Discussion Structure Prediction Based on a Two-step Method

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

Abstract

Conversations are often held in laboratories and companies. A summary is vital to grasp the content of a discussion for people who did not attend the discussion. If the summary is illustrated as an argument structure, it is helpful to grasp the discussion's essentials immediately. Our purpose in this paper is to predict a link structure between nodes that consist of utterances in a conversation: classification of each node pair into "linked" or "not-linked." One approach to predict the structure is to utilize machine learning models. However, the result tends to over-generate links of nodes. To solve this problem, we introduce a two-step method to the structure prediction task. We utilize a machine learning-based approach as the first step: a link prediction task. Then, we apply a score-based approach as the second step: a link selection task. Our two-step methods dramatically improved the accuracy as compared with one-step methods based on SVM and BERT.

1 Introduction

Meetings are often held in laboratories and companies to come up with new research ideas and management strategies. A summary is vital to grasp the content of a discussion for people who did not attend the discussion. Summaries are suitable for understanding the main points in discussions. Assume that a summary is illustrated as a discussion structure. The summary is more powerful and helpful to understand the main points in the discussion because users can immediately capture the flow of the discussion by using links between utterances. For this purpose, we need to predict the discussion structure of 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 four subtasks as follows: component identification, component classification, relation identification, and relation classification. Component identification is a task that extracts argument components from a given document. Argument components denote sentences and paragraphs related to the discussion structure. Component classification is a task that assigns a label, e.g., claim, to each argument component. Relation identification is a task that predicts whether each pair of argument components is related or not. Relation classification is a task that assigns a label, such as "attack" and "support," to the related pairs of argument components.

In this paper, we deal with relation identification for constructing a discussion structure in a multi-party conversation. In other words, we construct a link prediction model for nodes consisting of some utterances. Methods in previous work often predicted the discussion structures by using machine learning approaches, such as neural networks. Himeno and Shimada (2020) have reported that such machine learning models tended to over-generate links between nodes. Here we focus on some rules in discussion structures; e.g., a child node has one parent note. We incorporate score-based selection rules with the machine learning model as post-processing to improve the accuracy. We introduce top-down and bottom-up approaches for selecting edges. The result shows that the proposed methods are more accurate than the method without the selection.

2 Related Work

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

Proceedings of Recent Advances in Natural Language Processing, pages 538–546
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_061

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 Gurevych (2014) have tackled the relation identification for essays written by students. They created some features capturing the characteristics of the essay and then predicted links between argument components. The essay is usually formalized, such as the form of a claim followed by premises. However, multi-party conversations are not always formalized because many people freely speak to assert their opinions.

Discussion structures can be regarded as a kind of graph structure. In fact, some studies handled the selection of edges of argument structures as the shortest path problem of graphs (Dijkstra et al., 1959; Gabow and Tarjan, 1985). The shortest path problem is an optimization problem to find the path with the minimum weight among the paths connecting two given nodes in a weighted graph. In the shortest path problem, the method usually optimizes the entire path from parent nodes to multiple terminal nodes. We also apply this idea to our task, namely relation identification. However, it is not always suitable to adapt our task directly because there are some conditions in discussions: e.g., only one parent node for a child node. In this paper, we propose a link selection method for local utterance pairs to handle the characteristics of the discussions.

In addition, many studies have focused on the visualization of discussions (Chamberlain et al., 2018; Lugini et al., 2020). In this paper, we also visualize predicted links to understand the result easily.

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 various useful 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. One discussion is held four times. Each utterance in the AMI corpus contains speaker ID, time information, and a dialog act.

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

Table 1: Detail of the unit labels in the TAS.

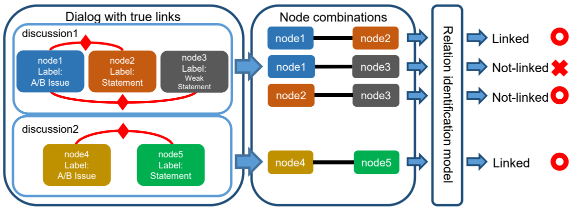

Figure 1: Relation identification. We handle all combinations of node pairs in each discussion for the relation identification task. The task is to classify each pair into "linked" or "not-linked."

