2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) | 979-8-3503-3086-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAC3N60023.2023.10541662

BERT: Extractive Text Summarization

Abhishek Kumar
Department of Computer Science and Engineering
Delhi Technological University
New Delhi, India
abhishekkumar_2k21afi23@dtu.ac.in

***Abstract* - In the last ten years, the disciplines of artificial intelligence, machine learning, and data science have experienced substantial growth. Artificial Intelligence is a crucial area of research that enables us to obtain accurate and concise information through Text Summarization Techniques. This project demonstrates the practical application of extractive text summarization technique, using the BERT model. The process involves obtaining a single sentence through extractive summarization and comparing it with the human-generated summarized text. The Rouge value, which represents the Precision, Recall, and accuracy between the two summaries, is calculated.**

Keywords- BERT (Bi-directional Encoder Representation from Transformers), Rouge-1, precision, accuracy, recall, Rouge-2, f1 measure, Natural Language Processing, Artificial Intelligence.

I. INTRODUCTION

Text summarization is a key method used in data mining and natural language processing to extract significant information from massive text files. For text summarizing, a number of models are used, including transformers, KL-summarizer, Luhn, LEX, Word Rank, GPT-1, GPT-2, and Bert Model.

The two main categories of text summarizing methods are:

Extractive Text Summarization: This method involves selecting key phrases, lines, or sentences from a paragraph and combining them to create a summary. For example, if the original text mentions that Helen and Jim attended a party in Delhi and Jim adopted a cat named Perl in the city, the extractive summary would be: "Helen and Jim attended a party in Delhi. Jim adopted Perl."

Abstractive Text Summarization: Abstractive summarization entails creating new words and phrases that successfully communicate the important details, even if they aren't stated clearly in the original text. Compared to extractive summary, it is a more difficult work since it calls for a deeper comprehension of the text and the capacity to construct grammatically sound and logical phrases. For instance, an abstractive summary of the same example could be: "Helen and Jim came to Delhi, where Perl was born."

These techniques play a vital role in condensing and presenting the key information contained within lengthy texts.

A. Problem Statement

The problem is summarizing the text correctly and getting correct rouge value when compared with human summarized text.

B. Objective

• Suggest an extractive text summarization method that can choose the most pertinent sentences from a given textual unit.

• In order to create summaries, we plan to use BERT.

• Additionally, we use the ROUGE measure to demonstrate the quantitative merits of our suggested approach.

II. LITERATURE REVIEW

Bidirectional Encoder Representations from Transformers, or BERT, is a key advancement in machine learning and natural language processing (NLP). Researchers at Google first introduced it in 2018, and the NLP community took to it with great enthusiasm.

Prior to BERT, left-to-right or right-to-left model training, where the model predicted the next word in a sequence based on the prior context, was the dominating strategy in NLP. However, this constrained the model's comprehension of the context and made it difficult to fully encapsulate a statement.

Bidirectional training, which was pioneered by BERT, transformed this strategy. It makes use of a neural network model called a transformer-based architecture, which uses self-attention processes to capture links between words in a phrase. The use of masked language modelling (MLM) and next sentence prediction (NSP) during pre-training is the main novelty of BERT.

BERT is exposed to a substantial corpus of unlabeled text during pre-training, including web pages, books, and Wikipedia articles. In MLM, a certain percentage of the input sentence's words are hidden at random, and the model is taught to anticipate the hidden words based on the surrounding context. BERT is able to acquire a thorough contextual grasp of words and their relationships as a result.

In NSP, BERT is trained to determine if two sentences

appear consecutively in the original text or if they are random pairs. This helps the model grasp the coherence and relationship between different sentences.

BERT's pre-training phase is computationally demanding and needs a lot of computing power. The model may be fine-tuned on particular downstream tasks, such as text categorization, named entity identification, question answering, and text summarization, when pre-training is finished.

BERT's introduction had a profound impact on the NLP community, leading to substantial advancements in various NLP tasks. It earned cutting-edge outcomes on a variety of criteria, demonstrating its greater comprehension of context and semantic connections inside phrases.

Since the introduction of BERT, a number of variations and modifications have been created, including RoBERTa [14], ALBERT [15] , and ELECTRA [16] , all of which seek to enhance the original BERT concept even further.

