text{-}LSTM$ | 0.15 | 0.55 | 0.24 | 0.91 | **0.61** | **0.73** |
| $BERT$ | **0.16** | **0.60** | **0.25** | **0.92** | 0.60 | **0.73** |

Table 5: Result of methods without our proposed feature.

| Proposed Feature | Precision | Recall | F-score |
|---|---|---|---|
| $ALL$ | 0.41 | 0.84 | 0.56 |
| $-$NW | 0.40 | 0.86 | 0.55 |
| $-$NCW | 0.41 | 0.84 | 0.55 |
| $-$SI | 0.40 | 0.84 | 0.54 |
| $-$TI | **0.30** | 0.84 | **0.44** |
| $-$DN | 0.38 | 0.85 | 0.52 |
| $-$DA | 0.41 | 0.84 | 0.55 |
| $-$UL | 0.41 | 0.84 | 0.55 |
| $-$P | 0.40 | 0.84 | 0.55 |

Table 6: Ablation test. This result was based on the best model, namely $SVM_{W2V}$.

and BERT, from the downsized and balanced training data.

### 5.3 Experimental Results

We compared the effectiveness of our proposed features. Table 4 and Table 5 show the experimental result with our features and that without our features, respectively. The boldface denotes the best score for each criterion, namely Precision, Recall, and F-score, in the table. The scores with † in Table 4 denote that the scores are better than those of the models without our features. All the F-scores of the models with our features were improved as compared with those without our features. This result shows the effectiveness of our dialogical features. By applying our dialogical features, our model dramatically boosted all scores about not-linked pairs. The features contributed to reducing fallacious edges by the basic models based on only the linguistic feature.

Recall rates of linked pairs of $SVM_{BOW}$ and $BERT$ were higher than $SVM_{W2V}$ although the best F-score was produced by $SVM_{W2V}$. For this reason, $W2V$ lost information of specific words that were effective to the prediction due to the summation of word embeddings (Eq. (1)). On the other hand, $BOW$ and $BERT$ can handle such information correctly.

However, the precision rates of the linked pairs were not sufficient as compared with other criteria. The best score was just 0.41 on the $SVM_{BOW}$ model. Our models tended to become overly committed to the linked pairs. In other words, our models generated too many edges between linked pair nodes. On the other hand, there are some restrictions about true links; only one incoming arrow to a node is permitted although some outward arrows are permitted. The optimization of outputs of our models is interesting future work.

Next, we analyzed errors in the experimental result. One typical error was nodes with back-channel feedback. The back-channel nodes often incorrectly connected with nodes about questions. Nodes consisting of some words or short phrases such as back-channel feedback are ambiguous and tend to connect with all sorts of nodes. The improvement of this problem is the most important future work. Moreover, nodes with long utterances tended to be misrecognized. Although such nodes contained a lot of information, our models were not able to capture the characteristics. This leads to the problem of the relatively low recall rate of $SVM_{W2V}$. Specific expressions contribute to solving this problem. For example, adverbs and auxiliary verbs are often used to emphasize the claim. Some conjunctions are also used to support the explanation of the claim and assist the claim. Capturing such linguistic characteristics and surface expressions by additional features for the prediction is still important future work.

### 5.4 Analysis of Feature

We evaluated our features by the ablation test. The model in this ablation test is $SVM_{W2V}$, the best F-score.

Table 6 shows the experimental result. $ALL$ denotes the model with all features; the same as the scores in Table 4. "$-$" denotes the ablation; e.g., $-$NW denotes the proposed model without the NW (Number of words in node pair) feature. If the score decreases, the feature is important. The most important feature

was the TI (Time Information) feature. The result was probably caused by a dialogical assumption that participants tend to immediately utter additional opinions or show a reaction after a participant's opinion.

The DN (Distance between nodes) was also relatively effective for the prediction. Although the concept of DN is similar to that of TI, they are complementary; TI is for active discussions and DN is for stagnant discussions. Such dialogical features that were not captured from words themselves, namely linguistic features, were important for argument mining on multi-party conversations.

## 6 Conclusions

In this paper, we proposed three models with additional features for the relation identification task on argument mining. Our target was the multi-party conversation corpus. We applied eight dialogical features into SVMs, Bi-LSTM, and BERT models. We compared the models with our features and those without our features. Our models outperformed the basic model that did not handle our features. The best model was SVMs with W2V and our features. We obtained the high recall rates for linked pairs although the precision rates were insufficient. The improvement of the precision rate by additional features and the optimization of the model's output is important future work.

We also evaluated our features by the ablation test. The best feature was the time information feature and the second one was the distance feature. These features are complementary for active discussions and stagnant discussions. These results show the effectiveness of our proposed methods.

## References


Emma Barker and Robert Gaizauskas. 2016. Summarizing multi-party argumentative conversations in reader comment on news. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 12–20.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

C. Chang and C. Lin. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Mamoru Deguchi and Kazunori Yamaguchi. 2019. Argument component classification by relation identification by neural network and textrank. In *Proceedings of the 6th Workshop on Argument Mining*, pages 83–91.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Debanjan Ghosh, Aquila Khanam, Yubo Han, and Smaranda Muresan. 2016. Coarse-grained argumentation features for scoring persuasive essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–554.

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

D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Raymond T. NG. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, Sofia, Bulgaria, August. Association for Computational Linguistics.

Huy Nguyen and Diane Litman. 2016. Context-aware argumentative relation mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137.

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

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 1364–1373, Copenhagen, Denmark, September. Association for Computational Linguistics.

Kechen Qin, Lu Wang, and Joseph Kim. 2017. Joint modeling of content and discourse relations in dialogues. *arXiv preprint arXiv:1705.05039*.

Rutger Rienks, Dirk Heylen, and Erik van der Weijden. 2005. Argument diagramming of meeting conversations. In *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces*, pages 85–92.

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.

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

Christian Stab and Iryna Gurevych. 2017b. Training argumentation skills with argumentative writing support. In *Proc. SEMDIAL 2017 (SaarDial) Workshop on the Semantics and Pragmatics of Dialogue*, pages 166–167.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium, October-November. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700.