KitAi-PI: Summarization System for NTCIR-14 QA Lab-PoliInfo

Satoshi Hiai, Yuka Otani, Takashi Yamamura and Kazutaka Shimada

Department of Artificial Intelligence, Kyushu Institute of Technology
{s_hiai, y_otani, t_yamamura, shimada}@pluto.ai.kyutech.ac.jp

Abstract. This paper describes a summarization system for NTCIR-14 QA Lab-PoliInfo. For the summarization task, participants of the task need to generate a summary corresponding to an assemblyperson's speech in assembly minutes within the limit length. Our method extracts important sentences to summarize an assemblyperson's speech in the minutes. Our method applies a machine learning model to predict the important sentences. However, the given assembly minutes' data do not contain information about the importance of the sentences. Therefore, we construct training data for the importance prediction model using a word similarity between sentences in a speech and those in the summary. On the formal run, some scores by our method were the best in all the submitted runs of all participants. The result shows the effectiveness of our method.

Team Name. KitAi

Subtasks. Summarization task (Japanese)

Keywords: Extractive summarization · Sentence extraction · Automatic dataset construction · Machine learning

1 INTRODUCTION

This paper describes a summarization system for NTCIR-14 QA Lab-PoliInfo (summarization subtask) [2]. For the summarization task, an assemblyperson's speech and a limit length of the summary are given. Participants of the task need to generate a summary corresponding to the speech within the limit length.

Summarization methods are mainly classified into two categories: extractive and abstractive. Abstractive summarization methods can generate words and phrases not contained in the source text with pre-trained vocabulary. On the other hand, extractive summarization methods can generate grammatically well-formed summaries because the methods extract a set of sentences in the source text. Assembly minutes are primarily the evidential record of the assembly activities. Therefore, the preciseness of the summaries is more important than the readability of those. Therefore, we utilize an extractive summarization method for the task.

Our method, KitAi[1]-PI, extracts important sentences to summarize a speech. Our method applies a machine learning approach to predict the importance of sentences in documents. We require labeled data for learning the importance prediction model. However, the assembly minutes' data do not contain the importance labels for sentences. Therefore, we need to assign the importance scores to the sentences in the assembly minutes. For the assignment, we focus that the words in the summaries are used in the assembly minutes. We calculate the importance scores using a word similarity. We use the data with the importance scores to train the importance prediction model. The model predicts the importance of each sentence in the assembly minutes. We extract sentences on the basis of the importance score. In addition, we apply the sentence compression process. We extract compressed sentences with high importance to generate a meaningful summary under length constraints.

2 SYSTEM DESCRIPTION

Figure 1 shows the overview of the proposed method. We construct a sentence importance prediction model. In Section 2.1, we explain training data construction (**Step 1** in Figure 1) for the model. In Section 2.2, we explain features of the model. In the next step (**Step 2** in Figure 1), we apply a sentence compression process to generate a meaningful summary under length constraints. In the sentence extraction step (**Step 3** in Figure 1), we extract sentences on the basis of the predicted importance scores. We explain the step in Section 2.4.

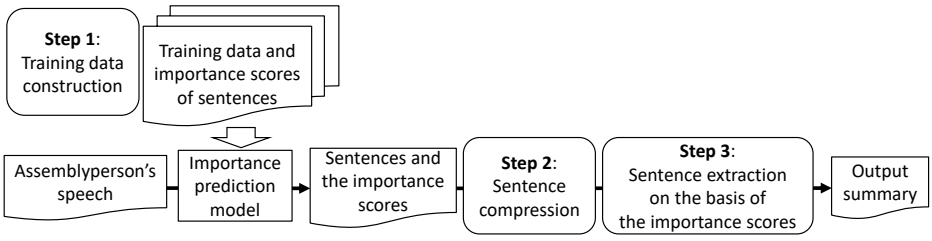

Fig. 1. Overview of our methods.

2.1 Training Data Construction

We explain the training data construction step (**Step1** in Figure 1). Figure 2 shows an example of this step. We assign importance scores to each sentence in the assembly minutes. For the assignment, we use word similarity measures between the sentences in the speech and the sentences in the summary. The

[1] Short of Kyushu Institute of Technology (Department of Artificial Intelligence). The English meaning is "expectation."

