Journal of Physics: Conference Series

Snippet generation using textbook corpus - an NLP approach based on BERT

Snippet generation using textbook corpus - an NLP approach based on BERT

Andrew Moses[1] and G. Bharadwaja Kumar[2]

[1]Student, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India.
[2]Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India.

[1]andrew2moses@gmail.com
[2]bharadwaja.kumar@vit.ac.in

Abstract. In today's technology-driven world, most millennials are tech-savvy. They have neither the time nor the interest in reading textbooks, newspapers or journals. They would like to immediately get instant answers and clarifications for all their doubts and questions. On many occasions, we are unable to find the exact word or meaning which we are searching for. So, if we have a clear, concise summary of a piece of literature, and we could understand what it contains with just a glimpse, we would be able to save a lot of time. This paper dwells about utilizing Natural Language Processing (NLP) to summarize a given text/textbook/paper. The state-of-the-art technology in this field has been demonstrated by Google's Bidirectional Encoder Representations from Transformers (BERT), one of the latest developments in NLP. BERT is believed to understand English better than other models because of its underlying bidirectional architecture. The present proposal is to use BERT as a sentence similarity extractor. By applying the TextRank algorithm, the sentences holding the most important information are extracted. This comes under the domain of extractive summarization. Abstractive summarization is much talked about, but since Google BERT is not built for generating text, we are utilizing it in a different way to achieve the requirement. This paper intends to discuss the use of BERT for the gen-next kids which will save time and initiate further interest for researchers in developing new programs continuously in the future.

1. Introduction

Text summarization has always been a challenging NLP task. Even though automatic evaluation techniques such as rouge scores have been introduced and practiced to bench mark Language Models (LM), still these techniques lack in understanding the efficiency of a model at semantic similarities between any two given text contents. This is because natural language is complex at its core. But these challenges never stopped research, rather fuelled it to invent newer techniques. Many top companies have research divisions who are working to compete each other every quarter. Especially 2020 has witnessed one of the greatest breakthrough in NLP with GPT-3 (Generative Pre-training model)'s release – A model which not just solves NLP task, but got capability to even code computer program on its own. When compared to GPT-3, BERT might look relatively old, but on taking a closer look, these

models might perform good in just bench marking. Industry is still yet to understand these model's capabilities and to find an appropriate industry use case. Thus, our research has picked up summarization NLP task as a use case to understand BERT in a deeper fashion oriented towards solving a specific NLP task.

Text Summarisation: we have extractive and abstractive methods. Extractive is about picking out the sentences or pieces of a given content and form a summary. While abstractive is about framing a summary which might not necessarily be part of the actual content. With the introduction of seq2seq (Sequence to sequence) modelling, abstractive summarisation is possible. But it has lot of challenges and its results have not been satisfactory so far. Couple of challenges include length of the input content, almost all the transformer models have a maximum of 512 tokens as input. Generally in a real world scenario a content length of minimum of 12,000 tokens might require the summarisation task, so the 512 token restriction fractures the transformers model's usage into the main stream industry. Hence, in this paper we would be discussing about extractive summarisation and BERT is built on transformers, even though there is a seq2seq modelling happening under the hood, BERT is not a generative model. Still, BERT's pretraining encoder stacks have proven to understand the context in a sentence. When compared with GPT-3, BERT's contextual understanding stands apart with embeddings capturing the underlying meaning. Going forward, the paper shall be discussing about the various existing techniques in text summarisation domain and then harnessing the abilities of BERT towards text summarisation problem.

2. Literature Survey

Language models has evolved over time. Machines being good at understanding numbers failed a lot when it comes to characters and natural language strings. To overcome this, common NLP pre-processing steps were utilized like stop words removal, lemmatization and stemming. These steps are very relevant till date. Even all the twitter streams undergo these common NLP steps. Basically, these steps reduce the vocabulary size of a language model and reduce the stress for machines to process complex English grammar. Word2vec is the most common technique used to convert tokens into vector embeddings. It follows either CBOW (continuous bag of words) or skip gram.

Word2vec had been the go to point for almost all NLP tasks, but the contextual knowledge is lost. Even though big deep networks could be used to process the vocabulary, still on performing the above mentioned common NLP steps the contextual semantic meaning between tokens are lost. Thus the models even after being trained with deep nets failed to perform. Recurrent neural nets (RNN) brought in the breakthrough by feeding in the output back into the net, thus networks evolved and was able to memorize its previous state to keep the contextual knowledge. As contextual knowledge plays an important role in NLP, RNNs became popular in NLP.

