Learning Sentence Ordering for Opinion Generation of Debate

Toshihiko Yanase **Toshinori Miyoshi** **Kohsuke Yanai** **Misa Sato**
Research & Development Group, Hitachi, Ltd.
{toshihiko.yanase.gm, toshinori.miyoshi.pd, kohsuke.yanai.cs, misa.sato.mw}
@hitachi.com

Makoto Iwayama **Yoshiki Niwa**
Research & Development Group, Hitachi, Ltd.
{makoto.iwayama.nw, yoshiki.niwa.tx}
@hitachi.com

Paul Reisert **Kentaro Inui**
Tohoku University
{preisert, inui}@ecei.tohoku.ac.jp

Abstract

We propose a sentence ordering method to help compose persuasive opinions for debating. In debate texts, support of an opinion such as evidence and reason typically follows the main claim. We focused on this claim-support structure to order sentences, and developed a two-step method. First, we select from among candidate sentences a first sentence that is likely to be a claim. Second, we order the remaining sentences by using a ranking-based method. We tested the effectiveness of the proposed method by comparing it with a general-purpose method of sentence ordering and found through experiment that it improves the accuracy of first sentence selection by about 19 percentage points and had a superior performance over all metrics. We also applied the proposed method to a constructive speech generation task.

1 Introduction

There are increasing demands for information structuring technologies to support decision making using a large amount of data. Argumentation in debating which composes texts in a persuasive manner is a research target suitable for such information structuring. In this paper, we discuss sentence ordering for constructive speech generation of debate.

The following is an example of constructive speech excerpts that provide affirmative opinions on the banning of gambling[1].

> Motion: This House should ban gambling.
> (1) Poor people are more likely to gamble, in the hope of getting rich.
> (2) In 1999, the National Gambling Impact Commission in the United States found that 80 percent of gambling revenue came from lower-income households.

We can observe a typical structure of constructive speech in this example. The first sentence describes a claim that is the main statement of the opinion and the second sentence supports the main statement. In this paper, we focus on this claim-support structure to order sentences.

Regarding the structures of arguments, we can find research on the modeling of arguments (Freeley and Steinberg, 2008) and on recognition such as claim detection (Aharoni et al., 2014). To the best of our knowledge, there is no research that examines the claim-support structure of debate texts for the sentence ordering problem. Most of the previous works on sentence ordering (Barzilay et al., 2002; Lapata, 2003; Bollegala et al., 2006; Tan et al., 2013) focus on the sentence order of news articles and do not consider the structures of arguments. These methods mingle claim and supportive sentences together, which decreases the persuasiveness of generated opinions.

In this paper, we propose a sentence ordering method in which a motion and a set of sentences are given as input. Ordering all paragraphs of debate texts at once is a quite difficult task, so we have

[1] This example is excerpted from Debatabase (http://idebate.org/debatabase). Copyright 2005 International Debate Education Association. All Rights Reserved.

Proceedings of the 2nd Workshop on Argumentation Mining, pages 94–103,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

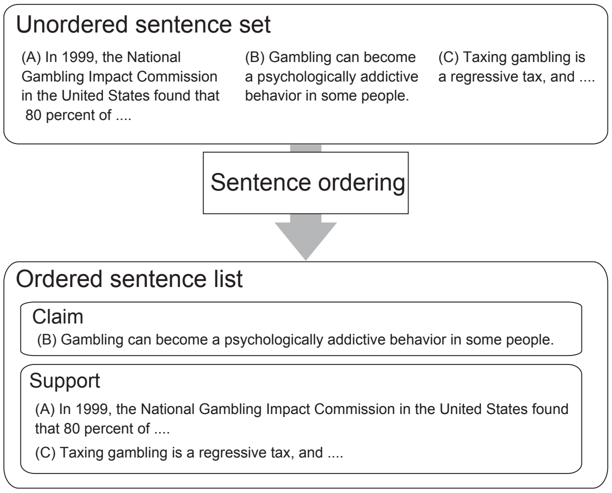
Unordered sentence set
(A) In 1999, the National Gambling Impact Commission in the United States found that 80 percent of ....
(B) Gambling can become a psychologically addictive behavior in some people.
(C) Taxing gambling is a regressive tax, and ....
Sentence ordering
Ordered sentence list
Claim
(B) Gambling can become a psychologically addictive behavior in some people.
Support
(A) In 1999, the National Gambling Impact Commission in the United States found that 80 percent of ....
(C) Taxing gambling is a regressive tax, and ....

