Improving Quality of Extractive Summarization with Coverage Analysis

Haihan Yu
University of Tokyo
Graduate School of
Information and Computer Science
dhyu@logos.t.u-tokyo.ac.jp

Yoshimasa Tsuruoka
University of Tokyo
Graduate School of
Information and Computer Science
tsuruoka@logos.t.u-tokyo.ac.jp

1 Introduction

Text summarization is a classic NLP task that has a long history. Starting from probabilistic model-based approaches in the last century, researches have now shifted their attention to summarization models based on pre-trained language models. Pre-trained language models, highlighted by BERT [1] introduced in 2018, have significantly improved the state-of-the-art performance of many NLP tasks, including summarization. However, when generating summaries, pre-trained language models have shown insufficiency in detecting repetition or negligence of key information. To cope with such a problem, we propose a refinement on the model by adding a module of semantic coverage detection implemented by probabilistic or neural network models, to help alleviate the problem.

2 Background

2.1 Text Summarization

Text Summarization is the NLP task that requires the model to give a stream of texts based on the source. The output should be a shorter text which keeps the main ideas from the source. By the nature of how the summary is generated, we can further classify the task into two categories, namely extractive summarization and abstractive summarization. Extractive summarization means to select sentences from the original text as the summary, while abstractive summarization requires the system to generate words and sentences on its own.

Until recently, due to the limitation of computational power and theory, the majority of summarization models focused on probabilistic-model-based extractive summarization methods. However, with the rapid development of deep learning technology, many researchers have started to explore the potentials of machine learning based techniques. SuMMaRuNNer [2], is one of the first extractive summarization models that uses Recurrent Neural Network (RNN) and has achieved a state-of-the-art performance at the time. Then, other models, such as NeuSum [3], Sumo [4], have improved the performance of extractive summarization. Recently, with the introduction of Pre-Trained Language Model, such as BERT [1], many researchers have successfully pushed forward the performance to a higher level.

2.2 Pre-Trained Language Model

Earlier this century, apart from the summarization task, scientists have created many different models and systems to solve various NLP tasks. However, due to the fact that these models are trained for very limited (in most cases, only one) tasks and with a small amount of data, such models show great insufficiency in adjusting to the change of input.

In order to counter this problem, researchers have started implementing the idea of creating a general language model that fits for all scenarios, and such idea is now referred to as the Pre-Trained Language Model (PLM). As we can see from Fig. 1, a pre-trained language model is first fed with an enormous amount of unannotated data. The model will be able to "learn" the language in this process, and then, we can "fine-tune" this model by training it with task-specific data sets. Because the model has been fed with large amounts of data about the language, it will be easier for it to adjust itself and of what to do with specific tasks.

Pre-trained language models, with all the corpus learned before fine-tuning, can minimize the confusion of the model when seeing a new sequence of text. It has been usually used to enhance performance in language understanding tasks, and have been proven to have obtained

state-of-the-art performance in many NLP tasks. Very recently, there are successful attempts to apply pre-trained models to various language generation problems.

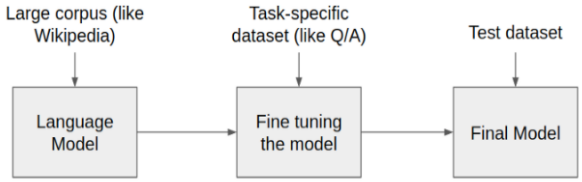

Figure 1 Generic Concept for Pre-Trained Language Model [5]