Overall, BERT's introduction marked a significant milestone in NLP, demonstrating the power of large-scale pre-training and transfer learning for natural language understanding and generation tasks. Its impact continues to be felt across academia and industry, driving advancements in language models and NLP applications.

III. METHODOLOGY

The fields of artificial intelligence, machine learning, and data science have all advanced greatly in 2018. Additionally, the number of underlying relationships and meanings needed to grasp the context of text and phrases is growing quickly in the field of natural language processing. The community for natural language processing has been developing fascinating components that users may use for free to build the pipelines and models they need.

One of the latest achievements, is the release in the development of the BERT model [1]. The Development of the BERT model has been marked as the beginning of new generation of Natural Language Processing. BERT is a model that broke all the records of handing language-based model tasks. The BERT model was developed by google brain in 2018.As soon as the model was release in the paper, it was already pre trained on the large datasets. It was a mesmerizing moment since its release given a major contribution in the field of machine leaning involving in natural language processing by saving resource.

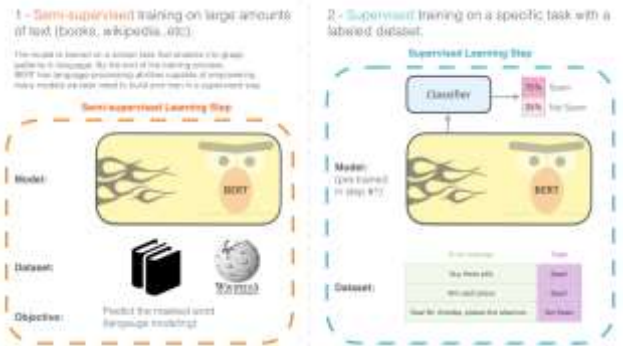

Fig-1. Supervised vs Semi-Supervised – [11]

BERT built on various ideas which is collection NLP community knowledge from various domain i.e., include Semi - supervised Sequence Learning [2], Embedding from Language Model [3], Universal Language Modelling-Fine Tuning [4], the Transformer model [5] and Open AI transformer[6].

There is number thing to be considered while using BERT model. So let look into this model.

A. Sentence Classification

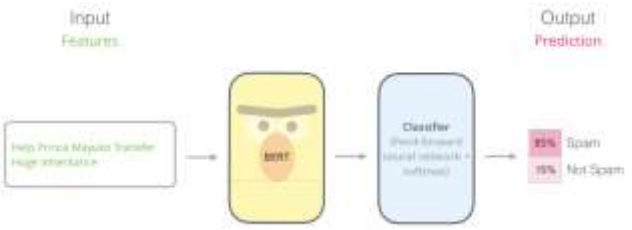

Fig -2. Word Classifier – [11]

One of the most important tasks performed by BERT model is to classify piece of sentence. The model will look like above. Here we have to train the classifier, in the training phase. We must account that there must be minimal changes to be made during the phase of training. Fine-Tuning [7] is termed for the training process and has connections derived from Universal language Modelling Fine tuning and Semi Supervised Learning. As we are talking about classifier, we must be familiar with the supervised learning domain of ML

We can take example to understand below. We need a labelled dataset to classify the sent email is spam or not. There are several datasets uploaded on the training dataset, based on those datasets it classifies whether the Email message is being classified as spam or not.

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

Fig 2. Spam classifier – [11]

Other examples include Sentiment Analysis, where we can Movie/product whether the. Is it good or bad?

B. Model Architecture

Now, we must look on BERT MODEL how it works.

A trained Transformer Encoder stack is essentially what BERT is. The paper [6] shows BERT model is comprised of two-part BERT Base and BERT Large.

We can differentiate BERT Base and BERT Large in the following ways.

