Data expansion method by combining unnecessary sentence deletion and most important sentence addition

Tomohito Ouchi, Masayoshi Tabuse
Graduate School of Life and Environmental Sciences, Kyoto Prefectural University,
1-5 Shimogamohangi-cho, Sakyo-ku, Kyoto 606-8522, Japan
E-mail:t_ouchi@mei.kpu.ac.jp, tabuse@kpu.ac.jp

Abstract

We are studying data expansion methods in automatic summarization systems. In our research, the method of expanding the input article with unnecessary sentences deleted is the most effective of the extended methods. In the previous research, we have tried a method of adding most important sentences. In this research, we propose a method that combines the deletion of unnecessary sentences and the addition of most important sentences. We propose a hybrid method with two methods, one is to add important sentences first and the other is to add important sentences last.

Keywords: automatic summarization, data augmentation, Pointer-Generator Model, Extractive summarization.

1. Introduction

Currently, the amount of information on the Internet is expected to increase at an average annual rate of 29% from 2010 to 2024, reaching 143 ZB by 2024 [1]. In terms of text data, the number of websites worldwide in 2018 was about 1.6 billion, however, it was about 1.8 billion in 2021 [2]. Since it has increased by about 200 million in three years, it is expected that text data will increase steadily in the future. Under such circumstances, the issue of selecting information is an urgent problem. Automatic summarization struggles that issue. However, it can be said that extractive summarization that only made up with sentences is not sufficient. Since the sentence-to-sentence connection is not taken into consideration, readability is lacking. Therefore, it is needed generative summarization as a technology that looks ahead. A generative summarization basically uses the Encoder-Decoder model, which learns the relationship between input and output and generates one word at a time in the output when a new input comes in during the test. Various models have been proposed [3,4]. In this study, the Pointer-Generator model [3] uses as the baseline model. One of the issues with the generative summarization model is that data maintenance is costly. We have to attach a manual summary to each article in order to make the generative summarization model. Therefore, we focused on data augmentation as a method that can be applied to any model. This is to create extended data from existing data. As a result, it was confirmed that the accuracy of the evaluation metric ROUGE [5] of Pointer-Generator model applied by the data augmentation method is improved by about 1% compared to baseline model.

Next, we explain the method of data augmentation simply. We decide the importance of each sentence in each article. To decide the importance of sentences, we used the topic model. In the existing research [6], the sentence with the lowest importance is removed to obtain extended data. Furthermore, a previous study [7] examined the effect of expansion by adding the most important sentences. This method is further divided into two methods depending on the position of addition. The method of adding to the beginning of existing data is called “add-s”, and the method of adding to the end of existing data is called “add-e”. Also, the method by removing the lowest important sentence is called “remove”. This method was proposed method in [7].

And in this research, we propose a new method. It's a combination of “remove” and “add”. It is called a “hybrid”. The method that combines “remove” and “add-e” is called “hybrid-e”. The method that combines “remove” and “add-s” is called “hybrid-s”. These five techniques (“remove”, “add-e”, “add-s”, “hybrid-e”, and “hybrid-s”) are described in Section 2. Experiments and results are described in Section 3. And discussions are given in Section 4.

2. Data Augmentation Method

This section describes the five models used in the data augmentation method. In each method, each sentence is scored in an article, and create the extended data using this score.

2.1. Topic Model

The topic model is used in the existing method and the new method. For how to determine the importance of sentences using the topic model, we referred to existing research [8]. The topic model is one of the language models that assumes that one document consists of multiple topics. In addition, each topic has an appearance word distribution. The method of determining the importance of a sentence is as follows.

1. Calculate the frequency of occurrence in a topic with words that make up a sentence
2. Sum of all the words that make up the sentence
3. Divide by the square root of the sentence length
4. Sum on all topics

2.2. Proposed method

The five methods ("remove", "add-e", "add-s", "hybrid-e", "hybrid-s") use the topic model. First of all, we calculate the score of sentences importance in input one article using topic model. In the “remove” method, the lowest important sentence is removed to existing data. In the “add-e” method, the highest important sentence is added to end of existing data. In the “add-s” method, the highest important sentence is added to beginning of existing data. In the "hybrid" method, important sentences are added and then unnecessary sentences are deleted. Figure 1 shows the remove method processing procedure.