Bidirectional Encoder Representations from Transformers, or BERT [1], is arguably the most important pre-trained language model in the past 5 years. Developed by Google in 2018, it is a multi-layer bidirectional transformer encoder. BERT is a model pre-trained on two tasks, namely masked language modeling and next sentence prediction. Masked language modeling is the task in which we try to predict the input tokens that are intentionally masked at random, while next sentence prediction is a binary classification task that requires the system to determine if the second input sentence is actually directly connected to the first input sentence. The two tasks help the model to understand the language on both word-level and sentence-level, therefore allowing the model to be able to handle any downstream single sequence and sequence-pair tasks without significant task-specific modification on the structure. At the same time, some small changes are still possible to fit the specific requirements of downstream tasks, which helps it gain both flexibility and generality. With such outstanding properties, many models based on BERT has been implemented in a short period of time, and have advanced the state-of-the-art performance for many NLP tasks, including summarization. Liu et al. [6] are among the first one to apply BERT in the task of summarization. In their approach, they created a general framework that incorporates the task of extractive and abstractive summarization （we will not go too much into the detail of abstractive summary in this research） together in one model. As shown by their results, compares to other models at the time, it has improved the ROUGE score by a decent amount.

2.3 Semantic Similarity

Semantic similarity is an idea that is often used in machine translation or paraphrasing tasks as a way to measure how similar two sentences are to each other. Defining similarity of sentences is a task that involves objective justification. So many researchers have proposed different approaches to this problem. We will visit some widely-used approaches in this section.

2.3.1 Probabilistic Similarity

TF-IDF [7], introduced in 1972, is one of the first numerical statistical approaches that aims at reflecting how important a word is to a document in a collection or corpus. The metric includes two parts, which are term frequency (TF) and inverse document frequency (IDF), respectively. TF measure how important one word is to a single document, based on the assumption that the words that show up more frequently in a document are more important. While on the other hand, IDF measure how much a word shows up in a group of documents, and gives words that shows up more often a lower score. Then, TF score is multiplied to IDF score to obtain the TF-IDF score, which gives higher scores to more important words with "actual meaning" in a document, and neglects common words with no actual meaning, such as "the", "a", etc. Justified by information theory, TF-IDF is one of the most popular term-weighting systems, and some report [8] states that more than 80% of digital library uses tf-idf in their text-based recommendation system.

2.3.2 Semantic Coverage Vector

While probabilistic models are easy to implement, and gives good results in many scenarios. They have very obvious shortcomings in comparing the texts that are in different languages or have many synonyms. Therefore, many researchers have proposed other approaches to counter this problem. One effective approach is the semantic coverage vector ("SCV") [9].

SCV is introduced in an attempt to deal with the over-translation and under-translation issue in Neural Machine Translation (NMT) task. In this approach, a coverage set is maintained to keep track of which source words have been translated (or "covered") by the translation. More specifically, the coverage is defined by an attention model that scores how well the generated sentence y_j matches with the original sentence h_j called coverage vector. Such a coverage vector will keep track of the attention history, and will be fed into the attention model to help adjust future attention. In this way, it will help determine to

what degree a new sentence will bring new information, therefore, help alleviate the problem of over-translation and under-translation.

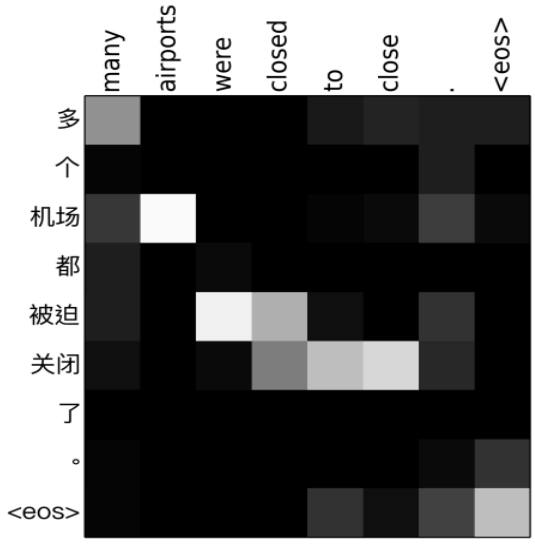

Figure 2 Semantic Coverage Vector Showing both over-translation and under-translation [9]

3 Purposed Method

As mentioned in previous sections, a PLM-based summarization model gives much improvement to the quality of extractive summarization. However, PLM-based models have shown a tendency of over-focusing the key points, thus creating repetition and negligence of information at the same time. To deal with this problem, we define the degree of coverage to be the degree of how much information from the original text is being touched in the summary, and we here make the assumption that the degree of coverage is an indicator of summary quality.

