Mini-project report – Text summarization

Malo Adler, Quentin Ravaux

1 Introduction

Text summarization, the task of generating concise and meaningful summaries from large texts, has gained significant importance in the digital age. In opposition to extractive summarization, where a few sentences from the original text are extracted and concatenated to form a summary, abstractive summarization requires writing a summary from scratch, with words and sentences that do not necessarily appear in the original text. This abstraction requires a deeper understanding of the linguistic patterns and meanings of the original texts, making abstractive summarization naturally more challenging than extractive summarization. Transformer architectures, renowned for their ability to capture complex linguistic patterns, have revolutionized natural language processing, including text summarization. In this project, we explore the use of transformers for abstractive summarization and employ fine-tuning techniques to enhance their performance. To fine-tune our model, we use the dataset 'Amazon Fine Food reviews' [1], composed of reviews of food products, along with a hand-written summary.

2 Background

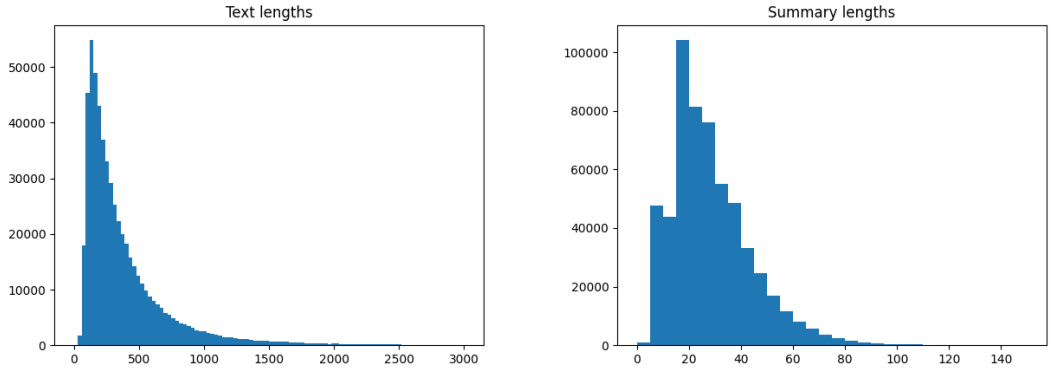
Given the recent success of Transformer models, we decided to focus on these architectures rather than more classical RNNs, which usually give poorer results. First, the Generative Pre-trained Transformer (GPT-2) model [3], developed by OpenAI, is known to yield satisfying results for a variety of NLP tasks. However, it was trained by learning to predict the next word, given all the previous words of a document. It is therefore not necessarily the most relevant training for a summarization task, because both tasks are quite different. Although fine-tuning the GPT model on a summarization task can be efficient, it requires a lot of computation, which decided us to look for another, more relevant model.

This led us to the Text-to-Text Transfer Transformer (T5) model [4], developed by Google Research. It is based on a "text-to-text" framework: it has been trained on a diverse range of tasks by converting them into a text generation problem. For text summarization, the T5 model can be trained by providing pairs of source text and target summaries, where the task is to generate a concise and coherent summary given the source text. It has been shown that the T5 model achieves state-of-the-art performance on various abstractive text summarization benchmarks, which is why we used it and fine-tuned it on our dataset.

3 Training data

We used the dataset 'Amazon Fine Food reviews' to fine-tune our model. Each element of this dataset is composed of various features (product ID, user ID etc.), most of which we will not need. From this dataset, we extracted the only two features that we were interested in: the original reviews, and their summary. The dataset contains 568 454 rows, but we decided to use only about 10% of this data to train our models, for computational reasons. The lengths of the texts and their summaries are represented with histograms

on Figure 1. Visualizing these histograms allowed us to set a limit length for the texts and the summaries, making the training more efficient both in time and storage. We set these limits to 500 words for the texts, and 40 words for the summaries. The texts and summaries shorter than this limit will therefore be padded to reach this maximum length, while those longer than this limit will be truncated to fit in this limit.



(a) Text lengths

(b) Summary lengths

Figure 1: Histograms of the length of the texts and their summaries

4 Testing and validation

We split our dataset in three distinct subsets: the training set (80% of the dataset), the validation set (10% of the dataset) and the test set (10% of the dataset). We used the validation set to see the improvements of the model during its training (which was made with the training set). Once we had a satisfying model, we computed the final performance of the model using the test set.

