Team 26: Text summarization

April 24, 2021

Abstract

In our project, we have decided to create text summarizer with BERT. There is already a Bert Extractive Summarizer -package. It's based on following article: https://arxiv.org/ftp/arxiv/papers/1906/1906.04165.pdf This package is intended for lecture-summarization, and our goal is to extend and fine-tune this model for news or scientific articles. Even though foundational work is already done for the package, there's still much to customize such as tokenizer and model. Our target language is english. Summarization tools such as Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) are available, and part of the project is to try to evaluate the texts automatically and manually.

1 Introduction

Automatic summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content. There are two main approaches to automatic summarization (independently of the application domain, e.g. text, images, video etc.):

- Extraction-based or extractive summarization
- Abstraction-based or abstractive summarization

When it comes to text documents, summarization is closely related to data compression and information understanding. The ability to produce coherent, well-structured summaries has the potential to transform efficiently the way that discovery systems work, as well as help human readers in skimming large datasets of text documents. That is why automatic summarization is considered one of the most important, yet least solved, tasks in NLP and a method that will transform the way people consume information on the Internet. In conclusion, applying text summarization reduces reading time, accelerates information retrieval and increases the amount of useful, dense information. In our case, we will deal with extractive summarization where a system produces summaries by choosing a subset of the initial text.

2 Background

The first summarization techniques go back already more than 50 years to Luhn's and Edmundson's seminal papers on automatic summarization (1958 and 1969 respectively, [3], [1]). Early work in the field dealt with single document summarization (news story, scientific articles etc.) Later, multi document summarization was applied in big data clusters to provide a coherent and brief digest to the users.

3 Evaluation and results

Evaluation was done on newsroom dataset [2]. The model inference however is extremely slow, and only a subset of samples was chosen to be evaluated. The sample size was chosen to be 150, which resulted in tolerable running times (¡ 15 minutes).
Two different BERT models were tested, BERT$_{LARGE}$ and BERT$_{BASE}$. These models were compared against baseline model Lede-3 and GPT-2. GPT-2 was chosen to be compared against BERT models, because in the original study [4] they noticed that BERT should be generating more representative embeddings of the sentences. Also GPT-2$_{LARGE}$ was chosen as one model to be evaluated. GPT-2$_{LARGE}$ has 774 million parameters, which is roughly the double

of parameters in BERT$_{LARGE}$ and GPT-2$_{MEDIUM}$. Also GPT-2-XL and new GPT-NEO (GPT-3 replication) would've been interesting comparisons, but unfortunately they were too large to fit on GPU.

Tthe python package bert-extractive-summarizer [4] uses only sentences from original text, and the amount of sentences is defined by either fixed ratio, or amount of sentences. In this case the ratio was defaulted to 0.2, which means that we use 20% of the sentences from original text. We can calculate empirical distribution of the ratios seen in reference summaries as displayed in Figure 1.

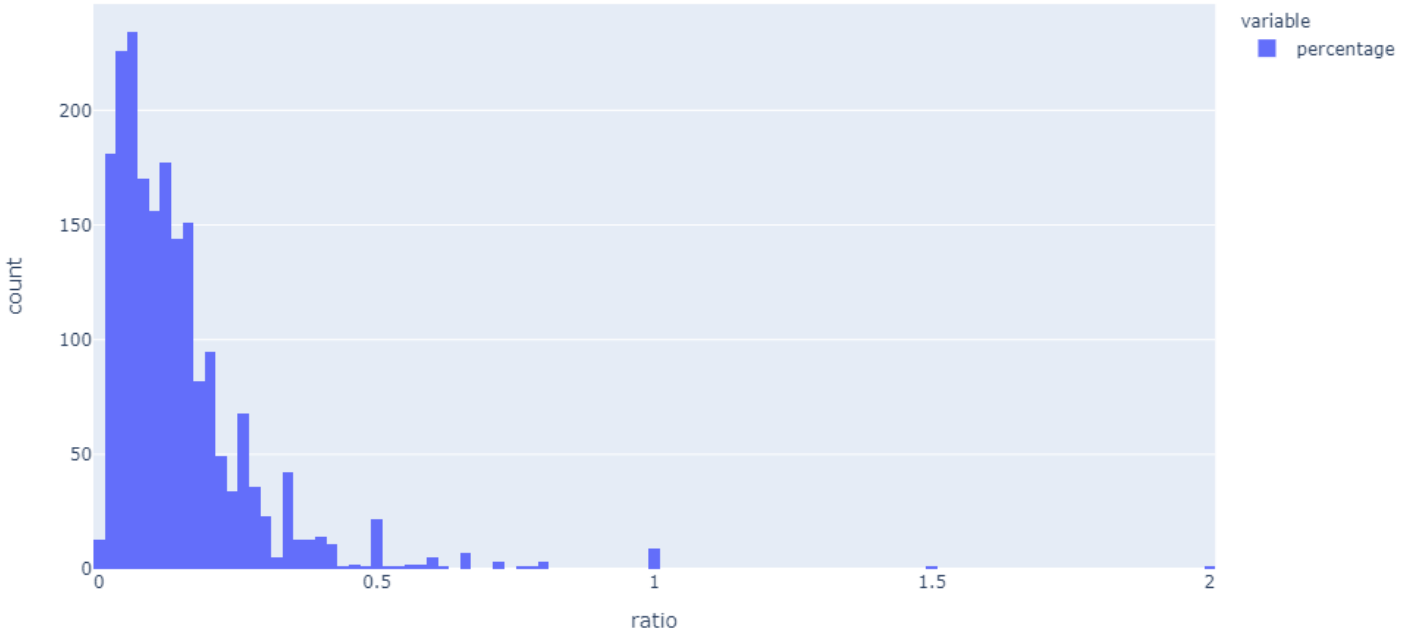

Figure 1: Distribution of observed ratios in the data

Here we can see that the mean is around 0.15. We can also plot what length of sentences there are as in following Figure 2.