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 a node, and the other is an 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, unit labels that represent the role of the node are also annotated. The details of the unit labels are shown in Table 1. 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. One unit label is assigned to each node.

3.2 Task

Figure 1 shows an example of relation identification in this paper. In Figure 1, the dialog contains two discussions: discussion1 and discussion2. The two discussions contain three nodes and two nodes,

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

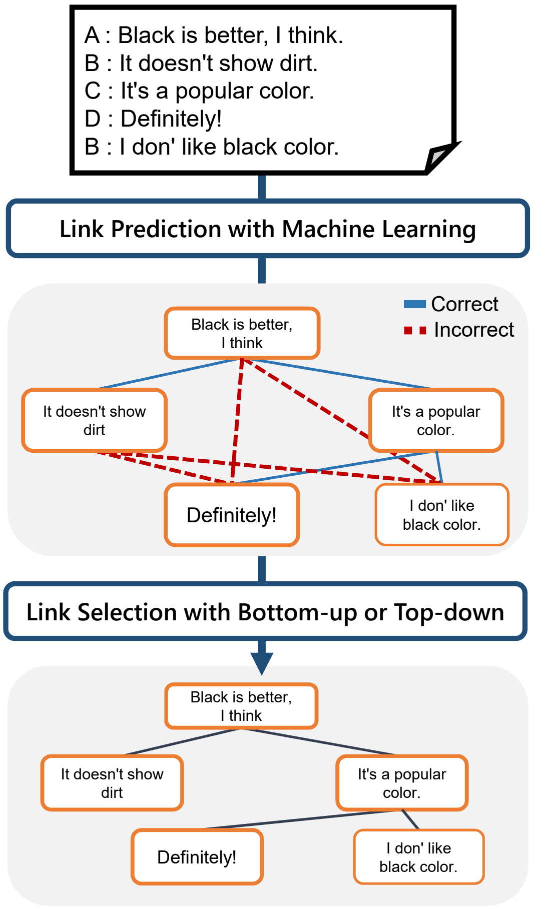

Figure 2: The proposed method with two steps. Our method distinguishes appropriate links (the blue lines: correct) from over-generated links (red lines: incorrect) from the link prediction model by machine learning models.

respectively. ♦ in each discussion denotes true links. For example, the pairs of $node_1$-$node_2$ and $node_1$-$node_3$ 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 $node_4$ and $node_5$ is extracted from discussion2. Next, our model classifies each pair into linked or not-linked pairs. Then, we apply a two-step method that is explained in the next section. We evaluate whether the result corresponds to the ground truth.

4 Proposed Method

In this section, we explain our method with two steps: link prediction and link selection. Figure 2 shows an overview of our method. The first step (link prediction) is based on machine learning techniques. We compare two models; one is Support Vector Machines (Vapnik, 2000) with word embeddings and selected features, and the other is BERT (Devlin et al., 2019). However, the models tend to over-generate links between nodes in this task. Hence, we introduce link selection approaches as the second step: top-down and bottom-up approaches.

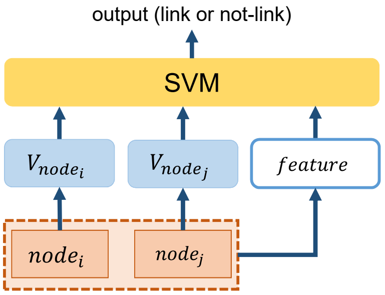

Figure 3: The link prediction model with SVM.

4.1 Link prediction

4.1.1 SVM

We apply Support Vector Machines (SVMs) to the link prediction task. Figure 3 shows the outline of the SVM-based model. It learns the link prediction model with two embeddings of two nodes and eight features. As the word embedding, we use word2vec (W2V) [2] published by Google. We generate the vector space as follows[3]:

$$V_{node_n} = \sum_{x=1}^{m} v_x \qquad (1)$$

where v_x denotes the word vectors of $node_n$ and m denotes the size of the $node_n$. For example, assume that we predict the relation between $node_i$ and $node_j$ consisting of some word embeddings (v_x). We obtain two summed word embedding vectors, namely V_{node_i} and V_{node_j} from $node_i$ and $node_j$. Finally, SVMs learn and predict the relation by using concatenated V_{node_i} and V_{node_j}.

