TLDR: Text Summarization from basic to advanced approaches

Nattapat Juthaprachakul
Simon Fraser University
301350117, njuthapr@sfu.ca

Siyu Wu
Simon Fraser University
301395909, swa246@sfu.ca

Abstract

In this paper, we explored several neural-based approaches to tackle the challenge of abstractive text summarization on news articles. The methods include vanilla sequence-to-sequence models without attention [1], BART transformer models with two different sizes [2], and T5 transformer models with three different sizes [3]. As these models have different architectures and complexity, we would like to evaluate and compare their performance using ROUGE metrics [4]. Through our experiments, we conclude that a LSTM sequence-to-sequence model lacks the ability to generalize with unseen data compared with fine-tuned and pre-trained transformers.

1 Introduction and Progress

1.1 Introduction

Text summarization is the task of producing a concise and fluent summary of texts or documents while preserving the key information and overall meaning. The goal of this project is to implement different text summarization techniques on the same dataset and compare their performance based on the same evaluation metrics such as ROUGE [4]. There are three main reasons why we choose this project.

First, thanks to the huge amount of available data, text summarization is becoming an important and useful task in several applications such as business analysis. For example, we always hear someone say, “I don’t want a full report, just give me a brief summary of the results.” With text summarization, we can gain the key information from articles and documents without reading through the whole article. This helps us save much time and effort.

Second, text summarization is a relatively new and very interesting topic in this course. We have gone through the list of final projects from the last term and term before last term. We found out that this topic has never been implemented before in this class.

Lastly, we now have an opportunity to implement several techniques we have learnt from this class with real-world dataset such as sequence-to-sequence and transformer models. Also, we have tried different techniques beyond scope of this course such as a text-to-text transfer network. We compare how these more advanced methods could improve the performance over the general approaches.

1.2 Progress

So far we have built up our training data and test data; implemented a vanilla sequence-to-sequence model [1], two types of BART models [2], and three types of Text-to-Text Transfer Transformer (T5) models [3]; and, evaluated their performance based on recall-oriented metrics called ROUGE [4]. We have gained some important findings regarding the prediction performance of different text summarization approaches.

2 Related Work

Text summarization is the process of creating a summary of a certain document that contains the most important information of the original one [10]. There are two main strategies for text summarization namely summarization by extraction, which consists of concatenating source sentences into a summary, and summarization by abstraction, which involves generating novel sentences for the summary [11].

For the extractive summarization, the earliest paper is in the 1950s [12]. The paper proposed a simple approach to count the frequency of words in documents. Words that occur often are likely to be the main topic of the document. However, this approach does not account for words in different contexts.

For neural network techniques, they are generally either sentence-extractive (choosing a lot of sentences as a summary) or abstractive (creating a summary from a sequence-to-sequence model).
In this paper, we focus on the abstractive summarization. Though neural network-based models work very well in several NLP tasks such as NMT, the abstractive summarization remains a major challenge, especially for a sequence-to-sequence model.

Nonetheless, the recent advances in neural network-based models with different architectures allow us to work with abstractive summarization more efficiently. For example, Pointer-generator networks [13] are introduced to deal with rare or out-of-vocabulary words, as well as repetitive words. An encoder-decoder model using LSTMs with the augmentation of hierarchical encoders and hierarchical attention can learn word and sentence level attention [14]. A paper proposed by [15] uses transformer-based model to do abstractive summarization of a very long sentence.

3 Approach

In the summarization task, given a news article, we would like to generate a shorter version of the story or a highlight while preserving the important points of the articles. More formally, for an input sequence $x = \{x_1, x_x, \dots., x_N\}$ of N words, we would like to generate $y = \{y_1, y_x, \dots., y_M\}$ such that $M < N$ while y preserves the essence of x. The words in y comes from the same vocabulary used in x and may not occur in the original text.
We have tried an extractive approach such as a frequency-based technique, but we find that the results are not suitable for comparison with the neural-based approaches. In addition, CNN/Daily Mail dataset [5] is commonly used for the abstractive summarization.

3.1 Model 1: LSTM sequence-to-sequence model

Our baseline model is a vanilla LSTM model without attention [1] and is a general Deep learning-based architecture used in NLP sequence-to-sequence tasks. Generally, we use LSTM with an encoder-decoder architecture inspired by Neural Machine Translation and summaries are generated from the decoder, using target vocabulary. This model provides us a baseline of the training time and accuracy; thus, it helps us better understand how much the improvement of other Deep learning-based approaches could offer regarding both accuracy and efficiency.