To evaluate the models, we used the ROUGE metric. More specifically, we used the three following metrics:

- **ROUGE-1**: It counts the rate of overlapping unigrams between the original and the predicted summary.
- **ROUGE-2**: It counts the rate of overlapping bigrams between the original and the predicted summary.
- **ROUGE-L**: It counts the normalized length of the longest sub-sequence that is present in both the original and the predicted summary.

For each of these metrics, we kept the F1-score as an evaluation metric, to try and have a good precision-recall balance. We used the function `load_metric("rouge")` from the Python module `datasets` [2] to compute these metrics.

	ROUGE-1	ROUGE-2	ROUGE-L
t5-small model	11.12	2.99	10.31
After 1 epochs	16.53	4.97	16.24
After 2 epochs	17.71	6.09	17.38
After 3 epochs	17.80	6.23	17.47
After 4 epochs	19.29	6.93	19.00
After 5 epochs	18.87	6.96	18.50
After 6 epochs	18.66	6.75	18.25

Figure 2: Evaluation results for the t5-small model, and for our fine-tuned model at different stages of training

	ROUGE-1	ROUGE-2	ROUGE-L
GPT-2 model	0.64	0	0.64
After 1 epochs	5.96	0	5.95

Figure 3: Evaluation results for the GPT-2 model, and for our fine-tuned model at different stages of training

5 Results

We have fine-tuned the GPT-2 model on 1 epochs, and the t5-small model on 6 epochs, with a learning rate of 10^{-4}, a weight decay of 0.01, and a batch size of 4. We have gathered in Figure 2 and 3 the evaluation results of the base models, and of our fine-tuned models at different stages of training.

For the fine-tuned GPT-2 model, we observe that the original results are really bad, and they do not improve a lot after training. Therefore, and because we had computational limitations, we decided to focus on the T5 model training, which seemed more promising.

For the fine-tuned T5 model, we can see that the results are fairly good, and that the fine-tuning really helps increasing the summarization capacity of the model. However, after several epochs, the validation score starts to slowly decrease. Although it is not represented on this figure, the training score keeps increasing: the model starts to overfit the training data. For our training parameters, and for the dataset we used, it seems that 4 epochs are therefore optimal for creating a good model. The ROUGE score are however quite far from state-of-the-art performances, probably because of the corpus we used to fine-tune our model.

Once the best models have been selected, we have finally tested them on the test dataset. These results are shown on Figure 4.

	ROUGE-1	ROUGE-2	ROUGE-L
T5 model	19.12	6.85	18.88
GPT-2 model	6.04	0	6.06

Figure 4: Final results on the test dataset for the GPT-2 fine-tuned model, and for the T5 fine-tuned model

Original summary	Predicted summary
Best sour cream and onion chip I've had	Delicious and a little too salty
So Delicious...Yet my companions wont touch them.	The best chips ever
So much flavor your farts will smell like sweet onions	Awesome
Do not taste from bottle! Mix with vanilla for true flavor.	YUCK!

Figure 5: Examples of summaries predicted by our T5 fine-tuned model

To see how the model behaves on concrete examples at the end of the fine-tuning, we generated some summaries for given reviews, and compared them to the original summaries. These examples are presented on Figure 5.

6 Discussion

A first limitation of our project comes from its evaluation: the ROUGE score is the most commonly used metric for text summarization tasks, and it is relevant in some way. However, comparing the exact words of the predictions and those of the original summaries has its limits: two texts can mean the exact same thing but express it in two different ways (different formulations, synonyms etc.). The relatively low ROUGE scores we obtained are therefore not very problematic in themselves: they should not be our only way of evaluation. When we look at the results provided by our model (see Figure 5 for instance), the overall idea is conveyed, even though it is not always perceived by the ROUGE metric.

Secondly, we have fine-tuned the models on a small amount of training data (50 000 examples) which limits the performances of the models. Moreover, we have taken the small version of GPT-2 and T5. It could be interesting to fine-tune bigger models with all the training data available to see if the results are better and the prediction more satisfying.

7 Conclusion

In this project, our goal was to use pre-trained Transformer models, and fine-tune them for abstractive text summarization purposes. For this, we compared the GPT-2 model, which has been trained on a next word prediction task, and the T5 model, which has

been trained on multiple text generation tasks. After fine-tuning these on the 'Amazon Fine Food reviews' dataset, we evaluated their performances mainly using the ROUGE metric, and to a certain extent, a qualitative by-hand analysis. The results yielded by the T5-model are more satisfying, and given the training circumstances (training dataset and model size, computational capacities), are great. However, for a real-world use of this model, we would need more training data and more computational power, to create a really efficient abstractive summarization tool.

References

[1] Amazon Fine Food Reviews.

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

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[4] 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020.