We also utilize eight features as follows:

- Number of words in node pair

  If a speaker supports and attacks another speaker's claim, the size of the node tends to be larger[4]. In a similar way, the node also tends to be larger if the speaker wants to convey much information to the other speakers.

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

[3] we eliminate some words by the stopword list of NLTK.

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

On the other hand, the size of a 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

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

- Speaker information

  Agreement or negative statements to an opinion from a speaker tends to be uttered from another speaker. Besides, the situation that the same speaker gives a positive opinion to his/her own claim or points out a problem of his/her claim is very rare. Therefore, the 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

  If the discussion is active, the time interval between nodes tends to become shorter. As another example, a link 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. Thus, time information between nodes has an important role. To capture this feature, we focus on time information in the corpus. We compute the time information by using the end time of $node_i$ and the start time of $node_j$ as the feature.

- Distance between nodes

  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[5]. Thus, the distance, namely the number of nodes 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

  Dialog act tags are important information for the prediction model. For example, if a node contains an "Inform" tag, the node tends to connect with nodes containing "Backchannel" and "Assess" because of the nature of discussions. On the other hand, a node with an "Elicit-inform" tag does not usually connect with an "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

  The unit labels described in Section 3.1 also have an important role in the prediction of the link between nodes. They contain three types of labels related to questions: "Open Issue," "A/B Issue," and "Yes/No Issue." If 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

  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.

4.1.2 BERT

The second model is BERT (Devlin et al., 2019). BERT is a Transformer-based machine learning model that is pre-trained by a large corpus. It can fine-tune the target tasks. BERT is known to perform well in various natural language processing

[5] 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.

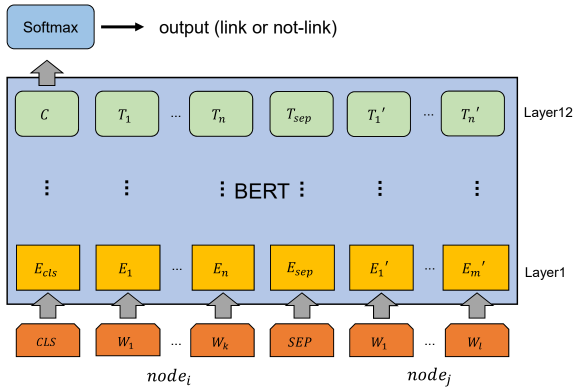

Figure 4: The link prediction model with BERT.

tasks such as text entailment, sentiment analysis, and question answering (Ke et al., 2020; Liu et al., 2020). Figure 4 shows the outline of the model. We input the class, namely "linked" or "not-linked," as the CLS token in BERT. We also input word embeddings of $node_i$ and $node_j$ with the SEP token. Finally, we compute the softmax function of the CLS token in the 12th layer for the classification.

4.2 Link selection

The link prediction models in Section 4.1 tend to over-generate links. In this section, we introduce two types of selection approaches: bottom-up and top-down.

4.2.1 Bottom-up approach

In this approach, first of all, each model (SVM and BERT) predicts the relation of each node: "link" or "not-link." Figure 5 shows an overview of the bottom-up approach. In this figure, we obtain the link prediction result; e.g., $node_1$ is linked to $node_2$, $node_3$, and $node_4$. Then, we prune links on the basis of a cost parameter in the case that a child node has two or more parent nodes; e.g., $node_5$ has two parents ($node_2$ and $node_3$). We employ the value of `decision_function` on scikit-learn for SVM. For BERT, we employ the value of the softmax function.

We select the node pair with the highest value as the final result if a child node has some parent nodes. In Figure 5, $node_5$ has two parents with values ($node_2$ with 0.8 and $node_3$ with 0.9). The bottom-up approach selects $node_3$ as the final link. We apply this bottom-up approach to the SVM model in Section 4.1.1 and the BERT model in Section 4.1.2 as the post-processing step, namely the selection process.