Transformers: it has taken NLP to its next era. Where generally it contains either a encoder, a decoder or a combination of both. It involves a tokenization process, were the input tokens are translated into model's understandable format. Then later comes a decoder block which again translates back the intermediate state of tokens back to its original state. The advantage of transformers lies in its vocabulary and embeddings of sentences. For example a sentence of 10-15 tokens might have a sentence embedding vector of size 512 dimensions. Even though it is difficult to hold such huge sized vectors in memory, but the model now has the capability to related with similar sentences and preserves the contextual information. Bidirectional encoders [1] also follows two sub tasks while consuming the widely available unannotated text in the world wide web. Next sentence predict and mask language modelling (MLM) are the two sub tasks. Next sentence predict: a semi supervised technique while helps the network to label the raw text in an unsupervised method were 50% of the text content is paired with its next sentence and label given as true for next sentence attribute. The other 50 is paired randomly and labelled as false. Similarly MLM is another simulated technique to label raw text. Both of the above discussed technique

are followed to understand the context in a truly bidirectional manner. This helps in having a pretrained model with flexibility of adding one additional output layer. The additional output layer would be used to fine tune the pretrained model with labelled data. Fine-tuning process extends the base model to solve specific NLP tasks like text similarity, etc.

Get to Point: Summarization with pointer generator networks [3] is an architecture which exploits neural sequence to sequence model, it solved summarization problem in abstractive method (summary which is not just rearranging passages from source, rather it generates text which need not necessarily be part of the source text). When it comes to benchmarking models, a crowd sourced Stanford Question Answering Dataset (SQuAD) [2] is much discussed. It is a labelled dataset which is made publicly available by Stanford University, the dataset was constructed by crowd sourcing questions from Wikipedia articles, where the answers to the questions will be a segment in the corresponding article. Now this dataset would serve in fine tuning the pretrained BERT model to accomplish Machine comprehension NLP task. SentEval: An evaluation toolkit [4] for representing sentences in a common platform. The toolkit provides scripts to which can be used to pre-process the dataset, thus helps in evaluating the sentences with different encoders.

Deeper self-attention exploration has indeed helped to cross this challenge. Natural language as an entity is available in abundance, but deriving machine understandable representations is lacking. Steps taken to derive labels from this abundantly unlabeled text is a framework proposed via machine learning researches. Coming to benchmarking evaluation techniques n-gram probabilistic language model has paved a standard way to track various model's efficiency.

3. Proposed Methodology

The proposed solution is to harness the power of pretrained BERT model to accomplish an open ended challenging NLP task – text summarisation. Several pretrained models have been open-sourced, larger the model better the performance. But, even the smallest model requires a decent Graphical Processing Unit (GPU) to perform fine tuning process. Among the various fine tuned NLP tasks that could be performed, sequence classification is the process which the proposed solution is targeted. In order to prepare the dataset to be consumed for the finetuning process the following steps are followed: Given a chunk of text for which summary needs to be derived, the raw text is sliced into meaning sentences using sentence tokenization technique. Sequence classification requires the data points to be a pair of sentences, hence each sentence tokenized sentence is paired up with every other sentence of the given input text content. Now the pre-processed data points are ready to be fed into the model for generating the sentence similarity score for all the pairs.

A out of box pretrained model might be good at generating embeddings, but on solving a NLP specific task, the model needs to undergo a supervised learning process. This is where a labelled dataset is fed into the network to train all the parameters of the model to solve a specific NLP task. Thus in this proposed solution the model is fine tuned with varying datasets and experiment results are discussed in detail in the subsequent sections. The model output is the predicted similarity score between each pair of sentence. Now the datapoints undergo a post process step which helps us to get the desired summary of a given passage. A similarity matrix is built, then text rank algorithm is applied. This generates a rank to every original sentence. Sorting the original sentences with its corresponding rank gets us the top K sentences as summary refer Fig 3 for the complete workflow.

What is special about computing similarity score via a BERT model: traditionally the same objective can be fulfilled by converting the tokens to vectors and then by calculating the distance between the vectors would yield the similarity score. But as highlighted earlier, in this approach, the contextual knowledge is lost. In the proposed solution the stop words are preserved, tokens are left at its original form (no lemmatization or stemming) and even the sentence context is consumed as such. Fig 4 shows

us two examples were BERT performance compared with simple cosine distance calculation. Note that cosine technique did not understand the semantic meaning within the sentence, rather its result is based on the magnitude of occurrences of tokens in given two sentence. But if one can read the sentence, they have completely an opposite meaning in example 2. Thus the proposed solution focuses on consuming BERT: a better and greater approach in solving NLP tasks.

