NATURAL LANGUAGE PROCESSING COURSE (FALL 2022) FINAL PROJECT: TITLE GENERATION USING TWO-PHASED SUMMARIZATION TECHNIQUE

Kuldeep Singh[†] *Reza Khan Mohammadi*[†] *Ye Ma*[‡]

[†]Computer Science and Engineering Department, Michigan State University
[‡]Department of Linguistics, Languages, and Cultures, Michigan State University

ABSTRACT

The generation of titles is an interesting process. The generation of titles is a challenging process due to the significance of the title, especially in the news. There is no straightforward approach to generating the proper title. Still, titles are known to play the initial most crucial effect on the reader. In this project, we tackle title generation in the news domain. We explored several summarization techniques, mostly revolving around sequence-to-sequence architectures. Finally, we develop a two-phased title generation technique that first generates a short summary of the raw input news and, from that, generates the final title using a finetuned version of T5. We evaluated our approach using the ROUGE metric. Our final results demonstrate the significance of news' first paragraphs in generating proper titles. Our final technique yielded a ROUGE score of +++++++.

Index Terms— title generation, text summarization, abstractive summarization, extractive summarization, T5

1. INTRODUCTION

Generating a title for a given text is not an easy task. From a literature perspective, the title of a text is supposed to have certain charactaristics. For instance, it has to be expressive, meaning that it is expected to convey the heart and soul of the text's content. It also should be interesting enough to grab the attention of the audience. Also, titles can intentionally/unintentionally cause bias and be written in a way that directs readers' mindset towards a specific perspective which can act as a double-edged sword. This may cause either grab the audience's attention and cause a stream of views and money, or lead to public disappointment and cause cash loss. All things considered, the combination of such literary aesthetic values and economic significance makes title generation a challenging task.

In this project, would tackle title generation in the news domain. Having studied recent advancements in the field, we perform a series of experiments to analyze the efficiency of previously propose architectures and develop a two-phased application of title generation using extractive and abstractive summarization techniques. We also experiment with transformers such as BERT [1] and T5 [2]. We perform several experiments and ablation studies to determine means of improving the efficiency of our method. We used ROUGE [3] as our primary metric of assessing generated titles.

The remaining of this paper is organized as follows: first, we describe previous studies in section 2. Then, we fully describe our approach in section 3 and performed experiments and ablation studies in section 4. Finally, we conclude the paper in the section 5.

2. RELATED WORK

Researchers have been using the task of Title Generation to provide an apt summary for a blog, a news title for a blog, commit message for a code snippet, generate a Youtube video title using description, create a generic title for clustered documents, the title for Spoken Broadcast news, generating StackOverflow Questions given the code snippet and description, etc. To put in simple words it is basically a model which produces a one-liner summary or representation for a longer document. A document can be multiple text files, code snippets, or news articles. It can be looked at as a similar task to summarization.

One of the earlies studies in title generation is [4] which was published in 1999. This paper discusses the generation of one-sentence length summarizations by statistically learning models of both content selection and realization. Given an appropriate training corpus, they try to generate summaries similar to the training ones, of any desired length. They use newswire articles from Reuters and the Associated Press available from the LDC. This is basically a statistical summarization model and most of the work before this paper was focused on extractive summarization and information retrieval methods. Another paper that uses a probabilistic approach for the task is [5], published in 2002. This paper basically builds upon previous statistical approaches by introducing a hidden state called ”Information Source” which samples the important content word out of a document and the title will be computed based on the sampled ”information source” instead of the original document. The dataset they used comes from a

CD of 1997 broadcast news transcriptions published by Primary Source Media in 1997. There were a total of 50,000 documents and corresponding titles in the dataset. Both of these papers use a metric to measure the overlap of the words between the ground truth and model-generated titles.

The main problem the older statistical models face is the coherence of the titles generated. Since these models were evaluated just by word overlap they ignored whether the title is cohesive with the text provided or whether the words are making any sense or not for example title word ordering and even word selection. These approaches require annotated data with as much feature engineering as possible like annotation pos tagging, and requisite markup to indicate additional information, such as focus, discourse structure, or even co- or anaphoric-reference information.

The recent approaches in the field of title generation are Sequence to Sequence approaches. [6] discusses a very interesting approach of generating Question titles for stack overflow using code snippets. They created a dataset using the publicly available Stack Overflow dataset (they used different methods to identify which is poor and which is a good title for the provided text; to achieve this there are several identifiers in the metadata for eg. The number of times the question was edited etc.). They used an LSTM encoder-decoder model with attention. Additionally, they used an integrated copy mechanism (to handle the rare-words problem), and a coverage mechanism (to handle meaningless repetition). This approach is fully data-driven and doesn't rely on any handcrafted features. They also tried several different approaches on the dataset and mentioned a comparison among them, these approaches are Information Retrieval (using TFIDF), Moses (a widely used phrase-based machine translation system), and NMT (Neural Machine Translation). The evaluation metrics used are both BLUE and ROUGE scores for unigram, bigrams, and trigrams i.e. BLUE-1, BLUE-2, BLUE-3 and ROUGE-1, ROUGE-2, ROUGE-L (longest common subsequence). They also used human evaluators for better comparison of different approaches. The scores look acceptable but still not that great given we are just summarizing the code snippet to a question title.

