Journal of Robotics, Networking and Artificial Life
Vol. **8**(**2**); September (2021), *pp.* 85–89
DOI: https://doi.org/10.2991/jrnal.k.210713.003; ISSN 2405-9021; eISSN 2352-6386
https://www.atlantis-press.com/journals/jrnal

JOURNAL OF ROBOTICS, NETWORKING AND ARTIFICIAL LIFE

Research Article

Comparison of Data Augmentation Methods in Pointer–Generator Model

Tomohito Ouchi[*], Masayoshi Tabuse

Graduate School of Life and Environmental Sciences, Kyoto Prefectural University, 1-5 Shimogamohangi-cho, Sakyo-ku, Kyoto 606-8522, Japan

ARTICLE INFO

Article History

Received 19 October 2020
Accepted 01 May 2021

Keywords

Automatic summarization
data augmentation
pointer–generator model
extractive summarization

ABSTRACT

In this research, we proposed a data augmentation method using topic model for Pointer–Generator model. This method is that adding important sentences to an article as extended article. Furthermore, we compare our proposed method with data augmentation methods using Easy Data Augmentation (EDA), LexRank and Luhn. EDA consists of synonym replacement, random insertion, random swap, and random deletion. LexRank is based on Google's search method and Luhn defines sentence features and ranks sentences. We considered which method is suitable for data augmentation. We confirm that most accurate model is the model using data augmentation method by topic model.

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 3 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 Recall Oriented Understudy for Gisting Evaluation (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. In the previous proposed method [6], an extended article is made by removing unimportant sentence from original article. We humans do not consider unimportant sentences when making summaries. Therefore, in this research we propose a new method that adding important sentences to an article as extended article. We humans usually repeat important parts of article when making summaries. In addition, as a comparison method, a method called Easy Data Augmentation (EDA) [10] was used. This method was adapted to document classification, and we adapted it to the automatic summarization system. In the previous proposed method, the topic model was used as the sentence importance determination method, but as the comparison method, LexRank [8] and Luhn [7] was used as the sentence importance determination method. The new proposal 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 previous method which removing the lowest important sentence is called "remove". This method was proposed method in Ouchi and Tabuse [6]. These six techniques ("EDA", "LexRank", "Luhn", "remove", "add-e", "add-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 methods used in the data augmentation method. In each method except EDA, each sentence is scored in an article, and create the extended data using this score.

2.1. EDA

Easy Data Augmentation is a data augmentation method for tasks in document classification such as polarity classification and review

[*] *Corresponding author. Email: t_ouchi@mei.kpu.ac.jp*

estimation. We chose EDA as a comparison method because it is a data augmentation that changes the article itself. The outline of the data augmentation method is as follows.

- Synonym Replacement (SR)
  Replace a word with a synonym
- Random Insertion (RI)
  Insert synonyms of a word at random positions in the article
- Random Swap (RS)
  Randomly choose two words in the article and swap their positions
- Random Deletion (RD)
  Randomly remove a word in the article

In EDA, the frequency of the above four methods is calculated by multiplying the total number of words used in an article by the hyperparameter α. In the existing study [10], α is set to 0.1 when the number of training articles is more than 5000, so in this study α is set to 0.1 when the number of training articles is more than 5000. According to a paper on EDA, the best results were obtained when the number of articles to be expanded for one article was set to 4. In this study, each method of EDA (SR, RI, RS, and RD) are used as comparison methods.

2.2. LexRank

First, we explain PageRank, which is the basis of LexRank. The basic idea of PageRank is that many linked pages are good pages, and links from more linked pages are evaluated highly. This rating is equivalent to the user inflow to the page. This is because if links are provided from many pages, it is easy to flow in, and the inflow from popular pages is larger than the inflow from normal pages. Links between pages can be represented by a matrix, which is the probability that the user will transition from that page to another linked page. The matrix is made with dividing by the total number of links on each page. The purpose of PageRank is to use this matrix to determine the probability that a user will stay on each page, that is, the rating of the page. PageRank is based on the premise that the page stay probability will eventually stabilize if the page transition is repeated many times, so that the transition matrix multiplied by the stay probability vector becomes closer to the transition matrix.

In LexRank, the transition matrix is he matrix of the cosine similarity of the Tf-Idf score between sentences. The basic idea of LexRank is that sentences similar to many sentences and sentences similar to important sentences are considered to be important sentences. For creating extended data in this method, the sentence of the lowest score is removed from existing data. And, this method is called “DA-LexRank”.

