Team 26: Text summarization

March 11, 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, [7], [3]). 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.

A typical flow of an abstractive summarization system [5] consists of:

- Interpretation to obtain an intermediate representation
- Transformation into a summary representation (sentence-level ranking)
- Generation based on semantically important sentences

A significant part of the intermediate representation is finding the most representative words of the text. The most common topic representation approaches [1] include:

- Topic words
- Frequency-based approaches (TF-IDF, word probability)

- Latent Semantic Analysis (LSA)
- Bayesian Topic Models (LDA)

Lately, neural modeling and huge pre-trained models have dominated the field [9].

3 BERT

In many NLP tasks, the shortage of training data is often the source of the problems. There has been some research to tackle this issue by using unannotated text on the internet for training language representation models that serve general purpose. This process is also known as pre-training. The idea is that the pre-trained model can be fine-tuned later on small-data NLP tasks like question answering and sentiment analysis to have significant accuracy improvements compared to training on these datasets from scratch.

BERT, also known as Bidirectional Encoder Representations from Transformers, is an example of pre-trained language representation models which uses a combination of masked language modeling objective and next sentence prediction on a large plain text corpus including Wikipedia and the Toronto Book Corpus. Unlike other language representation models, BERT is designed bidirectional in the way that it reads the input sequences. This characteristic allows the model to learn the context of a word from both left and right sides, capturing the previous and next contexts. BERT can also be fine tuned with one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. [2]

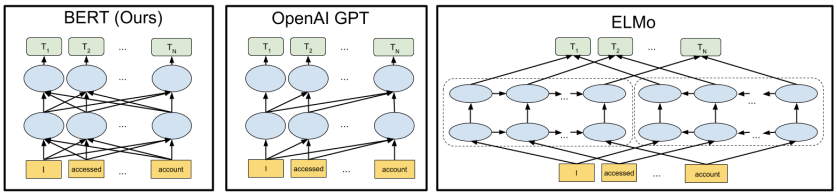

Figure 1: Architectures of known pre-training methods

In the above figure created by Google AI, the neural network architectures state-of-the-art contextual pre-training methods are shown. The arrows show the information flow through layers while the green boxes indicate the final contextualized representation for input words. It can be seen that BERT is deeply bidirectional while OpenAI's GPT is unidirectional and ELMo has a shallow bidirectional structure.

BERT can also be considered conceptually simple. As mentioned earlier, BERT uses the advantage of bidirectionality by using a masked language modeling objective. Some words in the input text are masked out and each word

are conditioned bidirectionally to predict the those masked words. It also uses next sentence prediction in which the model predicts if a sentence logically follows another sentence which was given previously. This task enables BERT to capture the relationship between sentences. [2]

BERT has also been used in previous work for text summarization for news. In his paper, Yang suggests a fine tuned variant of BERT called BERTSUM for extractive summarization in which the input sequence and embeddings of BERT are modified to make it possible for extracting summaries. It is concluded that BERTSUM with inter-sentence Transformer layers achieve the best performance for extractive summarization of news, using CNN/DailyMail news and New York Times Annotated Corpus. [6]

4 The dataset

The Cornell University NEWSROOM dataset [4] consists of 1.3 million news articles and summaries between the years 1998 and 2017. It contains articles from 38 major publications and the summarization strategies combine both extraction and abstraction. The Newsroom website (summari.es) contains various different tools for analyzing the large article database. The dataset is written by humans with purpose of summarization. Dataset was created by scraping the website for content and using summaries in the HTML metadata. These metadata summaries were created to be found from search engines and social medias. Top news websites gathered from Alexa.com were used.

We will be filtering the dataset [4] to use extractive summaries, as our model is extractive. Another option for training dataset would be the New York Times corpus, which contains over 650 000 manually summarized news articles. These summarizations were done afterwards by library scientists. [8]

5 State of art

Often, for the baseline precision Lede-3 is used. It's automatic summarization strategy that copies first n words of the first sentence and uses it as summary. In our work we will be using this as a baseline also. Current state of art models in text summarization can be classified to fully extractive, fully abstractive or mixed models. In the study [4] ROUGE-1, ROUGE-2 and ROUGE-L F_1 score variants were used to account for different summary lengths. In our work we will also be using ROUGE and BLEU scores for testing. We can test on both datasets [4] [8]

In the leaderboard it seems that mixed models perform the best, then comes the extractive and abstractive is the least efficient. TextRank was one of the best models in the study [4] and it will also be used as comparison for our model. The very least goal is to achieve better results than the Lede-3, but it will be

interesting to see how well the extractive BERT performs ranks compared to TextRank.

6 Project plan

We will be using Google Colabs resources for computing. It's convenient because we can store data in common cloud space, Google Drive, and Colab has a lot of computing power. We will be using BERT Extractive Summarizer - package for the actual modeling. The training data will be extractive data filtered from the NEWSROOM dataset [4]. For evaluation, we will be using ROUGE and BLEU scores with the NEWSROOM dataset, and possible also New York Times Corpus dataset [8]. We will be comparing our result with the previously mentioned baseline Lede-3, and possibly also to more advanced TextRank.

References

[1] M. Allahyari, S. A. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268, 2017. URL `http://arxiv.org/abs/1707.02268`.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

[3] 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`.

[4] 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`.

[5] K. S. Jones. Automatic summarising: factors and directions. *CoRR*, cmp-lg/9805011, 1998. URL `http://arxiv.org/abs/cmp-lg/9805011`.

[6] Y. Liu. Fine-tune bert for extractive summarization, 2019. URL `https://arxiv.org/abs/1903.10318`.

[7] 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`.

[8] E. Sandhaus. The new york times annotated corpus. In *The New York Times Annotated Corpus LDC2008T19*. Linguistic Data Consortium, 2008. URL https://doi.org/10.35111/77ba-9x74.

[9] M. Zhong, P. Liu, D. Wang, X. Qiu, and X. Huang. Searching for effective neural extractive summarization: What works and what's next. *CoRR*, abs/1907.03491, 2019. URL http://arxiv.org/abs/1907.03491.