Effect of Semantic Content Generalization on Pointer Generator Network in Text Summarization

YIXUAN WU KEI WAKABAYASHI
Faculty of Library, Information and Media Science
University of Tsukuba
s1921656@s.tsukuba.ac.jp kwakaba@slis.tsukuba.ac.jp

Abstract

Semantic content generalization is a method proposed recently for text summarization that reduces the difficulty of training of neural networks by replacing some phrases such as named entities with generalized terms. The semantic content generalization has achieved remarkable results in enhancing the performance of the sequence to sequence attention model. Besides that, the pointer generator network has been proposed to ease the training of the summarization based on a mechanism that copies words from the original text, which shares a similar idea with semantic content generalization. The purpose of this work is to test and verify the effect of semantic content generalization on the pointer generator network. Therefore, we use the pre-processing of semantic content generalization and then combine it with the pointer generator network. We examine the performance through an experiment using CNN / DailyMail data set.

1 Introduction

With the explosive growth of text information in recent years, people can access massive text information every day such as news, blog, report, paper, etc. Therefore, it is very necessary to do the automatic text summarization.

The purpose of automatic text summarization is to retain the most important content in long articles and summarize the articles to get short texts. Automatic text summarization can be divided into two types: extractive and abstractive. Extractive summarization is a method that judges the important sentences in the original text and extracts these sentences as a summary. The abstractive method uses the advanced natural language processing algorithm to generate a more concise summary through the technology of paraphrase, synonym replacement, sentence abbreviation and so on.

In 2017, See et al. [**?**] proposed a summary model based on the pointer generator network. The pointer generator network can train fast, easier to generate words from the source text, and it can even copy informal words in the source text. In 2019, Panagiotis Kouris et al. [**?**] proposed a semantic content generalization method for abstract text summarization based on the sequence to sequence attention model. Inspired by these works, this paper attempt to merge semantic content generalization into the pointer generator network. These two independent but combinable methods seem to have a similar idea as we explain in the following sections, therefore, we need to examine whether the combination can improve the pointer generator network or not.

2 Related work

In 2016, Nallapati et al. [**?**] improved the abstractive summarization model with multi-point improvement on the basis of encoder-decoder structure and achieved better performance through training and testing based on English corpus. There are four improvements in the model. One is to introduce more external linguistic feature information. The second is the hierarchical attention mechanism, which extracts the attention information of word level and sentence level respectively, and then multiplies them correspondingly. The third is to introduce a simple pointing judgment mechanism of generating and copying of the original text to alleviate the occurrence of unknown words and low-frequency words. The fourth is to solve the problem of too large prediction vocabulary of decoder by negative sampling [**?**].

In 2017, See et al. [**?**] proposed a summary model based on the pointer generator. In terms of model architecture, it is an improved model based on the sequence to sequence model. The author of this paper proposes that only using sequence to sequence model to generate summary will bring two problems. One is that we may not be able to accurately reproduce the details, and you may not be able to deal with out-of-vocabulary words (OOV). The second is the possibility of repetition. The pointer generator network is improved from two aspects. First,

the network can copy words from the source text by pointer, which helps to copy information accurately. At the same time, it can ensure the ability to generate new words through generators. Second, the method uses a coverage mechanism to track what has been summarized to prevent duplication. Specifically, the method solves the problem of unknown words and low-frequency words by combining the generation probability and copy probability of abstract words by introducing weight, and by referring to the idea of solving the problem of "over translation" and "missing translation" in machine translation [**?**]. The attention weight of some words which is decoded at the decoder side is taken into consideration in the objective function of generating a summary for reducing the probability of generating the same word, to some extent alleviate the problem of repeated words in the summary. This pointer generator network model was tested on the English dataset and achieved a better performance than the summary model proposed by Nallapati et al. [**?**].

In 2019, Panagiotis Kouris et al. [**?**] proposed a semantic content generalization method on abstract text summarization based on the sequence to sequence attention model. The method aims at enhancing the performance of abstractive summarization based on the sequence to sequence model by replacing some phrases with its generalized terms. They proposed two approaches for generalization that are called named-entity generalization (NEG) and level-based generalization (LG).

