Leveraging Pre-trained models for Lecture notes Summarization

Latifa Adhibi

Faculty of Informatics, ELTE Eötvös Loránd University
Budapest, Pázmány Péter stny. 1/C., 1117
sjrejv@inf.elte.hu

Abstract. With the emerging of e-learning, getting text representation of video lectures is becoming easier. Researches have worked on text summarization in different domains. In this project, we aim to leverage pre-trained models in order to summarize lectures. The main challenge is that the data used has no human-written reference summaries. To overcome the problem, we used a metric that does not require reference summaries. Results showed that BART model outperformed BERT and Pegasus.

Keywords: Text Summarization - Natural Language Processing - Deep Learning - Transformers

1 Introduction

Text summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content [1]. A lot of researches aimed to automate that task. Automatic text summarization is divided into two main approaches: abstractive and extractive.
Abstractive text summarization is similar to human summarization, it generates sentences using a vocabulary that is not limited to the specified text. In other words, the obtained sentences might or might not be present in the original text. While in extractive text summarization, generated summaries are limited to the content of the source text; it extracts the core and most relevant information and keeps their original structure.
Text summarization can be used in education to summarize the transcripts of video lectures. The main goal is to get the most valuable information. While getting datasets having text representation of lectures is becoming more feasible. It's difficult to evaluate the generated summaries due to the fact that these datasets have no reference summaries; evaluating the results using metrics like ROUGE and BLUE score can't be done since they require the presence of human-produced summaries. In this project, we will use pre-trained models to summaries lectures obtained from a kaggle data set [2]. In the evaluation phase, we will compare the results using a metric that does not require reference summaries.

2 Related Work

Extractive text summarization is the topic of many research papers, different techniques and comparative studies had been performed.

D. Krishnan et al. [3] proposed a supervised approach to implement extractive text summarization. They defined the problem as a binary classification where the goal is to predict if a sentence from the input document belongs to the summary or not. Five classifiers were tested; Naive Bayes, KNN (K- Nearest Neighbour), Random Forest, SMO (Sequential Minimal Optimization), J48 and Bagging. Before the modeling phase they extracted minimal discriminative features and handled class imbalance. The model achieved an average ROUGE-1 score of 0.51 across 5 domains on the BBC news summary dataset.

H. Gupta and M. Patel [4] used an LSTM based model. They scored vectors obtained from The Elmo Embedding using the cosine similarity measure. Based on these scores, sentences of the text document were ranked and the summary was generated by selecting the five sentences with the highest scores. Compared to TextRank algorithm, the model gave a higher average for the F-measure metric.

S. Nath and B. Roy [5] worked on generating release notes using a graph based ranking algorithm; TextRank. Due to the fact that TextRank ignores the semantic similarity, they integrated the GloVe word embedding to improve the keyword extraction process. The model is then compared to the Latent Semantic Analysis (LSA) algorithm. The results of their evaluation showed that the improved TextRank method outperforms LSA.
Since it's language independent, researches had also tested the TextRank algorithm on languages other than English; Telugu language for example[6].

D. Miller leveraged BERT for Extractive Text Summarization on lectures [7]. His motivation was to provide a RESTful service with configurable summary sizes for the lecture notes using an up to date approach. He used the BERT model for text embeddings and K-Means clustering to pick out sentences closest to the centroid for summary selection. Due to the fact that the created dataset has no golden truth summaries for lectures, only human comparison and quality of clusters were used for evaluation.
Fine-tune BERT for Extractive Summarization [8] is a paper in which BERTSUM model was presentd; it's a variant of BERT for extractive summarization. After obtaining the sentence vectors from BERT, different summarization-specific layers were added on top of the BERT outputs and jointly fine-tuned: Simple Classifier, Recurrent Neural Network (LSTM) and Inter-sentence Transformer. The results obtained on two large-scale datasets showed that BERTSUM with inter-sentence Transformer layers can achieve the best performance.
Due to its strong performance, BERT was leveraged in various research: Arabic Text Summarization Using AraBERT [9], SciBERTSUM: Extractive Summariza-

tion for Scientific Documents[10] and DP-BERT: Dynamic Programming BERT for Text Summarization [11] are exaplmes of its usage.

J. Chen and H. Zhuge [12] proposed a Text-Image extractive summarizer using Multi-Model RNN. They start by encoding documents and images with a multi-modal RNN, and then calculates the summary probability of sentences through a logistic classifier. Text coverage, text redundancy, and image set coverage were the features selected for the classifier. Their experiments showed that the proposed method outperforms the state-of-the-art neural summarization methods.