2.3. Luhn

We measure position of the top 100 in most frequent words removed stop-words. We define words with a distance of 5 or less as one cluster. The score of a cluster is the square of the number of important words in a cluster divided by the distance between the first and last words of the cluster. Finally, the maximum score of each cluster becomes the score of the sentence. For creating extended data in this method, the sentence of the lowest score is removed from existing data. And, this method is called “DA-Luhn”.

2.4. 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 [9]. 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 score of 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.

2.5. Proposed Methods

The three methods (“remove”, “add-s”, “add-e”) use 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-s” method, the highest important sentence is added to beginning of existing data. In the “add-e” method, the highest important sentence is added to end of existing data.

3. EXPERIMENT AND RESULTS

3.1. Parameter Setting

The CN/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 uses 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 See et al. [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.

Table 1 | The values of ROUGE when the maximum number of words is 400 and 2380

	ROUGE-1-*f*	ROUGE-1-*r*	ROUGE-1-*p*	ROUGE-2-*f*	ROUGE-2-*r*	ROUGE-2-*p*	ROUGE-L-*f*	ROUGE-L-*r*	ROUGE-L-*p*
400	0.3935	0.4372	0.3800	0.1709	0.1891	0.1662	0.3616	0.4014	0.3493
2380	0.3958	0.4181	0.3994	0.1741	0.1832	0.1770	0.3644	0.3846	0.3679

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 2380. And the upper limit of the number of words used in encoding the input article was set to 2380. 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.

Table 1 shows when the upper limit of the number of words is increased from 400 to 2380, the value of ROUGE increases slightly. In the following, the experiment is performed with the upper limit of the number of words set to 2380.

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. 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
learning rate	0.15
input word size	2,380

3.3. Results

The results are shown in Tables 3–5.

Table 3 shows the results of the 10 methods (normal, EDA, LexRank, Luhn, proposed methods) when 287,226 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.

The new proposed method performed better than the baseline model except when 28,000 articles were used. Among the proposed methods, “remove” gave the best results. However, the new proposed method gave worse results than the previous proposed method. The reason for this is that the Pointer–Generator model becomes more difficult to learn as the number of input words increases. Therefore, as a future task, we must confirm a method of combining with the previous proposed method and the new proposed method.

Table 3 | Results of learning 287,226 articles using 10 methods

	normal	SR	RI	RS	RD
ROUGE-1	0.3841	0.3865	0.3876	0.3760	0.3828
ROUGE-2	0.1631	0.1702	0.1674	0.1633	0.1667
ROUGE-L	0.3516	0.3554	0.3563	0.3448	0.3520
	DA-LexRank	DA-Luhn	remove	add-e	add-s
ROUGE-1	0.3777	0.3739	0.3916	0.3850	0.3908
ROUGE-2	0.1616	0.1640	0.1702	0.1666	0.1698
ROUGE-L	0.3489	0.3449	0.3613	0.3519	0.3587

Table 4 | Results of learning 57,000 articles using 10 methods

	normal	SR	RI	RS	RD
ROUGE-1	0.3340	0.3487	0.3402	0.3264	0.3390
ROUGE-2	0.1289	0.1389	0.1289	0.1274	0.1308
ROUGE-L	0.3096	0.3235	0.3147	0.2999	0.3138
	DA-LexRank	DA-Luhn	remove	add-e	add-s
ROUGE-1	0.3309	0.3502	0.3591	0.3510	0.3424
ROUGE-2	0.1270	0.1386	0.1454	0.1396	0.1311
ROUGE-L	0.3078	0.3244	0.3321	0.3249	0.3167

Table 5 | Results of learning 28,000 articles using 10 methods

	normal	SR	RI	RS	RD
ROUGE-1	0.3358	0.3506	0.3420	0.3298	0.3427
ROUGE-2	0.1303	0.1399	0.1330	0.1214	0.1311
ROUGE-L	0.3122	0.3247	0.3180	0.3071	0.3175
	DA-LexRank	DA-Luhn	remove	add-e	add-s
ROUGE-1	0.3275	0.3007	0.3473	0.3395	0.3300
ROUGE-2	0.1197	0.0833	0.1370	0.1292	0.1253
ROUGE-L	0.3027	0.2754	0.3206	0.3143	0.3054

A summary example generated in each model is shown.