Figure 1: Target sentence ordering problem.

simplified by assuming that all input sentences stand for a single viewpoint regarding the motion.

We use this claim-support structure as a cue of sentence ordering. We employ two-step ordering based on machine learning, as shown in Fig. 1. First, we select a first sentence that corresponds to a claim, and second, we order the supportive sentences of the claims in terms of consistency. For each step, we design machine learning features to capture the characteristics of sentences in terms of the claim-support structure. The dataset for training and testing is made up of content from an online debate site.

The remainder of this paper is structured as follows. The next section describes related works dealing with sentence ordering. In the third section, we examine the characteristics of debate texts. Next, we describe our proposed method, explain the experiments we performed to evaluate the performance, and discuss the results. After that, we describe our application of the proposed sentence ordering to automated constructive speech generation. We conclude the paper with a summary and a brief mention of future work.

2 Related Works

Previous research on sentence ordering has been conducted as a part of multi-document summarization. There are four major feature types to order sentences: publication dates of source documents, topical similarity, transitional association cues, and rhetorical cues.

Arranging sentences by order of publication dates of source documents is known as the chronological ordering (Barzilay et al., 2002). It is effective for news article summarization because descriptions of a certain event tend to follow the order of publication. It is, however, not suitable for opinion generation because such generation requires statements and evidence rather than the simple summarization of an event.

Topical similarity is based on an assumption that neighboring sentences have a higher similarity than non-neighboring ones. For example, bag-of-words-based cosine similarities of sentence pairs are used in (Bollegala et al., 2006; Tan et al., 2013). Another method, the Lexical Chain, models the semantic distances of word pairs on the basis of synonym dictionaries such as WordNet (Barzilay and Elhadad, 1997; Chen et al., 2005). The effectiveness of this feature depends highly on the method used to calculate similarity.

Transitional association is used to measure the likelihood of two consecutive sentences. Lapata proposed a sentence ordering method based on a probabilistic model (Lapata, 2003). This method uses conditional probability to represent transitional probability from the previous sentence to the target sentence.

Dias et al. used rhetorical structures to order sentences (de S. Dias et al., 2014). The rhetorical structure theory (RST) (Mann and Thompson, 1988) explains the textual organization such as background and causal effect that can be useful to determine the sentence order. For example, causes are likely to precede results. However, it is important to restrict the types of rhetorical relation because original RST defines many relations and a large amount of data is required for accurate estimation.

There has been research on integrating different types of features. Bollegara et al. proposed machine learning-based integration of different kinds of features (Bollegala et al., 2006) by using a binary classifier to determine if the order of a given sentence pair is acceptable or not. Tan et al. formulated sentence ordering as a ranking problem of sentences (Tan et al., 2013). Their experimental results showed that the ranking-based method outperformed classification-based methods.

Viewpoint	Debate	News
Word overlap in neighbors	3.14	4.30
Word overlap in non-neighbors	3.09	4.22
Occurrence of named entity	0.372	0.832

Table 1: Characteristics of debate texts and news articles.

3 Characteristics of Debate Texts

Topical similarity can be measured by the word overlap between two sentences. This metric assumes that the closer a sentence pair is, the more word overlap exists. In order to examine this assumption, we compared characteristics between debate texts and news articles, as shown in Table 1. In the Debate column, we show the statistics of constructive speech of Debatabase, an online debate site. Each constructive speech item in the debate dataset has 7.2 sentences on average. Details of the debate dataset are described in the experiment section. In the News column, we show the statistics of a subset of Annotated English Gigaword (Napoles et al., 2012). We randomly selected 80,000 articles and extracted seven leading sentences per article.