The aforementioned paper, as we believe, missed out on a very important characteristic of a stack overflow post, the description text. The description text can be a major signal to write more cohesive and apt questions on stack overflow. Moreover, they have not done any ablation study to see whether their model works on more complex code snippets or not. It is possible that the model is performing for just simple code snippets well and performing even worse on more complex code snippets as compared to the statistical and Moses approach.

In my project work with my team, we want to integrate the text description of the code snippets as well as do an ablation study with respect to the complexity of the code snippet. We will be using separate encoders for text and code snippets and see for example which type of encoder is preferred to generate an appropriate question title. We also try to integrate more sophisticated encoders like BERT, and RoBERTa (after fine-tuning on code snippets and descriptions separately)using hugging face and see how much improvement we can achieve on the same dataset.

In the following literature, [7] used a machine learning method to conduct the automatic generation of Chinese short product titles for mobile display. They proposed a novel feature-enriched network model to solve this short title extraction problem. They started with creating a dataset of human-annotated short titles and their original long titles, including 6,481,623 pairs of original and short product titles. They define the input as a sequence of Chines/English words and the output as binary labels, 0 or 1. If a word is labeled 0, it means that this word should not exist in the shortened title. If a word is labeled 1, this word should be included in the shortened title. Recurrent Neural Network (RNN) was used as the main building block of the sequential classifier. What was different from the traditional RNN was that they divided features into 3 parts including Content, Attention, and Semantic feature.

Firstly, to get the Content feature, word embeddings were fed into a bidirectional LSTM network. Then they got the representation of the product title by concatenating the forward hidden state of each word with the corresponding backward hidden state and then calculated the content feature. Secondly, to get the Attention feature, they calculated a relevance score between the hidden vector of words and the representation of the entire title sequence. Thirdly, the semantic feature consisted of TF-IDF and NER Tagging. Finally, all the features were combined in an ensemble using a sigmoid function to get binary output. In the experiments, they trained the model on 500,000 product titles and tested it against another 500,000 title data. ROUGE was used as the evaluating metric. The accuracy was at a reasonable level.

The major difficulty was to decide what words in the long title should be kept when generating the short title considering the strict constraint of the number of words. The neural model they proposed performed well in this task. But I think other approaches, such as topic modeling based on Bayesian probability may also work in this case. Instead of outputting 0 or 1, we can also calculate the probability of each word in the original long titles being kept in shortened titles.

Among recent studies, [8] took significant advantage of neural networks. More specifically, they used an encoder-decoder architecture where the encoder is Transformer-based and the decoder is based on an autoregressive technique. To compare their approach, first, they select the state-of-the-art Post Title generation approach Code2Que as the first baseline. As the remaining baselines, they select five classical approaches from the source code summarization and text summarization domains (i.e., BM25, NMT, HybridDeepCom, Transformer, and BART) and apply them in the same

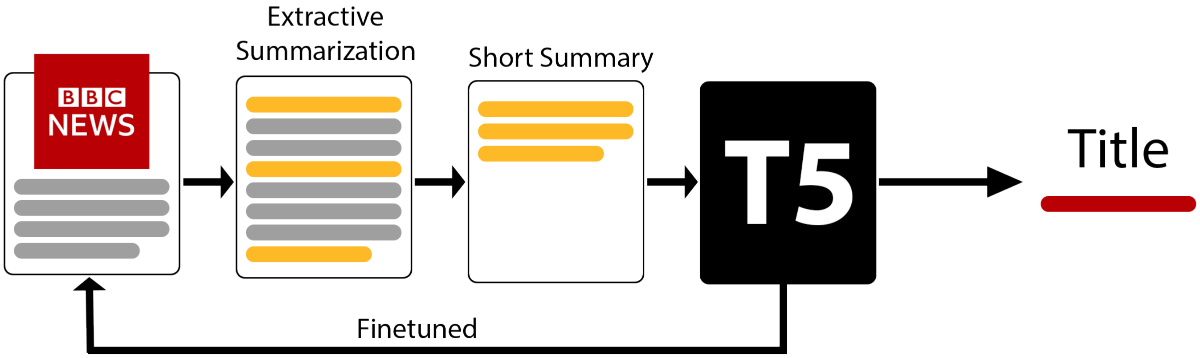

Fig. 1: An overview of our summarization technique. First, we extract short summaries from the input news using a combination of extractive summarization techniques. Then, having finetuned T5 on our news corpus, we generate abstractive summaries, also known as titles, from previously-generated short summaries.