3 Dataset Description

AutoBlog [2] is a kaggle dataset that has files related to Medical Engineering and Pattern Recognition Lectures.
For Medical Engineering: audios, transcripts and videos are provided. While for Pattern Recognition, there is also PDF Slides, Youtube SBV and blog posts.
In this project, we worked on 13 transcripts from the Medical Engineering Lectures.

4 Experiments

In this section, we will test BART [13], Pegasus[14] and Bert Extractive Summarizer[15].

Both BART and Pegasus are abstractive summarizers and their implementation does not support long text so they're not able to directly summarize the long lectures transcripts. To overcome this problem, we chunked the input text by breaking it up in many smaller pieces. We fixed the chunk size, then summarized each of them separately. The final result is obtained by joining the summaries of each chunk.

We used Google Colab to run the models. The code is shared in a Github repository.[1]

4.1 BART

BART is a denoising autoencoder for pretraining sequence-to-sequence models [16]. This means that a fine-tuned BART model can take a text sequence (for example, English) as input and produce a different text sequence at the output (for example, French). This type of model is relevant for machine translation (translating text from one language to another), question-answering (producing answers for a given question on a specific corpus), text summarization (giving a summary of or paraphrasing a long text document), or sequence classification

[1] `https://github.com/Latifaa17/Lecture-notes-Summarization`

(categorizing input text sentences or tokens). Another task is sentence entailment which, given two or more sentences, evaluates whether the sentences are logical extensions or are logically related to a given statement.[17]
BART is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text.[16]
BART uses the standard sequence-to-sequence Transformer architecture as presented in Fig.1
The encoder part is similar to BERT and the decoder part is similar to GPT.

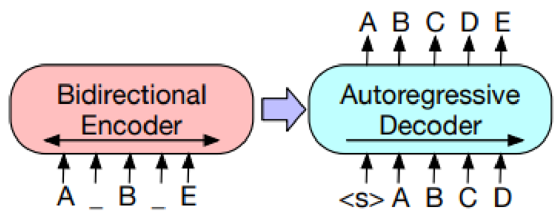

Fig. 1. BART architecture.

4.2 PEGASUS

Pegasus is a State-of-the-Art Model for Abstractive Text Summarization proposed by Google researchers [19].
PEGASUS stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization. It is a sequence-to-sequence model with gap-sentences generation as a pretraining objective tailored for abstractive text summarization. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. [20]
Fig. 2 shows the architecture as it appears in the original paper [20].

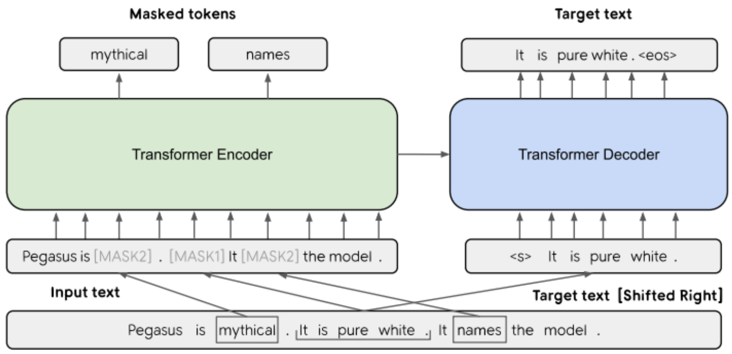

Fig. 2. PEGASUS architecture.

Both GSG and MLM are applied simultaneously to this example as pre-training objectives. Originally there are three sentences. One sentence is masked with [MASK1] and used as target generation text (GSG). The other two sentences remain in the input, but some tokens are randomly masked by [MASK2](MLM) [20].

4.3 BERT Extractive Summarizer

BERT Extractive Summarizer was presented in a paper by D. Miller [7]. He implemented a python-based RESTful service that uses the BERT model for text embeddings and K-Means clustering to determine sentences closest to the centroid for summary selection.
BERT goals are specific for pre-training. It has two major tasks:

- **Masked Language Modeling (MLM):** Mask 15% of the words in each sequence of the input tokens at random, and then predict those masked tokens.
- **Next Sentence Prediction (NSP):** BERT is given two sentences A and B, the model predicts whether the sentence B properly follows the input sentence A.

5 Results

In order to evaluate the models, we need a measure that does not require human-written reference summaries. We chose to use the BLANC measure. O. Vasilyev et al. presented this method that automatically estimates the quality of generated summaries. BLANC achieves this by measuring the performance boost gained by a pretrained language model with access to a document summary while carrying out its language understanding task on the document's text. [18]

Two versions were presented:

- **BLANC-help** uses the summary text by directly concatenating it to each document sentence during inference.
- **BLANC-tune** uses the summary text to finetune the language model, and then processes the entire document.