In this research, we try to explore the effect of different metrics for coverage. We will try both probabilistic model and deep learning based model. We use BERTSUM [6] as the basic structure for the model. For training, we followed the process being used in their paper. When generating the summary, we add the coverage score in as an independent factor. For an extractive summary, the coverage score is calculated for each sentence in the extractive summary task, and then a new score is obtained by combining the score given by BERTSUM model and the coverage score. We use this is as the basis of choosing sentence into the summary. TF-IDF calculation does not require extra training, we utilitize the statistical data from the dataset that is being tested to calculate the idf score, as for the semantic coverage vector approach, we train the data with the same data set for summarization.

Model	R-1	R-2	R-L
BERTSUM(benchmark) [6]	41.82	19.08	38.28
BERTSUM + TFIDF	41.90	19.20	38.31
BERTSUM + SCV	41.34	18.55	37.63

Table 1 Extractive Summarization with Information Coverage Analysis

4 Experiments Set-up and Results

4.1 Data Set

We evaluated our model on the benchmark dataset that is widely used to evaluate a summarziation task, namely the CNN/DailyMail dataset [10]. The dataset has collected the about 300,000 pieces of news from CNN and Daily Mail, each with a three-sentence summary created by the editors. The average document length is around 700 words, while the average summary length is 50 words. These summaries are treated following the standard split into training, validation, and test sets.

4.2 Evaluation

We used the standard ROUGE [11] metric to measure the quality of summaries. ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, compares the generated text against a human-produced reference text to measure their word-level similarity. ROUGE has a few variants, where ROUGE-N is calculated based on the overlap of N-grams between the generated text and the reference, while ROUGE-L is calculated based on the longest common sequence.

4.3 Results

We tested the effect of two coverage metrics on the dataset, and the result is listed in Table 1. As the numbers largely explain themselves, we have observed small improvement on ROUGE score when using TF-IDF as the metric of summary on both extractive and abstractive tasks. But the SVC metric does not bring improvements.

5 Discussion and Future Work

As shown from the result, the relatively easy method, TF-IDF, actually gives improvement to the summary qual-

ity. We believe the reason behind this improvement is that the sum of TF-IDF scores of all possible terms and documents will be able to recover mutual information between documents and terms taking into account all the specificities of their joint distribution [12]. That is, the TF-IDF score of each sentence is actually directly connected to how much information is brought by them. Such justification suggests the correctness of the idea that uses information coverage analysis for extractive summarization. However, for semantic coverage vector, as introduced in the background section, it is a method that has been originally used in the machine translation task, where the texts before and after translation are about the same size. This is not a property that the summarization task has, so due to this key difference, training SCV model with a summarization dataset without making changes to the model itself is not working as expected. We have noticed that when giving more weight to the SCV score, the ROUGE score will decrease even further. Therefore, it seems some change, or a more suitable neural model should be used for coverage analysis.

For future work, we will explore more approaches to determine the information coverage of sentences. There are some other candidate models that might work in this scenarios. We will explore if they can bring a better ROUGE score for summarization. Other than that, we will also explore if the dataset has anything to do with the score. Other than commonly used news datasets like XSum [13], the performance of models on non-traditional news datasets is also worth exploring.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of 31st AAAI Conference on Artificial Intelligence*, pp. 3075–3081, 2017.

[3] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[4] Yang Liu, Ivan Titov, and Mirella Lapata. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1745–1755, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Prakhar Ganesh. *Pre-trained Language Models: Simplified*. Accessed: 2021-01-10.

[6] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3721–3731, 2019.

[7] Sparck Jones Karen. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*.

[8] Stefan Langer Joeran Beel, Bela Gipp and Corinna Breitinger. Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*.

[9] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. 2016.

[10] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

[11] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[12] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing Management*, Vol. 39, No. 1, pp. 45 – 65, 2003.

[13] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.