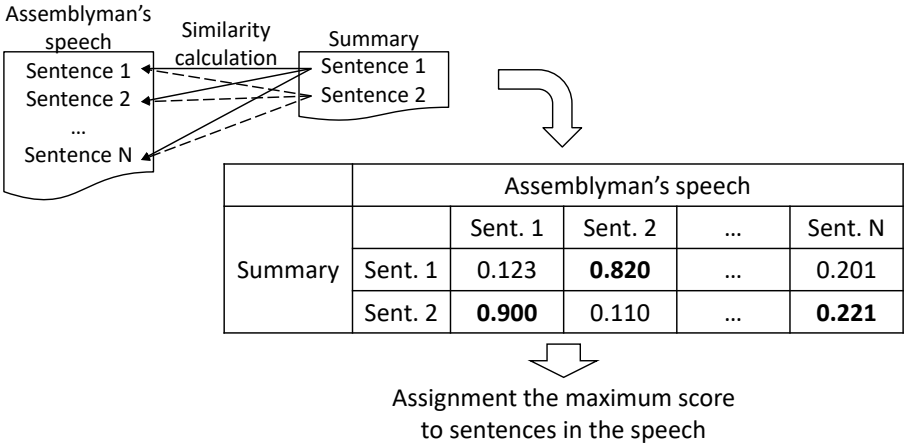

Fig. 2. Example of the training data construction step.

similarity scores corresponding to each sentence in the speech are calculated for all sentences in the summary. We assign the maximum similarity score to each sentence in the speech as the importance score.

We use several kinds of similarity measures. We need to determine the most suitable measure for training data construction. We evaluate the similarity measures on development data. We explain the evaluation in Section 3.1. The similarity measures are as follows.

- Cosine similarity between bag-of-words representations of two sentences: We use MeCab [3] with IPAdic to tokenize sentences.
- Levenshtein edit distance: The minimum number of character insertions, deletions, and substitutions that must be made to transform a sentence in a speech into a sentence in the summary. We divide the edit distance by the number of characters in the sentence to normalize the distance to the range of $[0, 1]$. Since this measure is a distance, we adopt $1-$(the normalized edit distance) as the similarity measure.
- Rouge-1 similarity score [4]: The score of the word unigram overlap between two sentences.
- Cosine similarity between sentence embeddings of two sentences: We adopt two methods to generate sentence embeddings. One is the weighted average of word embeddings generated with word2vec [5]. The other is the sentence embedding generated by doc2vec [6]. We used the given assembly minutes to train the models of word2vec and doc2vec.
- Average of all the similarity measures: The average score of the above measures. We expect that the above measures can complement each other.

2.2 Features

We construct the sentence importance prediction model using training data constructed with word similarity measures explained in the previous section. The features of the model are as follows.

- Bag-of-words: We use MeCab tokenizer [3] with IPAdic.
- Position: The features of sentence positions were used in some summarization studies [7, 1]. We use a sentence position normalized by the number of sentences in the speech.
- Speaker: The speaker of the speech is given. There are two kinds of speakers: questioners and respondents. Expressions in the summaries of questioners differ from those of respondents. For example, summaries of questioners contain expressions such as "働きかけを要請する (We request an action)." Those of respondents contain expressions such as "全力を尽くす. (We will make this work the best way we can.)" Therefore, we use a binary feature: whether the speaker is a questioner or a respondent. The questioners are given in the form of "name (a political party name)" and the respondents are given in the form of "an official position" in the speaker information.

2.3 Sentence Compression

We remove words in sentences in a speech to compress the sentences. We extract important sentences in the next step. We do not consider the context between extracted sentences. Therefore, we remove conjunctions between sentences. In addition, summaries tend to contain a sentence ending with a verbal noun. Therefore, first, we remove words before the first content word in the sentence in this step. Next, we remove words after the last verbal noun in the sentence. For example, from the sentence "このため、関係機関と連携し、狭隘道路における消火栓等の整備を促進してまいります。", we remove "このため、" and "してまいります。." If a sentence does not contain a verbal noun, we remove words after the last content word in the sentence. For example, from the sentence "事務用の場所を貸し出しております。", we remove "ております。"