BLANC measure was compared to human evaluations and to the ROUGE family of summary quality measurements, the results showed that it has good correlation with them.

For our project, we used the BLANC-help version. The following table contains the results.

	BART	**BERT**	**Pegasus**
BLANC	0.163	0.154	0.102

It is shown that BART model outperformed BERT and Pegasus when tested on the long lectures transcripts.

6 Conclusion

During this project, we tested BART, BERT Extractive Summarizer and Pegasus for the text summarization task. Our target was to generate summaries for lectures' transcripts that we obtained from a kaggle dataset. These transcripts has no golden truth summaries, that's why we used BLANC metric to evaluate the models. This metric aims to evaluate automatically generated summaries without the need of human reference summaries. The BART model achieved the best results.
As future work, the number of transcripts used can be increased and other NLP models can be tested.

References

1. Automatic Text Summarization with Machine Learning — An overview `https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25`
2. AutoBlog Dataset. `https://www.kaggle.com/datasets/abnerh/autoblog`
3. D. Krishnan, P. Bharathy, Anagha and M. Venugopalan, "A Supervised Approach For Extractive Text Summarization Using Minimal Robust Features," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 521-527 `https://ieeexplore.ieee.org/document/9065651`

4. H. Gupta and M. Patel, "Study of Extractive Text Summarizer Using The Elmo Embedding," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2020, pp. 829-834 https://ieeexplore.ieee.org/document/9243610
5. Nath, Sristy Sumana Roy, Banani. (2022). Towards Automatically Generating Release Notes using Extractive Summarization Technique. https://www.researchgate.net/publication/359918142_Towards_Automatically_Generating_Release_Notes_using_Extractive_Summarization_Technique
6. K. U. Manjari, "Extractive Summarization of Telugu Documents using TextRank Algorithm," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2020, pp. 678-683 https://ieeexplore.ieee.org/document/9243568
7. Miller, Derek. (2019). Leveraging BERT for Extractive Text Summarization on Lectures. https://www.researchgate.net/publication/333716903_Leveraging_BERT_for_Extractive_Text_Summarization_on_Lectures
8. Liu, Yang. (2019). Fine-tune BERT for Extractive Summarization. https://www.researchgate.net/publication/331986865_Fine-tune_BERT_for_Extractive_Summarization
9. Abu Nada Abdullah ,Alajrami Eman, Alsaqqa Ahmed and Abu-Naser Samy. (2020). Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach. https://www.researchgate.net/publication/344175857_Arabic_Text_Summarization_Using_AraBERT_Model_Using_Extractive_Text_Summarization_Approach
10. Sefid Athar and Giles C. (2022). SciBERTSUM: Extractive Summarization for Scientific Documents. https://www.researchgate.net/publication/358145432_SciBERTSUM_Extractive_Summarization_for_Scientific_Documents
11. Cao, Shiyun Yang, Yujiu. (2021). DP-BERT: Dynamic Programming BERT for Text Summarization. https://www.researchgate.net/publication/357483290_DP-BERT_Dynamic_Programming_BERT_for_Text_Summarization
12. J. Chen and H. Zhuge, "Extractive Text-Image Summarization Using Multi-Modal RNN," 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), 2018, pp. 245-248 https://ieeexplore.ieee.org/document/8703971
13. BartForConditionalGeneration https://huggingface.co/docs/transformers/model_doc/bart#transformers.BartForConditionalGeneration
14. PegasusForConditionalGeneration https://huggingface.co/docs/transformers/model_doc/pegasus#transformers.PegasusForConditionalGeneration
15. Bert Extractive Summarizer https://pypi.org/project/bert-extractive-summarizer/
16. Lewis, Mike et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.” Annual Meeting of the Association for Computational Linguistics (2019). https://arxiv.org/pdf/1910.13461.pdf
17. Bart explained. https://www.projectpro.io/article/transformers-bart-model-explained/553
18. Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, pages 11–20, Online. Association for Computational Linguistics. https://arxiv.org/abs/2002.09836

19. PEGASUS: A State-of-the-Art Model for Abstractive Text Summarization. https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html
20. ZHANG, Jingqing, ZHAO, Yao, SALEH, Mohammad, et al. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In : International Conference on Machine Learning. PMLR, 2020. p. 11328-11339. https://arxiv.org/pdf/1912.08777.pdf