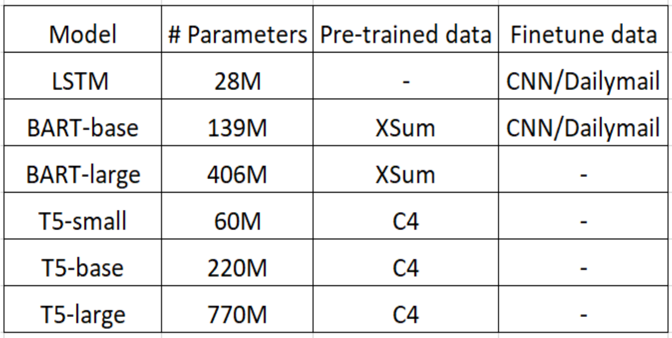
3.2 Model 2: Fine-tuned transformer

For our second model, we finetune the BART-base model on our pre-processed dataset. BART-base is a pre-trained transformer model by Facebook trained on XSum news dataset [6] with 139 million parameters [2]. BART uses a standard sequence-to-sequence (NMT architecture) with a bidirectional encoder (similar to BERT [7]) and a left-to-right decoder (similar to GPT [8]).

3.3 Model 3-6: Pre-trained transformers without fine tuning

BART-large is similar to BART-base but bigger in term of parameter numbers. T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks [3]. Each task is converted into a text-to-text format. It is invented by Google and pre-trained on C4 dataset, which is a cleaned version of Common Crawl's web crawl corpus.

For these four pre-trained models, we obtain them from Huggingface [9] and directly use them without any finetuning. These models have different numbers of parameters and pre-trained data sets as shown in the table 1.



Model	# Parameters	Pre-trained data	Finetune data
LSTM	28M	-	CNN/Dailymail
BART-base	139M	XSum	CNN/Dailymail
BART-large	406M	XSum	-
T5-small	60M	C4	-
T5-base	220M	C4	-
T5-large	770M	C4	-

Table 1: Model settings

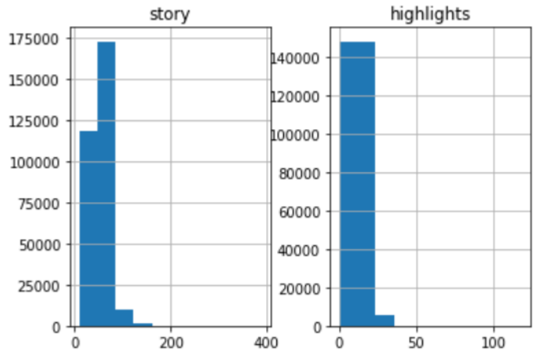
4 Evaluation

4.1 Dataset

We use raw data without data anonymization of the CNN/Daily Mail dataset [5], which consists of online news articles (or stories) paired with multiple summaries (or highlights). The first feature is a text of news articles which is used as the documents to be summarized while the second feature is the joined text of highlights which is the target text summarization.

The articles have an average of 781 tokens while the summaries have an average of 56 tokens. For pre-processing, we remove punctuation, number, and CNN and Daily Mail name tags at the beginning of every line. We also lower every letter in all words and remove noisy words that are not related to news articles such as advertisements.

After all pre-processing steps, we have total data of 305,758. We split them into train/validation/test sets with a fraction of 90/5/5. To save training and decoding time, we use only the first sentence of stories and two summaries. The pre-processed summaries and stories now have around 12 and 53 words on average respectively.



4.2 Evaluation Metric

The widely used general evaluation metric for text summarization is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) which automatically determines the quality of a summary by comparing the output texts produced by algorithms and the reference summaries [4]. There are several variations of ROUGE that we will use as follow:

ROUGE-n: it is a recall-based metric that is based on comparison of n-grams between reference summaries and candidate summaries.

ROUGE-L: it uses the concept of the longest common subsequence (LCS) between the two sequences of texts. The intuition is that the longer the LCS between two summary sequences, the more similar they are.

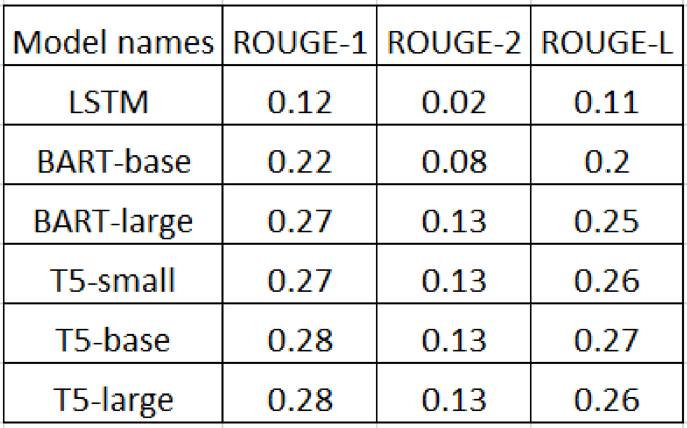
In this paper, we use F1 from the ROUGE metric and select only three specific ROUGE metrics namely: ROUGE-1, ROUGE-2, and ROUGE-L.