fashion as Code2Que. [8] model consists of three phases: the construction of the corpus, the construction of the model, and the application of the model). The first step the authors take in constructing the corpus is to develop three heuristic rules to collect high-quality question posts from StackOverflow as part of the corpus construction phase. Then, per each question, they extract the code snippet, the problem description, and the title as a triplet from each post. The model is constructed by simply concatenating the code snippet with the problem description. To eliminate the OOV problem, the authors use the SentencePiece method to split these two modalities by using the SentencePiece algorithm. The next step is to formalize question post title generation for each programming language as separate but related tasks and employ multi-task learning to complete them. Finally, our model is fine-tuned based on a pre-trained Transformer model T5 that is mostly used for generating purposes. The code snippet and problem description for each new question post are entered into the constructed model during the model application phase. Using the beam search algorithm, the trained model will be able to automatically generate the corresponding post title through the use of the trained model.

3. METHOD

Our title generation technique is demonstrated in Figure 1. First, we develop a short summary extraction method which combines a set of extractive summarization techniques and outputs the most-agreed-on set of sentences as a short summary of the input news. Using the news corpus that we have, we finetune T5 on our corpus to let it update its weights based on the domain that we will apply it to. Then, after extracting the short summaries, we feed them to a finetuned T5 and ask it to generate the final title. Evidently, T5 can be trained to perform text-to-text processing. In our case, we train it on body-title pairs of samples. This enables us to leverage a pretrained transformer, finetune it on the news, and generate titles given news body. We used PyTorch's embedding to learn how to represent present words in the news body in a 50-dimensional space. Since we have utilized an encoder-decoder architecture to generate titles given the article body, we should use the proper output representation to make learning possible for the network. Since we aim to generate a series of words, we set the decoder model to make a multi-class prediction among the vocabulary at each time step. This makes our models' output to be a sequence of multi-class classifications.

4. EXPERIMENTS

Dataset: We used the BBC News Summary Dataset [1] in our experiments. This dataset was created using a dataset used for data categorization that consists of 2,225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005 used in [9]; whose all rights, including copyright, in the content of the original articles are owned by the BBC. The BBC dataset contains news articles where each consists of a title, body, and news category. Since there is no official data split mentioned in the data, we will use 80:20 train-test split with stratify as True. The train test data size will be Train Data document are 2,517 We will test our model on 445 documents. .

Ablation Studies: The 512 tokens limitation of T5 made us wonder if the entire body of the news is worthy to be fed into T5 to do the summarization. Previous studies also address this issue using a variety of techniques. Most of them used sliding windows and others leveraged other techniques as well. In this project, we perform another technique, which is to train different models using different portions of text to see which

[1] http://mlg.ucd.ie/datasets/bbc.html

Text Portion	ROUGE-1			ROUGE-2			ROUGE-L		
	r	p	f	r	p	f	r	p	f
All text	0.3582	0.3616	0.3560	0.1220	0.1230	0.1211	0.3405	0.3442	0.3386
First paragraph only	0.3337	0.3372	0.3308	0.0989	0.1009	0.0984	0.3166	0.3208	0.3142

portion is most informative and efficient to be used for title generation. Meaning which part of the text is enough to be used to generate a title. This approach aids us significantly since we have scarce data and limited computational power.

5. CONCLUSION

In this project, we worked on a two-phased text summarization technique for generating informative and fluent news titles. We used extractive summarization techniques and leveraged text-to-text transformers for title generation. Our final results showed that in the news domain, the first paragraph of the news satisfies the input in terms of informativeness and can solely be used to generate title. Our final results indicated that using the aforementioned two-phased title generation technique, we can gain improvement in analogy to the baseline.

6. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019.

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

[4] Michael Witbrock and Vibhu Mittal, “Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries (poster abstract).,” 01 1999, pp. 315–316.

[5] Rong Jin and Alexander Hauptmann, “A new probabilistic model for title generation,” in *COLING*, 2002.

[6] Zhipeng Gao, Xin Xia, John Grundy, David Lo, and Yuan-Fang Li, “Generating question titles for stack overflow from mined code snippets,” *ACM Transactions on Software Engineering and Methodology*, vol. 29, no. 4, pp. 1–37, oct 2020.

[7] Yu Gong, Xusheng Luo, Kenny Q. Zhu, Wenwu Ou, Zhao Li, and Lu Duan, “Automatic generation of chinese short product titles for mobile display,” 2018.

[8] Ke Liu, Guang Yang, Xiang Chen, and Chi Yu, “Sotitle: A transformer-based post title generation approach for stack overflow,” 02 2022.

[9] Derek Greene and Pádraig Cunningham, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, ICML ’06, p. 377–384, Association for Computing Machinery.