2.4 Sentence Extraction

We extract compressed sentences in order of predicted importance scores within the limit length (**Step 3** in Figure 1). When the most important sentence exceeds the limit length, we use the latter words of the sentence within the limit length as the output. For example, when the compressed sentence "今後、不燃化特区の制度構築に当たりましては、都の特別の支援策として事業執行体制の確保などについても検討を進め、区と連携して木密地域の不燃化を推進" (72 characters) is the most important and the limit length is 50 characters, we use the letter words "都の特別の支援策として事業執行体制の確保などについても検討を進め、区と連携して木密地域の不燃化を推進" (50 characters).

3 EVALUATION

In this section, we describe the results of our method on the development data first. Next, we discuss the formal run results.

3.1 Development Data

We created development and training datasets. The given assembly minutes corpus contains 529 speeches. We used 477 speeches for the training dataset and 52 speeches for the development dataset. Training dataset contains 6,551 sentences. Development dataset contains 675 sentences. We applied the method described in Section 2.1 to the datasets. We assigned the importance scores to the sentences in the training dataset. We constructed the importance prediction models trained with the datasets. We used a support vector regression (SVR) implemented using the scikit-learn library [8] with a default parameter setting. We construct models using each similarity measure. We extract important sentences on the basis of the predicted importance scores. We evaluated the summaries using the Rouge-1 score. The result is shown in Table 1. The result of the models using the average of all the similarity measures outperformed all other models. Therefore, we adopted the average of all the similarity measures on the formal run.

Table 1. Experimental results on the development dataset

Similarity measure	Rouge-1
Cosine similarity between bag-of-words	0.333
Levenshtein edit distance	0.338
Rouge-1 similarity score	0.341
Cosine similarity between sentence embedding (Word2vec)	0.306
Cosine similarity between sentence embedding (Doc2vec)	0.316
Average of all the similarity measures	**0.349**

3.2 Formal Run

In this section, we describe our methods for the formal run and the results. We constructed two methods, KitAi-PI-1 and KitAi-PI-2. KitAi-PI-1 is the method without the step of the sentence compression (**Step 2** in Figure 1) to avoid generating ill-formed summaries due to this step. KitAi-PI-2 is the method with all steps described in Section 2.

As the evaluation measure, organizers used scores in the Rouge family [4] and scores of quality questions by the participants. The quality questions are assessed by three-grade evaluation (0, 1, and 2), from the viewpoints of content, formedness, and total, respectively. Table 2, 3 and 4 show the quality question

scores, the recall scores of Rouge and the f-measure scores of Rouge, respectively. X in Table 2 is a constant representing whether acceptable summaries that are different from the gold standard summary are regarded as correct or not. If such summaries are regarded as correct (X=2), the score of the summary is 2. Otherwise (X=0), the score is 0. OtherSysAve in Table 2, 3 and 4 denotes the average scores of all the submitted runs of all participants.

On the quality question scores in Table 2, all the scores of KitAi-PI-1 outperformed the scores of KitAi-PI-2 and the OtherSysAve. The content (X=2) score and the total score of KitAi-PI-1 were the best scores in all the submitted runs of all participants. The results show the effectiveness of our method. The formedness score of KitAi-PI-2 was the lower score than the score of the OtherSysAve. The result shows that KitAi-PI-2 generated ill-formed summaries due to the sentence compression step. The improvement of the sentence compression step is important future work.

On the Rouge scores in Table 3 and 4, all the scores of KitAi-PI-1 outperformed the scores of the OtherSysAve. The recall score and the f-measure of Rouge N4 on Content word were the best scores in all the submitted runs of all participants. The results show that KitAi-PI-2 can generate summaries containing important phrase although it generates ill-formed summaries.