Overall, we found less word overlap in debate texts than in news articles in both neighbor pairs and non-neighbor pairs. This is mainly because debaters usually try to add as much information as possible. We assume from this result that conventional topical similarity is less effective for debate texts and have therefore focused on the claim-support structure of debate texts.

We also examined the occurrence of named entity (NE) in each sentence. We can observe that most of the sentences in news articles contain NEs while much fewer sentences in debate texts have NEs. This suggests that debate texts deal more with general opinions and related examples while news articles describe specific events.

4 Proposed Method

4.1 Two-Step Ordering

In this study, we focused on a simple but common style of constructive speech. We assumed that a constructive speech item has a claim and one or more supporting sentences. The flow of the proposed ordering method is shown in Fig. 2. The system receives a motion and a set of sentences as input and then it outputs ordered sentences. First, syntactic parsing is applied to the input texts, and then features for the machine learning models are then extracted from the results. Second, we select the first sentence, which is likely to be the claim sentence, from the candidate sentences. This problem is formulated as a binary-classification problem, where first sentences of constructive speech items are positive and all others are negative. Third, we order the remaining sentences on the basis of connectivity of pairs of sentences. This problem is formulated as a ranking problem, similarly to (Tan et al., 2013).

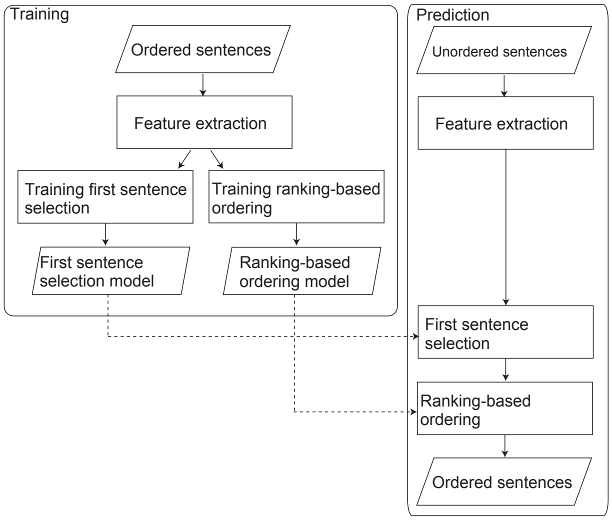

Figure 2: Flowchart of two-step ordering.

4.2 Feature Extraction

We obtained the part of speech, lemma, syntactic parse tree, and NEs of each input sentence by using the Stanford Core NLP (Manning et al., 2014).

The following features, which are commonly used in sentence ordering methods to measure local coherence (Bollegala et al., 2006; Tan et al., 2013; Lapata, 2003), are then extracted.

Sentence similarity: Cosine similarity between sentence u and v. We simply counted the frequency of each word to measure cosine similarity. In addition to that, we also measured the cosine similarity between latter half of u (denoted as $\mathrm{latter}(u)$) and former half of v (denoted as $\mathrm{former}(v)$). The sentences are separated by the most centered comma (if exists) or word (if no comma exists).

Overlap: Commonly shared words of u and v. Let $\mathrm{overlap}_j(u, v)$ be the number of commonly shared words of u and v, for $j = 1, 2, 3$ representing lemmatized noun, verb and adjective or adverb, respectively. We calculated $\mathrm{overlap}_j(u, v)/\min(|u|, |v|)$ and $\mathrm{overlap}_j(\mathrm{latter}(u), \mathrm{former}(v))/\mathrm{overlap}_j(u, v)$, where $|u|$ is the number of words of sentence u. The value will be set to 0 if the denominator is 0.

Expanded sentence similarity: Cosine similarity between candidate sentences expanded with synonyms. We used WordNet (Miller, 1995) to expand the nouns and verbs into synonyms.