Here we actually see, that there are multiple summaries which have sentence length 1. This is important to remember because it might favor Lede-3 classifier due to shortness of text.

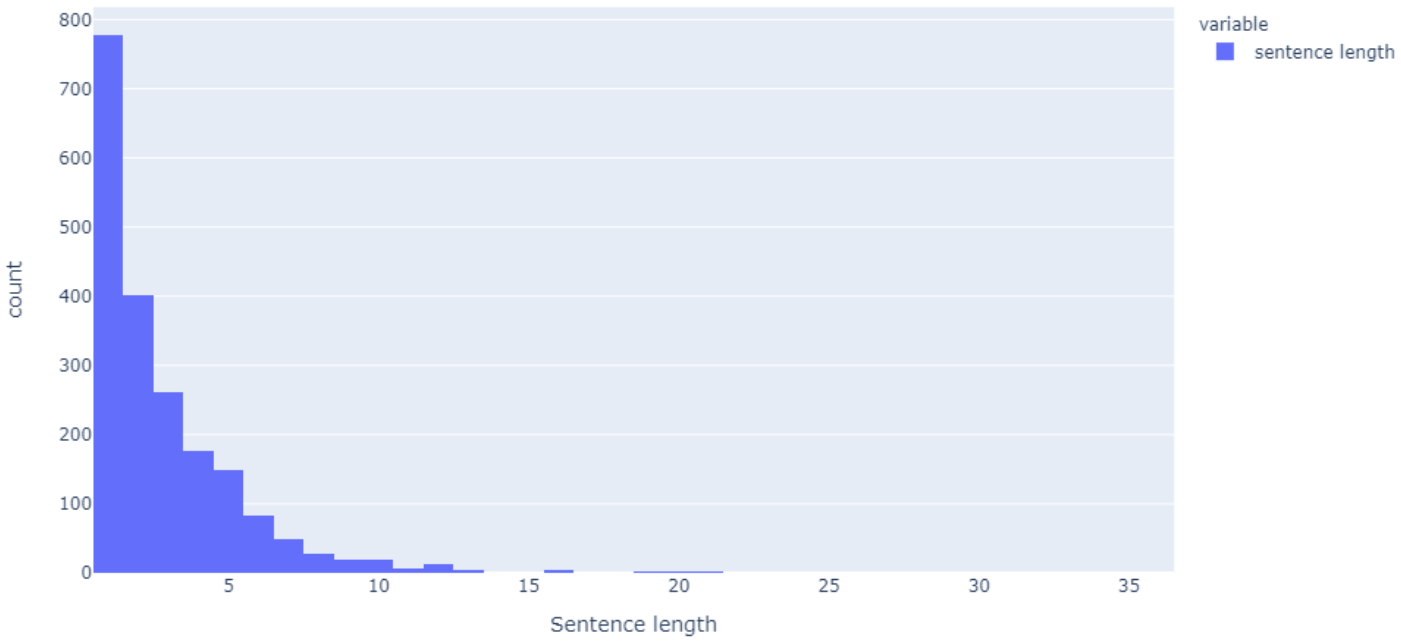

Figure 2: Distribution of observed ratios in the data

Rouge-1

Model name	f1-score	precision	recall
BERT-large	41.19	35.23	64.36
BERT-base	41.14	35.58	63.2
GPT2-medium	42.53	36.36	66.02
GPT2-large	41.17	35.25	64.49
Lede-3	55.73	52.74	74.42

Heres image

Rouge-2

Model name	f1-score	precision	recall
BERT-large	32.37	27.81	50.19
BERT-base	32.05	28.2	48.42
GPT2-medium	33.56	28.88	52.37
GPT2-large	32.16	27.74	50.4
Lede-3	50.63	48.04	66.88

Rouge-L

Model name	f1-score	precision	recall
BERT-large	41.77	35.64	61.76
BERT-base	41.29	35.68	59.93
GPT2-medium	43.06	36.83	63.33
GPT2-large	41.73	35.59	61.94
Lede-3	57.52	53.79	74.16

References

[1] H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2): 264–285, Apr. 1969. ISSN 0004-5411. doi: 10.1145/321510.321519. URL `https://doi.org/10.1145/321510.321519`.

[2] M. Grusky, M. Naaman, and Y. Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL `http://aclweb.org/anthology/N18-1065`.

[3] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2), 1958. URL `http://www.research.ibm.com/journal/rd/022/luhn.pdf`.

[4] D. Miller. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165, 2019. URL `http://arxiv.org/abs/1906.04165`.