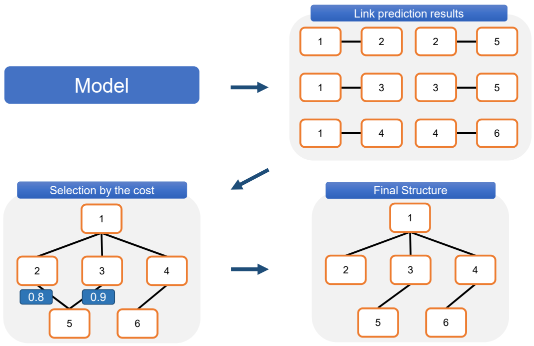

Figure 5: The bottom-up approach.

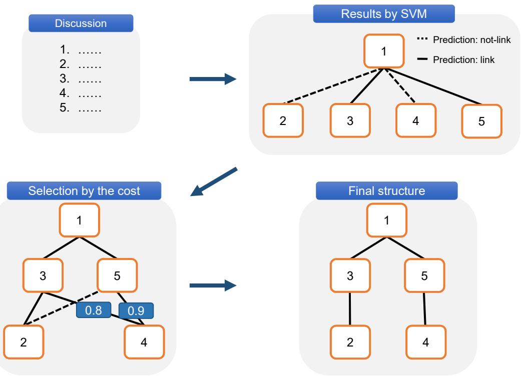

Figure 6: The top-down approach.

4.2.2 Top-down approach

Figure 6 shows an overview of the top-down approach. Firstly, in this approach, we assume that the first node in each discussion is the first parent node. Then, we predict the presence of the link between the parent node and each node in a discussion. In Figure 6, the solid line denotes that a link prediction model judged "the two nodes contain the link." On the other hand, the dashed line denotes that the model judged "no link between the two nodes." For example, the model judges that $node_3$ and $node_5$ are linked with $node_1$ (the first parent node). Next, we set new parents by using the result; i.e., $node_3$ and $node_5$ are new parents. We repeat the process for the new parents. We select the final link in the case that two or more parent nodes have one child node. In Figure 6, $node_4$ has two parents, namely $node_3$ and $node_5$. In other words, the model predicts that $node_3$ is linked to $node_4$ and $node_5$ is also linked to $node_4$. In a similar way with the bottom-up approach, we select the node pair with the highest value as the final result if a child node has some parent nodes: e.g., $node_5$ in Figure 6.

The top-down approach needs a higher computa-

	Dialog	Discussion	Linked	Not-linked
Train	84	201	3850	38530
Dev	4	13	235	1822
Test	4	12	238	1875

Table 2: Distribution of the experimental data. For the training data, we select 3850 not-linked pairs randomly to generate balanced training data.

Model	Link			Not-Link		
	P	R	F1	P	R	F1
SVM	0.38	**0.84**	0.53	**0.98**	0.84	0.90
$+Bup$	**0.68**	0.63	**0.66**	0.96	**0.97**	**0.96**
$+Tdown$	0.58	0.47	0.52	0.94	0.94	0.94

Table 3: Results of SVM.

tional cost than the bottom-up approach because it needs to handle all combinations sequentially from the beginning. BERT tends to need a huge amount of inference time. Therefore, we apply this top-down approach to only the SVM model in Section 4.1.1 due to the calculation amount.

5 Experiment

5.1 Experimental Settings

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. The other parameters were default values on scikit-learn. The setting was determined from the training data with a grid search on scikit-learn.

For the BERT model, we used the BERT-Base as the pre-trained model. The text has been lowercased. The batch size for the training was 16, and the batch size for the development and test was 8. The number of epochs was 3. We used cross-entropy as the loss function. The optimizer was Adam, and the learning rate was 0.00002.

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 2.

Table 2 said that the experimental data were imbalanced; the linked pairs were just 3850 as against 38530 not-linked pairs. Models generated from imbalanced data tend to 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 SVM and BERT, from the downsized and balanced training data.