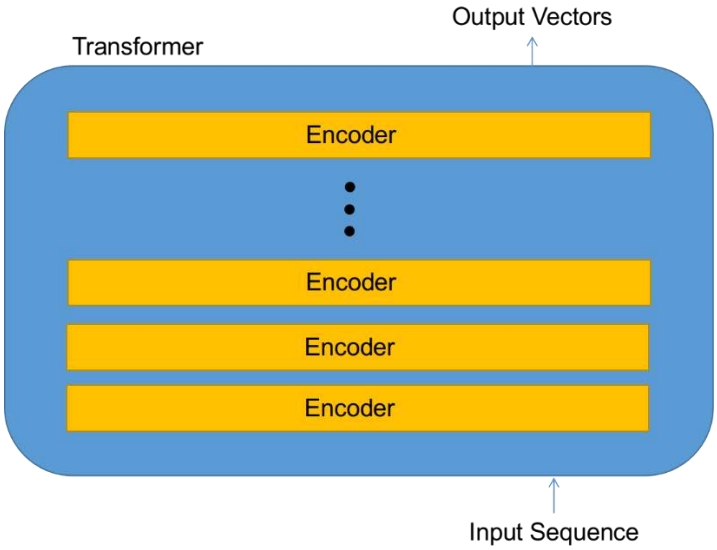

Figure 1. BERT Transformer

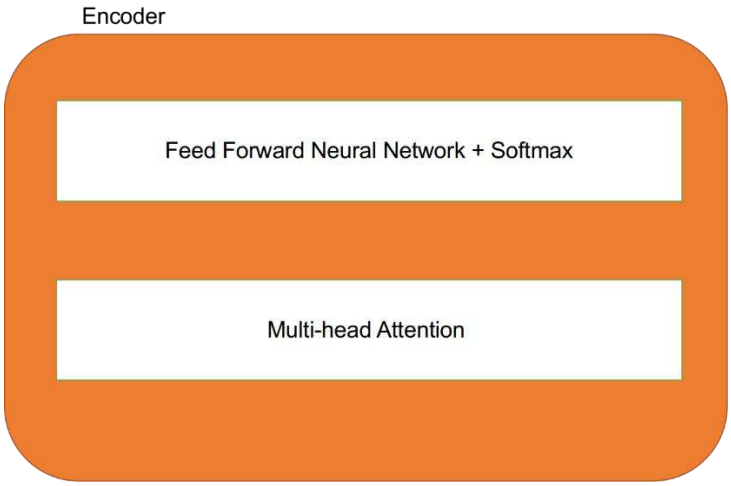

Figure 2. BERT Encoder

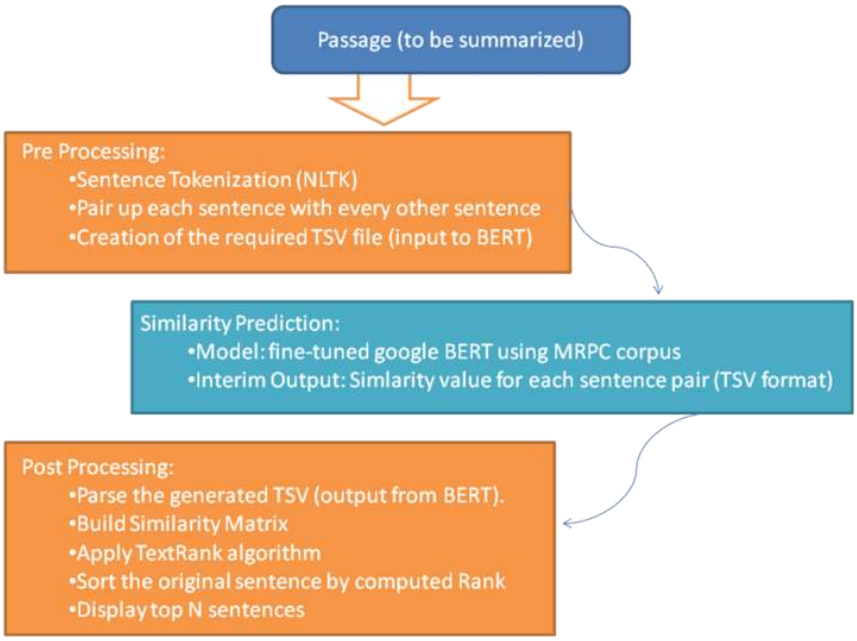

Figure 3. Workflow pipe-line

4. Design and Architecture

The core component of the proposed solution is the Bidirectional transformer block. It consists of a stack of encoders refer Fig 1. On contrast GPT2 transformer block consists of a stack of decoders. Each encoder block consists of Multi-head attention components, this helps the model to look back and front of a given sentence. Earlier the RNNs short comings were that the model could just see a short range of previous tokens to gather the contextual information, but in real world scenario, context need not always be at the previous N tokens, somethings it is not needed at all to look at previous N tokens rather the place to look out would be at the beginning/end of the passage. These challenges were addressed by the attention mechanism. The later half of the encoder block is the feed forward network with softmax as its output layer refer Fig 2.