Word transitional probability: Calculate conditional probability $P(w_v|w_u)$, where w_u, w_v denote the words in sentences u, v, respectively. In the case of the first sentence, we used $P(w_u)$. A probabilistic model based on Lapata's method (Lapata, 2003) was created.

The following features are used to capture the characteristics of claim sentences.

Motion similarity: Cosine similarity between the motion and the target sentence. This feature examines the existence of the motion keywords.

Expanded motion similarity: Cosine similarity of the target sentence to the motion expanded with synonyms.

Value relevance: Ratio of value expressions. In this study, we defined human values as the topics obviously considered to be positive or negative and highly relevant to people's values and then created a dictionary of value expressions. For example, health, education, and the environment are considered positive for people's values while crime, pollution, and high costs are considered negative.

Sentiment: Ratio of positive or negative words. The dictionary of sentimental words is from (Hu and Liu, 2004). This feature is used to examine whether the stance of the target sentence is positive, negative, or neutral.

Type	1st step	2nd step
Sentence similarity		✓
Expanded sentence similarity		✓
Overlap		✓
Word transitional probability	✓	✓
Motion similarity	✓	✓
Expanded motion similarity	✓	✓
Value relevance	✓	✓
Sentiment	✓	✓
Concreteness	✓	✓
Estimated first sentence similarity		✓

Table 2: Features used in each step.

Concreteness features are used to measure the relevance of support.

Concreteness features: The ratio of tokens that are a part of capital words, numerical expression, NE, organization, person, location, or temporal expression. These features are used to capture characteristics of the supporting sentences.

We use the estimated results of the first step as a feature of the second step.

Estimated first sentence similarity: Cosine similarity between the target sentence and the estimated first sentence.

4.3 First Step: First Sentence Selection

In the first step, we choose a first sentence from input sentences. This task can be formulated as a binary classification problem. We employ a machine learning approach to solve this problem.

In the training phase, we extract N feature vectors from N sentences in a document, and train a binary classification function f_{first} defined by

$$f_{\text{first}}(s_i) = \begin{cases} +1 & (i = 0) \\ -1 & (i \neq 0) \end{cases}, \qquad (1)$$

where s_i denotes the feature vector corresponding to the i-th sentence. The function f_{first} returns $+1$ if s_i is the first sentence.

In the prediction phase, we applied f_{first} to all sentences and determined the first sentence that has the maximum posterior probability of $f_{\text{first}}(s_i) = +1$.

We used Classias[2] (Okazaki, 2009), an implementation of logistic regression, as a binary classifier.

4.4 Second Step: Ranking-Based Ordering

In the second step, we assume that the first sentence has already been determined. The number of sentences in this step is $N_{\text{second}} = N - 1$. We use a ranking-based framework proposed by Tan et al. (2013) to order sentences.

In the training phase, we generate $N_{\text{second}}(N_{\text{second}} - 1)$ pairs of sentences from N_{second} sentences in a document and train an association strength function f_{pair} defined by

$$f_{\text{pair}}(s_i, s_j) = \begin{cases} N_{\text{second}} - (j - i) & (j > i) \\ 0 & (j \leq i) \end{cases}. \quad (2)$$

For forward direction pairs, the rank values are set to $N - (j - i)$. This means that the shorter the distance between the pair is, the larger the rank value is. For the backward direction pairs, the rank values are set to 0.

In the prediction phase, the total ranking value of a sentence permutation ρ is defined by

$$f_{\text{rank}}(\rho) = \sum_{u,v;\rho(u)>\rho(v)} f_{\text{pair}}(u, v), \quad (3)$$

where $\rho(u) > \rho(v)$ denotes that sentence u precedes sentence v in ρ. A learning to rank algorithm based on Support Vector Machine (Joachims, 2002) is used as a machine learning model. We used svm^{rank} [3] to implement the training and the prediction of f_{pair}.