Proposed Method
1. make topic model
2. $r \leftarrow$ None
3. $i \leftarrow 0$
4. For $s \in S$
   $i_s \leftarrow$ calculate score(s, topic model)
   If $i < i_s$
   $r \leftarrow s$
   $i \leftarrow i_s$
5. $E \leftarrow S$ remove r

出力 E

Figure 1 Proposed method(“remove”) processing procedure

3. Experiment and Results

3.1. Parameter Setting

The CNN / DailyMail dataset is used as the dataset for training, evaluating, and testing. The training data, evaluating data, and test data are 287,226 articles, 13,768 articles, and 11,490 articles, respectively. The model used for the experiment is the Pointer-Generator model, which is composed of a copy mechanism and a coverage mechanism when learning. In the Copy mechanism we calculate the error of the evaluating data each time the epoch ends and we use the model of the epoch with the lowest error in Early Stopping. Early Stopping what we mean here, uses a model that waits twice as many epochs as the error seems to be the minimum, unless the minimum value is updated. Next, in the coverage mechanism, the same processing is performed in the coverage loss. We use ROUGE as using for evaluation on existing research.

The program used in this research uses PyTorch. It has been confirmed that this program can achieve the same result as [3]. The hidden layer vector size was set to 256 and the embedded vector size was set to 128. The batch size was set to 8. In the original paper, the batch size is 16, so double learning is required to learn the same number of articles. The beam size was set to 4. The beam search will be described later. The number of vocabulary was set to 50,000. The learning rate was set to 0.15.

In this program, the number of words used to encode an input article is limited to 400. This setting has no effect on learning an extended data. Specifically, an extended data is the same as an original data. This is because the extracted sentence may not be within 400 words from the beginning. We must confirm that the extracted sentence is present in the input article. Therefore, I found the article with the most number of words among the articles used in the training data. The number of words with the most words was 2,380. And

the upper limit of the number of words used in encoding the input article was set to 2,380. Table 1 shows the values of ROUGE when the maximum number of words is 400 and 2,380. In the Table 1, f, r, and p represent the F value, recall, and precision, respectively.

Table1 the values of ROUGE when the maximum number of words is 400 and 2,380

	ROUGE-1-f	ROUGE-1-r	ROUGE-1-p	ROUGE-2-f	ROUGE-2-r	ROUGE-2-p
400	0.3935	0.4372	0.3800	0.1709	0.1891	0.1662
2380	0.3958	0.4181	0.3994	0.1741	0.1832	0.1770

	ROUGE-L-f	ROUGE-L-r	ROUGE-L-p
400	0.3616	0.4014	0.3493
2380	0.3644	0.3846	0.3679

Table 1 shows when the upper limit of the number of words is increased from 400 to 2,380, the value of ROUGE increases slightly. In the following, the experiment is performed with the upper limit of the number of words set to 2,380.

In this experiment, we calculated the average ROUGE value of three experiments in order to eliminate the randomness of the parameters as much as possible.

3.2. Beam search

Greedy method contrasts with beam search. Specifically, in greedy method, when generating a word, one word with the highest generation probability is selected, while in beam search, processing is performed while holding the top K words. Then, we make the final summarizations by multiplying the probabilities of each word generation, and make the highest one the final summarization. In this experiment, this K value is set to 4. The following Table 2 summarizes the parameter settings.

Table 2 Parameter settings

hidden vector size	256
embbed vector size	128
batch size	8
beam size	4
vocabulary size	50,000
lerning rate	0.15
input word size	2,380

3.3. Experimental Results

The results are shown in Table 3, 4 and 5.

Table 3 Results of learning 115,000 articles using 6 methods

	normal	remove	add-e	add-s	hybrid-e	hybrid-s
ROUGE-1	0.3441	0.3519	0.3470	0.3436	0.3429	0.3462
ROUGE-2	0.1389	0.1426	0.1400	0.1382	0.1374	0.1394
ROUGE-L	0.2968	0.3025	0.2977	0.2963	0.2945	0.2970

Table 4 Results of learning 57,000 articles using 6 methods

	normal	remove	add-e	add-s	hybrid-e	hybrid-s
ROUGE-1	0.3137	0.3302	0.3231	0.3210	0.3256	0.3302
ROUGE-2	0.1145	0.1272	0.1239	0.1183	0.1264	0.1269
ROUGE-L	0.2676	0.2828	0.2773	0.2745	0.2806	0.2813

Table 5 Results of learning 28,000 articles using 6 methods

	normal	remove	add-e	add-s	hybrid-e	hybrid-s
ROUGE-1	0.3056	0.3188	0.3043	0.3265	0.3187	0.3052
ROUGE-2	0.1079	0.1201	0.1048	0.1252	0.1212	0.1050
ROUGE-L	0.2595	0.2711	0.2601	0.2790	0.2730	0.2573

Table 3 shows the results of the 6 methods (normal, remove, add-e, add-s, hybrid-e, hybrid-s) when 115,000 articles were trained, Table 4 when 57,000 articles were trained, and Table 5 when 28,000 articles were trained. “normal” method is baseline model.