Example 1
S1: " This day is indeed very good"
S2: "I was happy the whole day"
Cosine Similarity Score : 0.2886
BERT's Similarity Score : 0.6471

Example 2
S1: "Although interchangeable, the body pieces on the 2 cars are not similar"
S2: "Although similar, the body parts are not interchangeable on the 2 cars"
Cosine Similarity Score : 0.8888
BERT's Similarity Score : 0.0145

Figure 4. Cosine Vs Bert comparison

Pre-processing: In order to perform sentence tokenization Natural Language Tool Kit (NLTK) package was utilized, couple of experiments were also tried out with Spacy package instead of NLTK. Both the approach produced similar results. Even though the proposed solution preserved the context by not performing bag of words or stop words removal. Still, the special characters removal and lower casing of the text were done to reduce the vocabulary size. So, the pretrained base model was chosen as

the starting point. At first, the Microsoft Research Paraphrase Corpus (MRPC) dataset was used to fine tune the model to equip the model to perform sentence similarity task. Uniqueness of MRPC dataset is that they are human annotated. The last layer of the model was tweaked to perform this NLP specific task. The model consumes a pair of sentence and ends with two neurons (one for isSimilar and th other for isNotSimilar). The raw outcomes from the neurons are logits. On feeding the logits into a softmax function, the actual similarity score is generated. Further experiments were conducted by changing the dataset with a larger better paraphrase dataset. Paraphrase Adversaries from Word Scrambling (PAWS) contains around 45 thousand paraphrases with human annotations. Again on performing the finetuning the performance of the model was observed by feeding in manual 2 datapoints and it was bench marked against simple traditional cosine distance calculation refer Fig 4. The scope of the proposed solution was to demonstrate text summarization with academic research papers, but the MRPC and PAWS datasets lacks the research paper domain knowledge. Thus, steps were taken to simulate dataset which can produce domain knowledge as well the structure needed to fine tune the pretrained model. Abstract-article pair generation: it is the process were every sentence in the abstract section was paired with every other sentence in the article. Using simple pre-processing steps and cosine distance formula similarity score for all the pairs were calculated. Now, the pairs were sorted by the derived cosine similarity score. Then the top 5 sentence pairs were labelled as isSimilar while the bottom 5 sentence pairs were labelled as isNotSimilar. The intuition behind this process is that sentence with similar token occurrence results in similar meaning (which is not always true, but still). The other reason behind this process is to capture the domain knowledge. For example, the generic pretrained model would not have jargon knowledge used in research scenarios. Hence, this dataset induces the research contextual knowledge to the network. Especially when this simulated dataset is used to fine tune the model.

5. Results and Evaluations

Deeper look on proposed model vs traditional model: Jupyter notebook is the development platform used in these experiments, as it had the sophisticated interaction capability. A full length research paper (8-10 pages) will serve as the input. Top 10 sentences which has the most important information is the desired output. In order to evaluate the resulting top sentences, the output and the original research paper's abstract portion is compared using rouge metrics. In the proposed model following steps were followed: The entire input was split into individual sentences using Spacy's sentence tokenizer. Each sentence was paired with every other sentence in the article. This yielded the sentences pairs prepared in TSV format which was fed into the fine-tuned BERT model. It gave similarity score for every sentence pair. Using the resulting intermediate output, similarity matrix was constructed and on applying networkx pagerank operation, individual sentences were tagged with a rank. This gives the top 10 sentences as a snippet. Traditional model: The entire input was split into individual sentences using NLTK sentence tokenizer. After few cleansing operations like lower casing, removal of punctuation and special characters. The tokenized sentences were further filtered off by removing all the stop words. Filtered sentences were converted into vector representation using GloVe pre-trained vocabulary. Using the resulting intermediate output, similarity matrix was constructed similar to the proposed model and sentences were tagged with a rank. As mentioned earlier, the resulting output of both the models were evaluated against the original abstract of the test article on rouge metrics. For data sample 1: proposed model scored 30 in rouge-1 (recall) whereas traditional model scored 20 refer Fig 5. For data sample 2: proposed model scored 35 in rouge-L (recall) whereas traditional model scored 31 refer Fig 6.