We used the sentence similarity, the expanded sentence similarity, the overlap, and the transitional probability in addition to the same features as the first step classification. These additional features are defined by a sentence pair (u, v). We applied the feature normalization proposed by Tan et al. (2013) to each additional feature. The normalization functions are defined as

$$V_{i,1} = f_i(u, v), \quad (4)$$

$$V_{i,2} = \begin{cases} 1/2, & \text{if } f_i(u,v) + f_i(v,u) = 0 \\ \frac{f_i(u,v)}{f_i(u,v)+f_i(v,u)}, & \text{otherwise} \end{cases} \quad (5)$$

$$V_{i,3} = \begin{cases} 1/|S|, & \text{if } \sum_{y \in S \setminus \{u\}} f_i(u,y) = 0 \\ \frac{f_i(u,v)}{\sum_{y \in S \setminus \{u\}} f_i(u,y)}, & \text{otherwise} \end{cases} \quad (6)$$

$$V_{i,4} = \begin{cases} 1/|S|, & \text{if } \sum_{x \in S \setminus \{v\}} f_i(x,v) = 0 \\ \frac{f_i(u,v)}{\sum_{x \in S \setminus \{v\}} f_i(x,v)}, & \text{otherwise} \end{cases} \quad (7)$$

where f_i is the i-th feature function, S is a set of candidate sentences, and $|S|$ is the number of sentences in S. Equation (4) is an original value of the i-th feature function. Equation (5) examines the priority of (u, v) to its inversion (v, u). Equation (6) measures the priority of (u, v) to the sentence pairs that have u as a first element. Equation (7) the priority of (u, v) to the sentence pairs that have v as a second element, similarly to Equation (6).

5 Experiments

5.1 Reconstructing Shuffled Sentences

We evaluated the proposed method by reconstructing the original order from randomly shuffled texts. We compared the proposed method with the Random method, which is a base line method that randomly selects a sentence, and the Ranking method, which is a form of Tan et al.'s method (Tan et al., 2013) that arranges sentences using the same procedure as the second step of the proposed method excluding estimated first sentence similarity feature.

Dataset

We created a dataset of constructive speech items from Debatabase to train and evaluate the proposed method. The speech item of this dataset is a whole turn of affirmative/negative constructive speech which consists of several ordered sentences. Details of the dataset were shown in Table 3. The dataset has 501 motions related to 14 themes (e.g., politics, education) and contains a total of 3,754 constructive speech items. The average sentence length per item is 7.2. Each constructive speech item has a short title sentence from which we extract the value (e.g., "health", "crime") of the item.

[2] http://www.chokkan.org/software/classias/

[3] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Affirmative	no. of constructive speech items	1,939
	no. of sentences	14,021
Negative	no. of constructive speech items	1,815
	no. of sentences	13,041

Table 3: Details of constructive speech dataset created from Debatabase.

Metrics

The overall performance of ordering sentences is evaluated by Kendall's τ, Spearman Rank Correlation, and Average Continuity.

Kendall's τ is defined by

$$\tau_k = 1 - \frac{2n_{\mathrm{inv}}}{N(N-1)/2}, \tag{8}$$

where N is the number of sentences and n_{inv} is the number of inversions of sentence pairs. The metric ranges from -1 (inversed order) to 1 (identical order). Kendall's τ measures the efforts of human readers to correct wrong sentence orders.

Spearman Rank Correlation is defined by

$$\tau_s = 1 - \frac{6}{N(N+1)(N-1)} \sum_{i=1}^{N} d(i)^2, \tag{9}$$

where $d(i)$ is the difference between the correct rank and the answered rank at the i-th sentence. Spearman Rank Correlation takes the distance of wrong answers directly into account.