Model	Link			Not-Link		
	P	R	F1	P	R	F1
$BERT$	0.15	**0.65**	0.24	**0.93**	0.54	0.68
$+Bup$	**0.40**	0.31	**0.35**	0.92	**0.95**	**0.93**

Table 4: Results of BERT.

5.2 Experimental Results

We compared the effectiveness of our two-step methods. Table 3 and Table 4 show the experimental result of SVMs and BERT, respectively. $Tdown$ and Bup denote the top-down and bottom-up approaches for each model, respectively. The boldface denotes the best score for each criterion, namely Precision, Recall, and F1-score, in the table. On the F1-score, the two-step methods with the bottom-up outperformed the methods without the link selection. The top-down approach on SVM improved the precision rate although the F1-score was lower than SVM without selection. These results show the effectiveness of our two-step methods, namely the introduction of the link selection approach.

5.3 Discussion

The link prediction models obtained high recall rates for the class “link” (0.84 on SVM and 0.65 on BERT) while the precision rates were low. In other words, the outputs contained many mistakes: child nodes with some parent nodes. The problem was remedied by introducing link selection models, especially the bottom-up. On the other hand, the improvement of the top-down approach was limited. The reason is that the top-down approach sequentially predicts a link of two nodes and selects the link from the top node. Although it can hold the relation between two nodes, it is not suitable to hold the relation of the whole discussion. As a result, the result was not sufficient. Moreover, in the top-down approach, the mistakes of the link prediction in the early stage lead to a negative impact on the later stage accumulatively. To obtain higher

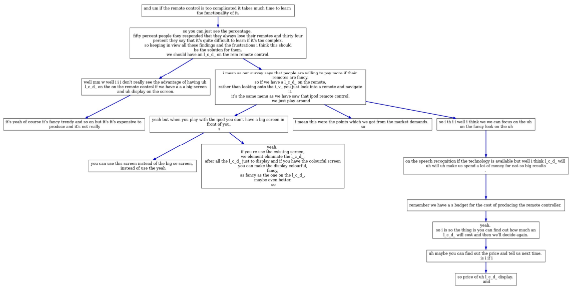
and um if the remote control is too complicated it takes much time to learn the functionality of it.
so you can just see the percentage, fifty percent people they responded that they always lose their remotes and thirty four percent they say that it's quite difficult to learn if it's too complex. so keeping in view all these findings and the frustrations i think this should be the solution for them. we should have an l_c_d_ on the rem remote control.
well mm w well i i i don't really see the advantage of having uh l_c_d_ on the on the remote control if we have a a a big screen and uh display on the screen.
i mean as our survey says that people are willing to pay more if their remotes are fancy. so if we have a l_c_d_ on the remote, rather than looking onto the t_v_ you just look into a remote and navigate it. it's the same menu as we have saw that ipod remote control. we just play around
it's yeah of course it's fancy trendy and so on but it's it's expensive to produce and it's not really
yeah but when you play with the ipod you don't have a big screen in front of you, s
i mean this were the points which we got from the market demands. so
so i th i i well i think we we can focus on the uh on the fancy look on the uh
you can use this screen instead of the big se screen, instead of use the yeah
yeah. if you re-use the existing screen, we element eliminate the l_c_d_, after all the l_c_d_ just to display and if you have the colourful screen you can make the display colourful, fancy, as fancy as the one on the l_c_d_, maybe even better. so
on the speech recognition if the technology is available but well i think l_c_d_ will uh will uh make us spend a lot of money for not so big results .
remember we have a s budget for the cost of producing the remote controller.
yeah. so i is so the thing is you can find out how much an l_c_d_ will cost and then we'll decide again.
uh maybe you can find out the price and tell us next time. is i if i
so price of uh l_c_d_ display. and

Figure 7: The correct structure.