NEG generalizes only those named-entities (NEs) that are detected by named entity recognizer trained to detect entities of location, person, and organization. LG uses the concept of generalization to preprocess based on a dictionary such as WordNet. It is mainly to replace low-frequency words with high-frequency words. For example, the method turns bananas into food. The dataset adopted by the author for evaluating the method was Gigaword and DUC2004.

3 Methods

3.1 Pointer Generator Network

The pointer generator network is not much different from the traditional sequence to sequence network [**?**]. For each time step of the decoder, calculate a probability that is between 0 and 1. This probability determines the probability of generating words from the vocabulary, rather than the probability of copying words from the source text. The final distribution is obtained by weighting and summing the vocabulary distribution and attention distribution, and predicting a word based on this. Therefore, the pointer generator network can either copy words through pointers or generate words from fixed vocabulary. It has the following advantages:

1. Pointer generator networks make it easier to generate words from source text. The network just needs to focus enough attention on the relevant words and make P_{gen} big enough [**?**].

2. The pointer generator network can even copy informal words in the source text. This is also the main advantage brought by this network, so that words that have not appeared in the training corpus can be generated, and a smaller vocabulary can be used with less sacrificing performance. It also requires less computing resources and storage space.

And the author adds a coverage mechanism [**?**] to the network, which successfully solves the common repetition problem in the sequence to sequence model.

$$C^t = \sum_{t'=0}^{t-1} a^{t'} \tag{1}$$

The coverage mechanism mainly maintains the coverage vector, which is the sum of the attention distribution of all previous decoder time steps. C^t is the distribution over the source document words that is calculated using the attention mechanism about the degree of coverage.

3.2 Semantic Content Generalization

3.2.1 Pre-processing

- **Named Entities-driven Generalization**: NEG is an abbreviation of named entities-driven generalization. It generalizes only those named entities (NEs) such as location, person, and organization. We pass the dataset through the preprocessing of NEG and replace named entities whose taxonomy path contains specific named entities with a special symbol that indicates the entity types (e.g., location, person, and organization). In order to formulate NEG, the author proposes several concepts.

    1. Taxonomy of concepts: Concept classification consists of a hierarchy of concepts related to is-a relationship types.

    2. Taxonomy path of concepts: Let C_a be a concept. For a given taxonomy of concepts, C_a's taxonomy path P_{C_a} is an ordered sequence of concepts $P_{C_a} = \{C_a, C_{a+1}, ..., C_n\}$, and C_i semantically contains C_j, C_j

is the hypernym of C_i. C_n is the root concept of taxonomy.

When the frequency of terms in the input text is less than the specified threshold θ_f and its classification path p_i contains a named entity c ∈ E, it can be generalized. In this case, C_i will be replaced by its superordinate word C. The output is the generalized text (genText) of the input text. And when the threshold is equal to infinity, NEG's algorithm is similar to the operation of named entity anonymization proposed in 2018 by Hassan et al. [**?**].

Algorithm 1 Pre-processing of Named Entities-driven Generalization

Require: C = Location, Person, Organization
1: Apply Stanford NER to given documents
2: **for** each word $w \in C$ **do**
3: Replace w in the documents with C
4: **end for**

- **Dictionary-based Generalization**: On the basis of NEG, we will create a simple dictionary composed of four categories: vehicle, weather, sports and crime.

  The entries of vehicles[1], weathers[2], sports[3] and crimes[4] in the simple dictionary are all retrieved through the web.

  We added the four categories to the set of concepts. We identify phrases belonging to the categories based on the dictionary lookup. Then, CNN / DailyMail is pre-processed, and the words in the dataset that belong to the names of Person, Location, Organization, Weathers, Sports, Crimes, and Vehicles are replaced with special symbols accordingly. The labeled dataset is sent to the pointer generator network to obtain an intermediate summarization.

3.2.2 Post-processing

Finally, the post-processing is performed to obtain the final output summary. The post-processing adopted in this paper is the same as that proposed by Panagiotis Kouris et al. [**?**]. As they described in the paper, the post-processing is a problem of optimal bipartite matching. The matching is based on the similarity of the context around the generalized concepts of the summary and the candidate concepts of the text.