Average Continuity is based on the number of matched n-grams, and is defined using P_n. P_n is defined by

$$P_n = \frac{m}{N-n+1}, \tag{10}$$

where m is the number of matched n-grams. P_n measures the ratio of correct n-grams in a sequence. Average Continuity is then defined by

$$\tau_a = \exp\left(\sum_{n=2}^{k} \log(P_n + \alpha)\right), \tag{11}$$

where k is the maximum n of n-grams, and α is a small positive value to prevent divergence of score. In this experiment, we used $k = 4, \alpha = 0.01$ in accordance with (Bollegala et al., 2006).

Method	Mean accuracy [%]	Std.
Random	17.9	0.81
Ranking	23.3	0.61
Proposed	42.6	1.58

Table 4: Results of the first sentence estimation.

Results

We applied 5-fold cross validation to each ordering method. The machine learning models were trained by 3,003 constructive speech items and then evaluated using 751 items.

The results of first sentence estimation are shown in Table 4. The accuracy of the proposed method is higher than that of Ranking, which represents the sentence ranking technique without the first sentence selection, by 19.3 percentage points. Although the proposed method showed the best accuracy, we observed that $f_{\mathrm{first}}(s_0)$ tended to be -1 rather than 1. This is mainly because the two classes were unbalanced. The number of negative examples in the training data was 6.2 times larger than that of positive ones. We need to address the unbalanced data problem for further improvement (Chawla et al., 2004).

The results of overall sentence ordering are shown in Table 5. We carried out a one-way analysis of variance (ANOVA) to examine the effects of different algorithms for sentence ordering. The ANOVA revealed reliable effects with all metrics ($p < 0.01$). We performed a Tukey Honest Significant Differences (HSD) test to compare differences among these algorithms. In terms of Kendall's τ and Spearman Rank Correlation, the Tukey HSD test revealed that the proposed method was significantly better than the rests ($p < 0.01$). In terms of Average Continuity, it was also significantly better than the Random method, whereas it is not significantly different from the Ranking method. These results show that the proposed two-step ordering is also effective for overall sentence ordering. However, the small difference of Average Continuity indicates that the ordering improvement is only regional.

5.2 Subjective Evaluation

In addition to our evaluation of the reconstruction metrics, we also conducted a subjective evaluation

Method	Kendall's τ	Spearman	Average Continuity
Random	-6.92×10^{-4}	-1.91×10^{-3}	5.92×10^{-2}
Ranking	6.22×10^{-2}	7.89×10^{-2}	7.13×10^{-2}
Proposed	1.17×10^{-1}	1.44×10^{-1}	8.36×10^{-2}

Table 5: Results of overall sentence ordering.

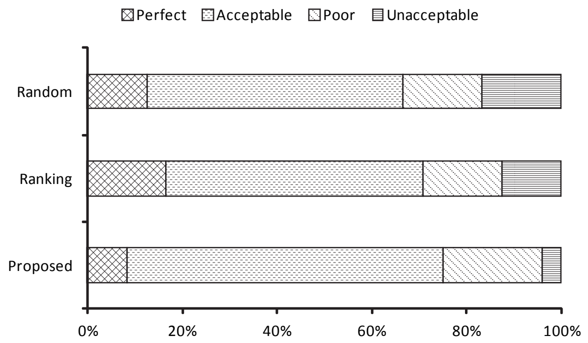

Figure 3: Results of subjective evaluation.

Position	Claim [%]	Support [%]
1	*62.5*	3.93
2	8.93	19.7
3	7.14	20.2
4	5.36	17.4
5+	16.1	*38.7*

Table 6: Sentence type annotation in constructive speech.

with a human judge. In this evaluation, we selected target documents that were ordered uniquely by people as follows. First, the judge ordered shuffled sentences and then, we selected the correctly ordered documents as targets. The number of target documents is 24.

Each ordering was awarded one of four grades: Perfect, Acceptable, Poor or Unacceptable. The criteria of these grades are the same as those of (Bollegala et al., 2006). A perfect text cannot be improved by re-ordering. An acceptable text makes sense and does not require revision although there is some room for improvement in terms of readability. A poor text loses the thread of the story in some places and requires amendment to bring it up to an acceptable level. An unacceptable text leaves much to be improved and requires overall restructuring rather than partial revision.