Table 2. Quality question scores in Formal run (the boldface indicates the best score in KitAi-PI-1, KitAi-PI-2, and OtherSysAve. † indicates the best score in the all the submitted runs of all participants.)

	content		formed	total
	X=0	X=2		
KitAi-PI-1	**0.856**	**1.134**†	**1.732**	**0.912**†
KitAi-PI-2	0.788	1.035	1.308	0.667
OtherSysAve	0.423	0.603	1.655	0.435

4 CONCLUSIONS

This paper described a summarization system for NTCIR-14 QA Lab-PoliInfo. Our method extracted important sentences to summarize an assemblyperson's speech in the minutes. Our method used a machine learning approach to predict the importance of sentences. We constructed training data for the sentence importance prediction model construction automatically. For the formal run, several scores by our method were the best score in all the submitted runs of all participants. The result showed the effectiveness of our method. However, our method often generated ill-formed summaries. The improvement of the sentence compression step in our method is important future work.

Table 3. Recall scores of Rouge in Formal run (the boldface indicates the best score in KitAi-PI-1, KitAi-PI-2, and OtherSysAve. † indicates the best score in the all the submitted runs of all participants.)

		Recall						
		N1	N2	N3	N4	L	SU4	W1.2
Surface form	KitAi-PI-1	**0.440**	**0.185**	**0.121**	**0.085**	**0.375**	**0.217**	**0.179**
	KitAi-PI-2	0.390	0.174	0.113	0.078	0.320	0.200	0.154
	OtherSysAve	0.282	0.096	0.058	0.038	0.240	0.119	0.112
Stem	KitAi-PI-1	**0.458**	**0.199**	**0.134**	**0.096**	**0.389**	**0.234**	**0.188**
	KitAi-PI-2	0.399	0.179	0.118	0.082	0.326	0.208	0.158
	OtherSysAve	0.290	0.102	0.062	0.042	0.247	0.125	0.115
Content word	KitAi-PI-1	**0.285**	**0.145**	**0.090**	0.050	**0.278**	**0.154**	**0.180**
	KitAi-PI-2	0.254	0.126	0.083	**0.053**†	0.247	0.131	0.156
	OtherSysAve	0.145	0.059	0.034	0.019	0.139	0.065	0.088

Table 4. F-measure scores of Rouge in Formal run (the boldface indicates the best score in KitAi-PI-1, KitAi-PI-2, and OtherSysAve. † indicates the best score in the all the submitted runs of all participants.)

		F-measure						
		N1	N2	N3	N4	L	SU4	W1.2
Surface form	KitAi-PI-1	**0.357**	0.147	0.096	0.067	**0.299**	0.168	**0.188**
	KitAi-PI-2	0.343	**0.154**	**0.101**	**0.069**†	0.281	**0.173**	0.176
	OtherSysAve	0.272	0.088	0.051	0.033	0.232	0.109	0.136
Stem	KitAi-PI-1	**0.373**	0.159	**0.106**	**0.075**†	**0.311**	**0.182**	**0.199**
	KitAi-PI-2	0.351	**0.160**	**0.106**	0.074	0.286	0.180	0.181
	OtherSysAve	0.281	0.093	0.055	0.036	0.238	0.115	0.140
Content word	KitAi-PI-1	**0.224**	**0.115**	**0.071**†	0.042	**0.217**	**0.107**	**0.170**
	KitAi-PI-2	0.214	0.109	0.069	**0.046**†	0.208	0.106	0.159
	OtherSysAve	0.128	0.050	0.027	0.016	0.123	0.051	0.093

References

1. Katragadda, R., Pingali, P., Varma, V.: Sentence position revisited: A robust light-weight update summarization baseline algorithm. In: Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3). pp. 46–52 (2009)
2. Kimura, Y., Shibuki, H., Ototake, H., Uchida, Y., Takamaru, K., Sakamoto, K., Ishioroshi, M., Mitamura, T., Kando, N., Mori, T., Yuasa, H., Sekine, S., Inui, K.: Overview of the ntcir-14 qa lab-poliinfo task. In: Proceedings of the 14th NTCIR Conference (2019)
3. Kubo, T.: Mecab: Yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net/
4. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the eighth workshop on the Annual Meeting of the Association for Computational Linguistics (ACL-04). pp. 74–81 (2004)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: The International Conference on Learning Representations: Workshops Track (2013)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26. pp. 3111–3119 (2013)
7. Ouyang, Y., Li, W., Lu, Q., Zhang, R.: A study on position information in document summarization. In: Proceedings of the 23rd International Conference on Computational. pp. 919–927 (2010)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay., E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**, 2825–2830 (2011)