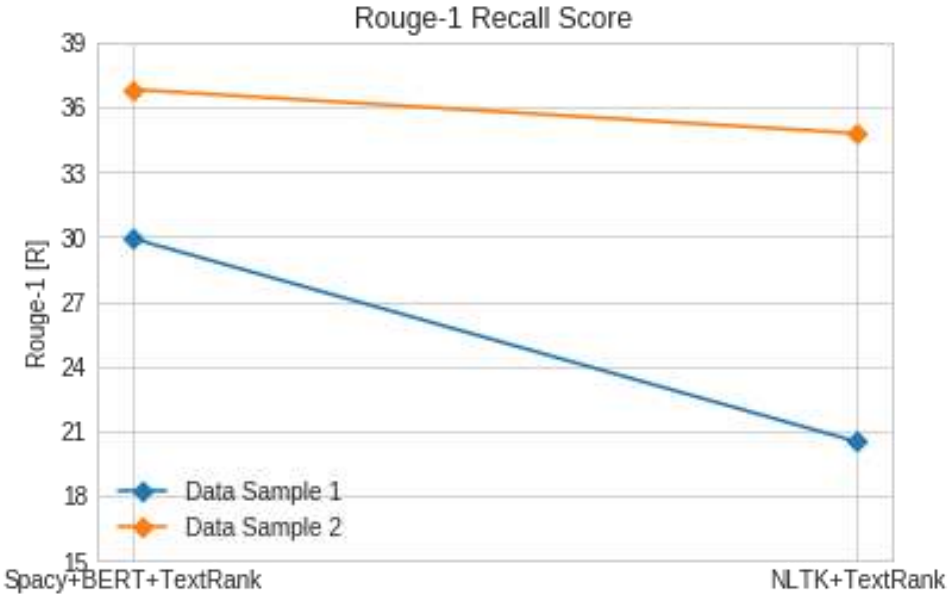

Figure 5. Comparison and Evaluation using Rouge-1 Recall Score

Figure 6. Comparison and Evaluation using Rouge-L Recall Score

6. Conclusion

Text summarization is an open ended NLP challenge. In short a famous news app is a direct business use case of text summarization's application. This research focuses on exploring a transformer model and understanding its capabilities. Also, we have gone a step ahead to tweak the model to solve a specific NLP task to benchmark the model and to pave a way to harness the real treasures of a pretrained model.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova 2019 BERT: pre-training of deep bidirectional transformers for language understanding *arXiv e-prints* 1810.04805.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev and Precy Liang 2016 SQuAD:100,000+ questions for machine comprehension of text *Proc. Conf. on Empirical Methods in Natural*

Language Processing (Austin) p 2383-2392.
[3] Abigail See, Peter J. Liu and Christopher D. Manning 2017 Get to the point: summarization with Pointer-Generator Networks *arXiv e-prints* 1704.04368.
[4] Alexis Conneau and Douwe Kiela 2018 SentEval: An evaluation toolkit for universal sentence representations *arXiv e-prints* 1803.05449.
[5] Alan Akbik, Duncan Blythe, and Roland Vollgraf 2018 Contextual string embeddings for sequence labeling *Proc. 27th Int. Conf. on Computational Linguistics (Santa Fe)* p 1638–1649.
[6] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones 2018 Character-level language modeling with deeper selfattention *arXiv e-prints* 1808.04444.
[7] Rie Kubota Ando and Tong Zhang 2005 A framework for learning predictive structures from multiple tasks and unlabeled data *Journal of Machine Learning Research* vol **6** p 1817–1853.
[8] Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo 2009 The fifth PASCAL recognizing textual entailment challenge *Proc. Text Analysis Conference (NIST)*.
[9] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992 Class-based n-gram models of natural language *Computational linguistics* vol 18(4) p 467–479.
[10] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson 2013 One billion word benchmark for measuring progress in statistical language modeling *arXiv e-prints* 1312.3005.
[11] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer 2017 Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension *arXiv e-prints* 1705.03551.
[12] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih 2018b Dissecting contextual word embeddings: Architecture and representation *Proc. Conf. on Empirical Methods in Natural Language Processing (Brussels)* p 1499–1509.
[13] Jeffrey Pennington, Richard Socher, and Christopher D. Manning 2014 Glove: Global vectors for word representation *Empirical Methods in Natural Language Processing* pages 1532– 1543.
[14] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer 2018 Deep contextualized word representations *arXiv e-prints* 1802.05365.
[15] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, Hsiao-Wuen Hon 2019 Unified language model pre-training for natural nanguage understanding and generation *arXiv e-prints* 1905.03197.