The results of our subjective evaluation are shown in Figure 3. We have observed that about 70 % of randomly ordered sentences are perfect or acceptable. This is mainly because the target documents contain only 3.87 sentences on average, and those short documents are comprehensive even if they are randomly shuffled.

There are four documents containing more than six sentences in the targets. The number of unacceptably ordered documents of the Random method, the Ranking method, and the proposed method are 4, 3, and 1, respectively. We observed that the proposed method selected the claim sentences successfully and then arranged sentences related to the claim sentences. These are the expected results of the first sentence classification and the estimated first sentence similarity in the second step. These results show that the selection of the first sentence plays an important role to make opinions comprehensive.

On the other hand, we did not observe the improvement of the number of the perfectly selected documents. We found misclassification of final sentences as first sentences in the results of the proposed method. Such final sentences described conclusions similar to the claim sentences. We need to extend the structure of constructive speech to handle conclusions correctly.

6 Discussion

6.1 Structures of Constructive Speech

We confirmed our assumption that claims are more likely to be described in the first sentence than others by manually examining constructive speech items. We selected seven motions from the top 100 debates in the Debatabase. These selected motions contain a total of 56 constructive speech items. A human annotator assigned claim tags and support tags for the sentences. The results are shown in Table 6.

Here, we can see that about two-thirds of claim

#	Text
1	The contributions of government funding have been shown to be capable of sustaining the costs of a museum, preventing those costs being passed on to the public in the form of admissions charges.
2	The examples of the British Labour government funding national museums has been noted above.
3	The National Museum of the American Indian in Washington was set up partially with government funding and partially with private funds, ensuring it has remained free since its opening in 2004 (Democracy Now , 2004).
4	In 2011 , China also announced that from 2012 all of its national museums would become publicly-funded and cease charging admissions fees (Zhu & Guo , 2011).

Table 7: A typical example ordered correctly by the proposed method. The motion is "This House would make all museums free of charge." The motion and sentences are from Debatabase.

sentences appeared at the beginning of constructive speech items, and that more than 90 % of supportive sentences appeared from the second sentences or later. This means that claims are followed by evidence in more than half of all constructive speech items.

6.2 Case Analysis

A typical example ordered correctly by the proposed method is shown in Table 7. This constructive speech item agrees with free admissions at museums. It has a clear claim-support structure. It first, makes a claim related to the contributions of government funding and then gives three examples. The first sentence has no NEs while the second and later sentences have NEs to give details about the actual museums and countries. Neighbor sentences were connected with common words such as "museum," "charge," and "government funding."

7 Application to Automated Constructive Speech Generation

We applied the proposed sentence ordering to the automated constructive speech generation.

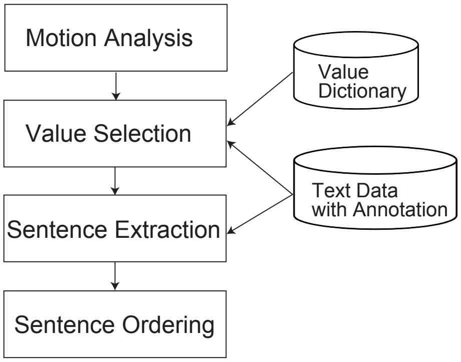

Figure 4: Flow of automated constructive speech generation.

System Description

The flowchart of constructive speech generation is shown in Fig. 4. Here, we give a brief overview of the system. The system is based on sentence extraction and sentence ordering, which we explain with the example motion "This House should ban smoking in public spaces." First, a motion analysis component extracts keywords such as "smoking" and "public spaces" from the motion. Second, a value selection component searches for related sentences with the motion keywords and human value information. More specifically, it generates pairs of motion keywords and values (such as (smoking, health), (smoking, education), and (smoking, crime)) and uses them as search queries. Then, it selects the values of constructive speech in accordance with the number of related sentences to values. In the third step, a sentence extraction component examines the relevancy of each sentence with textual annotation such as promote/suppress relationship and positive/negative relationship. Finally, a sentence ordering component arranges the extracted sentences for each value.