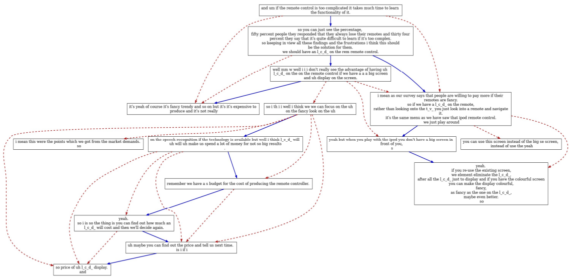
and um if the remote control is too complicated it takes much time to learn the functionality of it.
so you can just see the percentage, fifty percent people they responded that they always lose their remotes and thirty four percent they say that it's quite difficult to learn if it's too complex. so keeping in view all these findings and the frustrations i think this should be the solution for them. we should have an l_c_d_ on the rem remote control.
well mm w well i i i don't really see the advantage of having uh l_c_d_ on the on the remote control if we have a a a big screen and uh display on the screen.
it's yeah of course it's fancy trendy and so on but it's it's expensive to produce and it's not really
so i th i i well i think we we can focus on the uh on the fancy look on the uh
i mean as our survey says that people are willing to pay more if their remotes are fancy. so if we have a l_c_d_ on the remote, rather than looking onto the t_v_ you just look into a remote and navigate it. it's the same menu as we have saw that ipod remote control. we just play around
i mean this were the points which we got from the market demands. so
on the speech recognition if the technology is available but well i think l_c_d_ will uh will uh make us spend a lot of money for not so big results .
yeah but when you play with the ipod you don't have a big screen in front of you, s
you can use this screen instead of the big se screen, instead of use the yeah
remember we have a s budget for the cost of producing the remote controller.
yeah. if you re-use the existing screen, we element eliminate the l_c_d_, after all the l_c_d_ just to display and if you have the colourful screen you can make the display colourful, fancy, as fancy as the one on the l_c_d_, maybe even better. so
yeah. so i is so the thing is you can find out how much an l_c_d_ will cost and then we'll decide again.
uh maybe you can find out the price and tell us next time. is i if i
so price of uh l_c_d_ display. and

Figure 8: Output of SVM without selection.

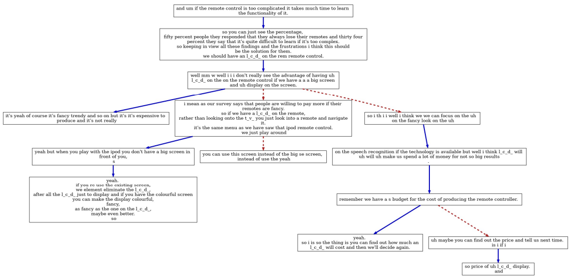
and um if the remote control is too complicated it takes much time to learn the functionality of it.
so you can just see the percentage, fifty percent people they responded that they always lose their remotes and thirty four percent they say that it's quite difficult to learn if it's too complex. so keeping in view all these findings and the frustrations i think this should be the solution for them. we should have an l_c_d_ on the rem remote control.
well mm w well i i i don't really see the advantage of having uh l_c_d_ on the on the remote control if we have a a a big screen and uh display on the screen.
it's yeah of course it's fancy trendy and so on but it's it's expensive to produce and it's not really
i mean as our survey says that people are willing to pay more if their remotes are fancy. so if we have a l_c_d_ on the remote, rather than looking onto the t_v_ you just look into a remote and navigate it. it's the same menu as we have saw that ipod remote control. we just play around
so i th i i well i think we we can focus on the uh on the fancy look on the uh
yeah but when you play with the ipod you don't have a big screen in front of you, s
you can use this screen instead of the big se screen, instead of use the yeah
on the speech recognition if the technology is available but well i think l_c_d_ will uh will uh make us spend a lot of money for not so big results .
yeah. if you re-use the existing screen, we element eliminate the l_c_d_, after all the l_c_d_ just to display and if you have the colourful screen you can make the display colourful, fancy, as fancy as the one on the l_c_d_, maybe even better. so
remember we have a s budget for the cost of producing the remote controller.
yeah. so i is so the thing is you can find out how much an l_c_d_ will cost and then we'll decide again.
uh maybe you can find out the price and tell us next time. is i if i
so price of uh l_c_d_ display. and

Figure 9: Output of SVM with the bottom-up approach (Our method).

accuracy by the top-down approach, we need additional rules and conditions in the link selection process: e.g., appearance order in the discussion.