3.4. Disccusion

Among all the number of articles, “remove” showed the best effect of expansion. In training data 115,000 articles, “remove” showed the best effect. Next, “add-e” had good effect. In training data 57,000 articles, “hybrid-s” and “remove” showed the best effect. In training data 28,000 articles, “add-s” showed the best effect. In training data 28,000 articles, “add-s” showed the best effect. Next, “remove” and “hybrid-e” had almost the same good effect. “add-e”, “add-s”, “hybrid-e”, and “hybrid-s” were not found to be superior or inferior in this experiment. For “remove”, too few or too many articles seemed to reduce the effect. However, the number of articles in the middle seemed to be the most effective. In order to verify this, the following additional experiment was conducted.

3.5. Additional Experiment

The purpose of this experiment is to see at which number of articles the effect is most visible. For this purpose, we tried experiments with training data of 10,000, 28,000, 45,000, 57,000, 90,000, 180,000, and 287,226 articles. The methods used were “normal” and “remove”. The results are shown in Table 6 and 7.

Table 6 Results of each the number of articles by "normal" and "remove"

	10,000	28,000	45,000	57,000
normal	0.3226	0.3358	0.3398	0.3340
remove	0.3335	0.3473	0.3506	0.3591
difference	0.0109	0.0115	0.0108	0.0251

	90,000	180,000	287,226
normal	0.3538	0.3758	0.3841
remove	0.3627	0.3827	0.3916
difference	0.0089	0.0069	0.0075

Thus, with 57,000 articles at the top, the expansion effect of remove is getting worse for both fewer and more articles than that. And we can also see that it gets better for fewer articles than for more. It is expected that the reason why it got worse with more articles is that it is not possible to generate a more varied input article vector. The reason why it was worse for fewer articles is that the input article vector was biased. Since the expansion is only a slight modification of the original article, if the vector is too biased, it will remain biased even after the expansion.

4. Conclusion

In this study, we proposed "hybrid-e" and "hybrid-s" in addition to the existing work [7]. Both methods showed the effect of expansion, but not better than "remove"." The reason why "add-e", "add-s", "hybrid-e", and "hybrid-s" had a certain effect was because the extended data was readable and understandable to humans. In the existing study [6], the EDA method was not so effective because the readability of the extended data was reduced.

In this study, we also examined at what number of articles the effect of expansion is most apparent. As a result of the verification, it was found that the effect of the expansion was diminished for both too few and too many articles.

As a future task, we would like to show the high applicability of the proposed method by verifying whether the effect of the expansion can be seen in a better model that is currently being devised [9], although in this study we only tested the Pointer-Generator model.

References

1. Worldwide Global DataSphere Forecast, 2020–2024: The COVID-19 Data Bump and the Future of Data Growth IDC ANALYZE THE FUTURE. https://www.idc.com/getdoc.jsp?containerId=US44797920
2. Total number of Websites Internet Live Stats. https://www.internetlivestats.com/total-number-of-websites/
3. Abigail. See, Peter J. Liu, et al., "Get To The Point: Summarization with Pointer-Generator Networks", arXiv:1704.04368 (2017)
4. Yang Liu, Mirella Lapata, "Text Summarization with Pretrained Encoders", arXiv:1908.08345v2 (2019)
5. Chin-Yew Lin, "ROUGE: A Package for Automatic Evaluation of Summaries", Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain (2004)
6. T. Ouchi, M. Tabuse, "Effectiveness of Data Augmentation in Pointer-Generator Model", Journal of Advances in Artificial Life Robotics, Vol. 1, No. 2, pp.95-99 (2020)
7. T. Ouchi, M. Tabuse, "Comparison of Data Augmentation Methods in Pointer–Generator Model", Journal of Robotics Networking and Artificial Life, Vol 8, No. 2, pp.85-89(2021)
8. H.Sigematsu, I.Kobayashi, "Generation of abstracts considering importance of potential topics", The Association for Natural Language Processing (2012) (In Japanese).
9. Liu, Y. and Lapata, M. "Text Summarization with Pretrained Encoders." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) pp. 3730-3740 (2019)

Authors introduction

Mr. Tomohito Ouchi

He received his Master's degree from Department of Environmental Science, Graduate School of Life and Environmental Sciences, Kyoto Prefectural University, Japan in 2019. He is currently a Doctoral course student in Kyoto Prefectural University, Japan.

Dr. Masayoshi Tabuse

He received his M.S. and Ph.D. degrees from Kobe University in 1985 and 1988 respectively. From June 1992 to March 2003, he had worked in Miyazaki University. Since April 2003, he has been in Kyoto Prefectural University. His current research interests are machine learning, computer vision and natural language processing. IPSJ and IEICE member.