[1] https://englishstudyonline.org/types-of-vehicles/
[2] https://www.enhancedlearning.com/wordlist/weather.shtml#wls-id-22
[3] https://www.lingokids.com/english-for-kids/list-of-sports
[4] https://critical.findlaw.com/critical-charges/view-Allcriticalcharges.html

Algorithm 2 Pre-processing of Dictionary-based Generalization

Require: C' = Weathers, Sports, Crimes, and Vehicles
1: Apply Algorithm 1 to given documents
2: Apply dictionary matching
3: **for** each word $w \in$ C **do**
4: Replace w in the documents with C
5: **end for**

4 Experiments

4.1 Datasets

We used CNN / DailyMail's multi-sentence summarization dataset. Compared with single sentences, multi-sentence summarization are obviously more complicated. The CNN dataset consists of more than 92,000 articles and corresponding summarization, while the DailyMail dataset consists of more than 219,000 articles and corresponding summarization. The training set is 287,226 training pairs, the validation set is 13,368 training pairs, and the test set is 11,490 training pairs.

For the dataset, we apply the following process. First, all the words in the dataset are lowercased, and the numbers are replaced with #. As the text length increases, the required video memory capacity will increase linearly and the running time will grow, so the text length is often limited. As a rule of thumb, we limited the maximum length of an article to 512. Secondly, due to the characteristics of the dataset, the length of the summary is uncertain, and the length of some summary is too long, which causes the excessive memory usage of neural networks, so the maximum length of the summarization is limited to 128. And we use a vocabulary of 500, 000 words for both source and target.

4.2 Settings

For the experiments, we used the pointer generator network with 256-dimensional hidden states and 128-dimensional word embeddings. And we used the coverage mechanism that encourages anti-repetition. The other settings are exactly the same as those of See et al. [**?**]. The baseline is the pointer generator network proposed by See et al. [**?**] with the coverage mechanism. It takes about one week to train the pointer generator network. When we apply the processes for semantic content generalization with dictionary and NEG, it takes about an extra three to

	ROUGE-1	*ROUGE-2*	*ROUGE-L*
Pointer Generator	0.3739	0.1578	0.3429
Pointer Generator+NEG	0.3900	0.1602	0.2593
Pointer Generator+NEG+Dictionary	0.3799	0.1551	0.2516

Table 1: The result on CNN/DailyMail

four days compared with just training the pointer generator network.

4.3 Results

The easiest way to evaluate the quality of the summary is manual evaluation, but in order to evaluate the automatic text summarization more efficiently, one or several metrics can be selected. Based on these metrics, we can compare the generated summary with a reference summary for automatic evaluation. At present, the most commonly used and most recognized metric is ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE is a set of metrics proposed by Lin [**?**]. We evaluate our model with Rouge metrics, and report the F1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

The result is shown in Table 1. The performance of the method that combines NEG with the pointer generator network is the highest excluding ROUGE-L. From the result, we can see that the semantic content generalization can improve the pointer generator network. However, only when we compare the methods with the metrics of ROUGE-L, the semantic content generalization seems not to work well. The result of dictionary-based generalization is lower than the NEG. We think the reason is that the processing method is too simple and rough. The processing of the dictionary needs to be improved.

5 Conclusion

In this work, we tested and verified the effect of semantic content generalization on the pointer generator network. From the result, we found that the semantic content generalization can improve the pointer generator network. In future work, we need to improve the dictionary for the semantic content generalization, rather than using the Stanford NER and WordNet to obtain a better result.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 19K20333, 16H02904 and JST CREST Grant Number JPMJCR16E3 AIP Challenge.

References

[1] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[2] Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. Abstractive text summarization based on deep learning and semantic content generalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5092, 2019.

[3] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[4] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.

[5] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*, 2016.

[6] Fadi Hassan, Josep Domingo-Ferrer, and Jordi Soria-Comas. Anonymization of unstructured data via named-entity recognition. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 296–305. Springer, 2018.

[7] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.