4.3 Experiment Detail

We have implemented one sequence-to-sequence model, pretuned one transformer-based model (BART-based [2]), and used 4 different pre-trained transformers (BART-large [2], T5-small, T5-base, and T5-large [3]) from Huggingface [9]. Later, we use all these models to predict the summaries and evaluated them with ROUGE scores as shown in the table 1 and 2.

We use Google Colab as our main training facilities. We have tried CPU, GPU (Tesla T4), and TPU. For training of our baseline model, it takes around 7, 2, and 0.5 hours respectively while it takes around 1 hour for pre-tuning the BART-base model [2] with our dataset. For testing, we use GPU for decoding which takes around 1-1.5 seconds per sample. Since we have 10,000 samples in the test set, it takes around 2-3 hours for this process per model.

4.4 Experiment Result



Model names	ROUGE-1	ROUGE-2	ROUGE-L
LSTM	0.12	0.02	0.11
BART-base	0.22	0.08	0.2
BART-large	0.27	0.13	0.25
T5-small	0.27	0.13	0.26
T5-base	0.28	0.13	0.27
T5-large	0.28	0.13	0.26

Table 2: Experiment results

5 Discussion

Reference summary	-manchester united have made bastian schweinsteiger their top summer target. -louis van gaal eyes bastian schweinsteiger reunion at old trafford.
LSTM seq2seq	-manchester united have been linked with a move to manchester united.
BART-base	-manchester united have made bastian schweinsteiger their top summer target.
BART-large	-bastian schweinsteiger is thought to be keen on joining his former boss louis van gaal at old trafford.
T5-small	-manchester united have made bastian schweinsteiger their top summer target. -the schweinsteiger is thought to be keen.
T5-base	-manchester united have made bastian schweinsteiger their top summer target. -the schweinsteiger is thought to be keen.
T5-large	-manchester united have made bastian schweinsteiger their top summer target. -schweinsteiger is thought to be keen on.

6 Conclusion

-

7 Future Work

We have already finished our implement with different text summarization methods and the evaluation part. For the rest of time, we are going to focus on our final report, poster, video, and Powerpoint. We would like to include everything we have done and every detail about our experiment in our final report. Also, we would like to make a concise and attractive Powerpoint showing our work to the teacher and our classmates to make sure that everyone could understand our experiments and our findings.

References

[1] Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to Sequence Learning with Neural networks. In Advances in Neural Information Processing (NIPS), Montreal, Canada.

[2] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, & Luke Zettlemoyer. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

[3] Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text TransformerCoRR, abs/1910.10683.

[4] Lin, C.Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out (pp. 74–81). Association for Computational Linguistics.

[5] R. N. et al., “Cnn/dailymail dataset,” CoNLL, 2016

[6] Shashi Narayan, Shay B. Cohen, & Mirella Lapata (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme SummarizationArXiv, abs/1808.08745.

[7] Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language UnderstandingCoRR, abs/1810.04805.

[8] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. OpenAI. 2018.

[9] Thomas Wolf and Lysandre Debut and Victor Sanh and Julien Chaumond and Clement Delangue and Anthony Moi and Pierric Cistac and Tim Rault and Rémi Louf and Morgan Funtowicz and Jamie Brew (2019). HuggingFace's Transformers: State-of-the-art Natural Language ProcessingCoRR, abs/1910.03771.

[10] X. Carreras, L. Màrquez, Introduction to the CoNLL-2004 shared task: Semantic role label-ing, in Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004(2004), pp. 89–97

[11] U.Hahn and I.Mani 2000. The challenges of automatic summarization. IEEE Computer, 33(11): 29-36

[12] Luhn, H. P. (1958) The automatic creation of literature abstracts, IBM Journal of Research and Development, vol. 2, no. 2.

[13] C. M. A. See, P. Liu, “Get to the point: Summarization with pointer-generator networks,” CoRR, 2017.

[14] Nallapati, R., Zhou, B., Santos, C. D., Gulcehre, C., and Xiang, B. (2016). Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. In CoRR, arXiv:1602.06023.

[15] P. L. et al., “Generating wikipedia by generating long sequences,” ICRL, 2018.