We visualized discussion structures from the predicted results. Figure 7 shows an example of the correct discussion structure of a discussion. Figure 8 and Figure 9 show the visualized structures of SVM and SVM with the bottom-up approach, respectively. Figure 10 and Figure 11 also show the visualized structures of BERT only and BERT with the bottom-up approach, respectively. In each figure, solid blue lines denote correct links, and dashed red lines denote incorrect links. From the figure, our method dramatically improved the discussion structure prediction task. In particular, the output of the BERT model was obviously refined, namely the deletion of dashed red lines, because the precision rate of the original BERT model was extremely low. For the SVM model, the output from our method reduced mis-prediction as a whole.

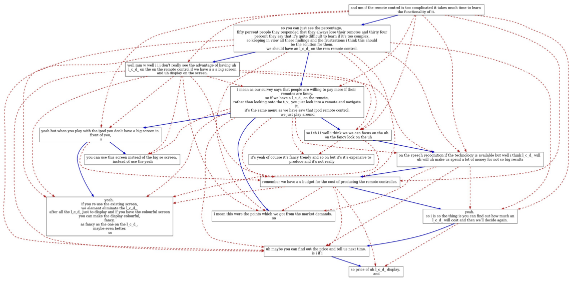
and um if the remote control is too complicated it takes much time to learn the functionality of it.
so you can just see the percentage, fifty percent people they responded that they always lose their remotes and thirty four percent they say that it's quite difficult to learn if it's too complex. so keeping in view all these findings and the frustrations i think this should be the solution for them. we should have an l_c_d_ on the rem remote control.
well mm w well i i i don't really see the advantage of having uh l_c_d_ on the on the remote control if we have a a a big screen and uh display on the screen.
i mean as our survey says that people are willing to pay more if their remotes are fancy. so if we have a l_c_d_ on the remote, rather than looking onto the t_v_ you just look into a remote and navigate it. it's the same menu as we have saw that ipod remote control. we just play around
yeah but when you play with the ipod you don't have a big screen in front of you, s
so i th i i well i think we we can focus on the uh on the fancy look on the uh
you can use this screen instead of the big se screen, instead of use the yeah
it's yeah of course it's fancy trendy and so on but it's it's expensive to produce and it's not really
on the speech recognition if the technology is available but well i think l_c_d_ will uh will uh make us spend a lot of money for not so big results .
remember we have a s budget for the cost of producing the remote controller.
yeah. if you re-use the existing screen, we element eliminate the l_c_d_, after all the l_c_d_ just to display and if you have the colourful screen you can make the display colourful, fancy, as fancy as the one on the l_c_d_, maybe even better. so
i mean this were the points which we got from the market demands. so
yeah. so i is so the thing is you can find out how much an l_c_d_ will cost and then we'll decide again.
uh maybe you can find out the price and tell us next time. is i if i
so price of uh l_c_d_ display. and

Figure 10: Output of BERT without selection.