Reference

Kitten Tilly had to be put down after inspectors found her ‘clearly dying’ teenager threw the cat against walls, dangled her into bathwater by her tail and flushed her head in the toilet in disturbing footage. cat was locked in the bathroom 24 h a day and never fed. 16-year-old claimed he was ‘accidentally’ high on ‘white rhino’ marijuana. attacker and owner, Sarah reeves, given 10-year ban from keeping animals.

Normal

Tilly had to look away as disturbing footage of tilly meowing in distress. he claimed he was ‘accidentally ‘high on’ white rhino’ - and was being egged on by friends when the abuse occurred. owner Sarah reeves was filmed throwing her cat against the walls.

SR

Defendant flushes Tilly's head in the toilet while high on super-strength cannabis. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. Tilly threw her against the door was played in swindon magistrates court.

RI

Defendant flushes Tilly's head in the toilet while high on super-strength cannabis. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. RSPCA inspectors found her that she had to be put to sleep.

RS

Defendant flushes Tilly's head in the toilet while high on super-strength cannabis. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. she was in such a sorry state when RSPCA inspectors found her.

RD

Defendant flushes Tilly's head in the toilet while high on super-strength cannabis. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. RSPCA inspectors said she suffered a lot in her short life.

DA-LexRank

Defendant flushes Tilly's head in the toilet while high on super-strength cannabis. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. RSPCA inspectors said she suffered a lot in her short life.

DA-Luhn

A 16-year-old filmed throwing a kitten against a door and flushing it in the toilet. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. it had to be put down because of the abuse it suffered.

remove

Tilly meowing had to look away as he threw her against the door. he claimed he was 'accidentally' high on 'white rhino' and was being egged on by friends when the abuse occurred. owner Sarah reeves, 19, was also given a 10-year ban from keeping animals.

add-e

The teenager, who cannot be named for legal reasons, had to look away as he threw her against the door was played in swindon magistrates court. he claimed he was 'accidentally' high on 'white rhino' and was being egged on by friends when the abuse occurred.

add-s

The teenager, who cannot be named for legal reasons, had to look away as disturbing footage of tilly meowing in distress as he threw her against the door was played in swindon magistrates court. he claimed he was 'accidentally' high on 'white rhino' - super-strength cannabis. the defendant admitted to two counts of causing unnecessary suffering and was given a 9-month referral order and banned for keeping animals for 10 years.

4. CONCLUSION

In this study, in addition to the existing study [6], we experimented with six data augmentation method ("EDA", "DA-LexRank", "DA-Luhn", "remove", "add-e", "add-s"). The results confirmed that the best data augmentation method is to use the topic model of the existing research. In the future, we would like to confirm the effectiveness of data augmentation for state-of-the-art models. When extracting a sentence from an article, I would like to try a method for extracting multiple sentences instead of one sentence. We also expect that the number of sentences will change depending on the length of the article. And, we must combine the previous proposed method and the new proposed method.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

REFERENCES

[1] D. Reinsel, J. Rydning, J.F. Gantz, Worldwide Global DataSphere Forecast, 2020–2024: the COVID-19 Data Bump and the Future of Data Growth, IDC, 2020.

[2] Total number of Websites, Internet Live Stats.

[3] A. See, P.J. Liu, C.D. Manning, Get to the point: summarization with pointer-generator networks, arXiv:1704.04368, 2017.

[4] Y. Liu, M. Lapata, Text summarization with pretrained encoders, arXiv:1908.08345v2, 2019.

[5] C.Y. Lin, ROUGE: a package for automatic evaluation of summaries, Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Association for Computational Linguistics, Barcelona, Spain, 2004 , pp. 74–81.

[6] T. Ouchi, M. Tabuse, Effectiveness of data augmentation in pointer-generator model, 2020 International Conference on Artificial Life and Robotics (ICAROB2020), ICAROB2020, Beppu, Oita, Japan, 2020, pp. 390–393.

[7] H.P. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (1958), 159–165.

[8] G. Erkan, D.R. Radev, LexRank: graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. 22 (2004), 457–479.

[9] H. Sigematsu, I. Kobayashi, Generation of abstracts considering importance of potential topics, The Association for Natural Language Processing, 2012 (in Japanese).

[10] J. Wei, K. Zou, EDA: easy data augmentation techniques for boosting performance on text classification tasks, arXiv: 1901.11196v2[cs.CL], 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 PhD 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, IEICE and RSJ member.