Ordering Results

The system outputs three paragraphs per motion. Each paragraph is composed of seven sentences. Currently, its performance is limited, as 49 out of the 150 generated paragraphs are understandable. To focus on the effect of sentence ordering, we manually extracted relevant sentences from generated constructive speech and then applied the proposed ordering method to them.

#	Text
1	Smoking is a serious public health problem that causes many diseases such as heart diseases, lung diseases, eye problems, as well as risks for women and babies.
2	Brendan McCormick, a spokesman for cigarette-maker Philip Morris USA, said, "We agree with the medical and scientific conclusions that cigarette smoking causes serious diseases in smokers, and that there is no such thing as a safe cigarette."
3	The study, released by the Rio de Janeiro State University and the Cancer Institute, showed that passive smoking could cause serious diseases, such as lung cancer, cerebral hemorrhage, angina pectoris, myocardial infection and coronary thrombosis.

Table 8: A result of sentence ordering in automated constructive speech generation. The motion is "This House would further restrict smoking." The motion is from Debatabase, and sentences are from Annotated English Gigaword.

The results are shown in Table 8[4]. We can observe that the first sentence mentions the health problem of smoking while the second and third sentences show support for the problem, i.e., the names of authorities such as spokesmen and institutes. The proposed ordering method successfully ordered the types of opinions that have a clear claim-support structure.

8 Conclusion

In this paper, we discussed sentence ordering for debate texts. We proposed a sentence ordering method that employs a two-step approach based on the claim-support structure. We then constructed a dataset from an on-line debate site to train and evaluate the ordering method. The evaluation results of reconstruction from shuffled constructive speech showed that our proposed method outperformed a general-purpose ordering method. The subjective evaluation showed that our proposed method is suitable for constructive speech containing explicit claim sentences and supporting examples.

In this study, we focused on a very simple structure, i.e., claims and support. We will extend this structure to handle different types of arguments in the future. More specifically, we plan to take conclusion sentences into account as a component of the structure.

[4] These sentences are extracted from Annotated English Gigaword. Portions ©1994-2010 Agence France Presse, ©1994-2010 The Associated Press, ©1997-2010 Central News Agency (Taiwan), ©1994-1998, 2003-2009 Los Angeles Times-Washington Post News Service, Inc., ©1994-2010 New York Times, ©2010 The Washington Post News Service with Bloomberg News, ©1995-2010 Xinhua News Agency, ©2012 Matthew R. Gormley, ©2003, 2005, 2007, 2009, 2011, 2012 Trustees of the University of Pennsylvania

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68. Association for Computational Linguistics.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1):35–55, August.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2006. A bottom-up approach to sentence ordering for multi-document summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 385–392, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, June.

Yan-Min Chen, Xiao-Long Wang, and Bing-Quan Liu. 2005. Multi-document summarization based on lexical chains. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics 2005*, volume 3, pages 1937–1942 Vol. 3, Aug.

Márcio de S. Dias, Valéria D. Feltrim, and Thiago Alexandre Salgueiro Pardo. 2014. Using rhetorical structure theory and entity grids to automatically evaluate local coherence in texts. *Proceedings of the 11th International Conference, PROPOR 2014*, pages 232–243.

Austin J. Freeley and David L. Steinberg. 2008. *Argumentation and Debate*. WADSWORTH CENGAGE Learning.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA. ACM.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 545–552, Stroudsburg, PA, USA. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

George A. Miller. 1995. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated English Gigaword ldc2012t21. AKBC-WEKEX '12, pages 95–100. Association for Computational Linguistics.

Naoaki Okazaki. 2009. Classias: a collection of machine-learning algorithms for classification.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2013. Learning to order natural language texts. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 87–91.