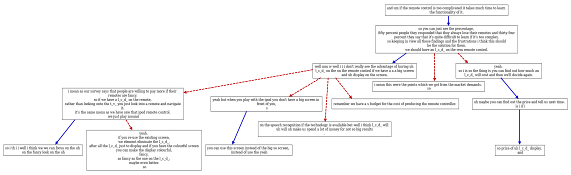
and um if the remote control is too complicated it takes much time to learn the functionality of it.
so you can just see the percentage, fifty percent people they responded that they always lose their remotes and thirty four percent they say that it's quite difficult to learn if it's too complex. so keeping in view all these findings and the frustrations i think this should be the solution for them. we should have an l_c_d_ on the rem remote control.
well mm w well i i i don't really see the advantage of having uh l_c_d_ on the on the remote control if we have a a a big screen and uh display on the screen.
yeah. so i is so the thing is you can find out how much an l_c_d_ will cost and then we'll decide again.
i mean this were the points which we got from the market demands. so
i mean as our survey says that people are willing to pay more if their remotes are fancy. so if we have a l_c_d_ on the remote, rather than looking onto the t_v_ you just look into a remote and navigate it. it's the same menu as we have saw that ipod remote control. we just play around
yeah but when you play with the ipod you don't have a big screen in front of you, s
it's yeah of course it's fancy trendy and so on but it's it's expensive to produce and it's not really
remember we have a s budget for the cost of producing the remote controller.
uh maybe you can find out the price and tell us next time. is i if i
on the speech recognition if the technology is available but well i think l_c_d_ will uh will uh make us spend a lot of money for not so big results .
so i th i i well i think we we can focus on the uh on the fancy look on the uh
yeah. if you re-use the existing screen, we element eliminate the l_c_d_, after all the l_c_d_ just to display and if you have the colourful screen you can make the display colourful, fancy, as fancy as the one on the l_c_d_, maybe even better. so
you can use this screen instead of the big se screen, instead of use the yeah
so price of uh l_c_d_ display. and

Figure 11: Output of BERT with the bottom-up approach (our method).

The parameters of this experiment were determined from the training data. Therefore, the parameters are not always the best parameters in the experiment. Moreover, we just evaluated our methods with one setting in Table 2. We need to investigate the best parameters and the effectiveness of our method through cross-validation. In addition, we used word2vec for the embeddings for SVM although we can currently obtain embeddings from BERT as a stronger embedding. The BERT embeddings might lead to the improvement of the accuracy of the SVM-based model. The replacement of word embeddings for SVMs is one future work.

In our previous work (Himeno and Shimada, 2020), link prediction models tended to over-generate links between nodes. In this experiment, the bottom-up approach was effective for both SVM and BERT. Therefore, we believe that our two-step method is versatile and effective for relation identification tasks of argument mining. However, we evaluated our method with only the AMI corpus. Applying our method to other corpora and evaluating the effectiveness of our method in relation identification tasks are important future work

6 Conclusions

In this paper, we proposed two types of discussion structure prediction methods. They were based on a two-step architecture: link prediction and link selection. For the link prediction, we evaluated two machine learning models, namely SVM and BERT. These models tended to over-generate links between nodes. To solve this problem, we introduced top-down and bottom-up approaches to the link selection task. Our methods outperformed SVM and BERT without the link selection approaches (0.53 vs. 0.66 on F1 for SVM and 0.24 vs. 0.41 on F1 for BERT). In the experiment, the bottom-up approach was better than the top-down approach. We visualized the discussion structures from the outputs. From the visualized data, we can see qualitatively that our methods dramatically improved the discussion structure prediction.

To obtain higher accuracy, the recall rate of the link prediction model is the most important factor. It was indicated by the results of SVM; the higher recall rate led to the best performance, as compared with the BERT-based model. Therefore the important future work is to improve the link prediction models by machine learning, especially the recall rate.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 20K12110.

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.

Jon Chamberlain, Udo Kruschwitz, and Orland Hoeber. 2018. Scalable visualisation of sentiment and stance. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4181–4185.

Chihchung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Edsger W Dijkstra et al. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Harold N Gabow and Robert Endre Tarjan. 1985. A linear-time algorithm for a special case of disjoint set union. *Journal of computer and system sciences*, 30(2):209–221.

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.

Taskumi Himeno and Kazutaka Shimada. 2020. Relation identification using dialogical features in multiparty conversation. In *Proceedings of the 8th International Symposium on Applied Engineering and Sciences*, pages C–O2–02.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2020. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online. Association for Computational Linguistics.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. RikiNet: Reading Wikipedia pages for natural question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6762–6771, Online.

Luca Lugini, Christopher Olshefski, Ravneet Singh, Diane Litman, and Amanda Godley. 2020. Discussion tracker: Supporting teacher learning about students' collaborative argumentation in high school classrooms. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 53–58, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

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.

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.

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.

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

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

Christian Stab and Iryna Gurevych. 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. Association for Computational Linguistics.

Vladimir Vapnik. 2000. *The Nature of Statistical Learning Theory*. Springer.