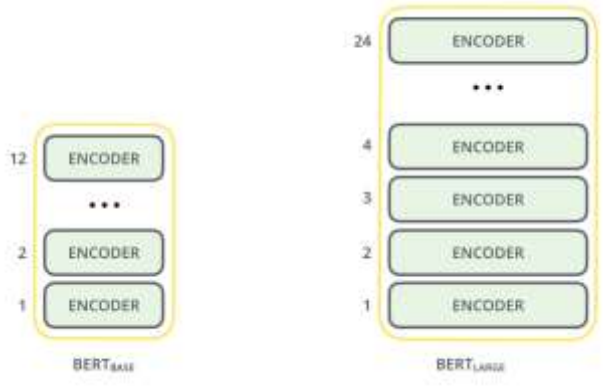

Fig 3. $BERT_{Base}$ Vs $BERT_{Large}$ (Encoder Stack) – [11]

TABLE 1 $BERT_{Base\ Vs\ Large}$ Structure

	BERT BASE	BERT LARGE
Size	Relatable in size to the AI Transformer	Ridiculously Huge Size
No. of Encoder Layers	12	24
Feedforward Networks	768	1024
Attention Heads	12	16

Conclusion is that, the default version has 6 Encoder layer,512 hidden units and 8 attention heads.

C. Model Input

The first input is applied with special padding [CLS], stands for classification. BERT takes sequence of words from sentence just like a vanilla encoder which moves up in the stack format. Each layer applies self-attention and the output is passed by feed forward neural network. Once the output is passed by the feed-forward neural network, it is transferred to the next encoder.

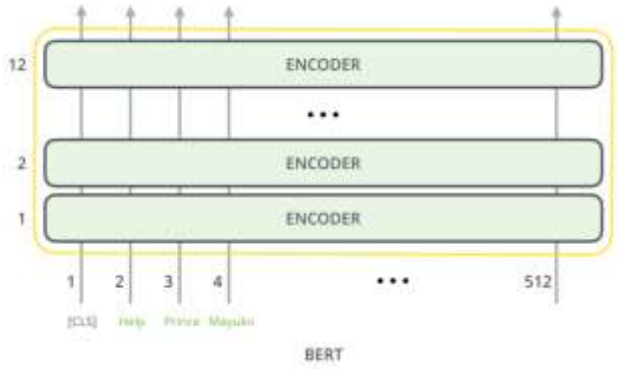

Fig 4. Encoder Stack – [11]

D. Model Output

Each component creates a hidden size of 768. For text classification, we just need to pay attention to entry position. We can now add that vector to the classifier's input for our requirement. The research yields great results even with a one-layer NN functioning as the classifier.

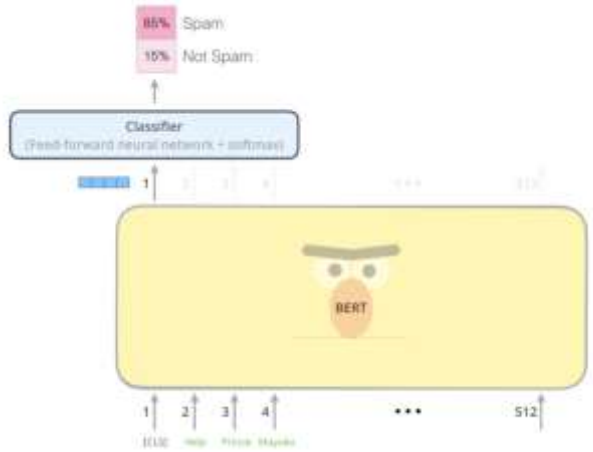

Fig. 5. BERT Classifier – [11]

If we wish to have more classifier in out BERT mode like "promotion or social" we need more output neurons in our classifier network.

E. Parallels With Convolutional Nets

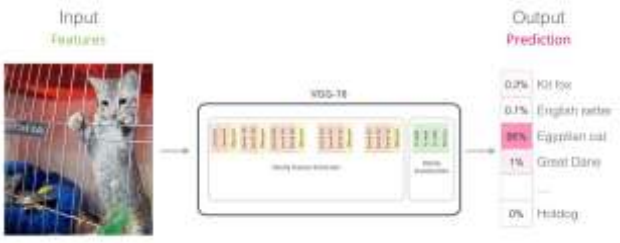

Fig 6. BERT Prediction – [11]

Here picture is passed from input feature. Once it is passed from input feature it is passed to VGG-16 convolution nets. After convolution, based on the classifier it is predicted as output i.e., Egyptian cat.

F. ELMo

Wordtovector [8] utilizes a static embedding for each keyword, whereas Elmo considers the entire text sentence before allocating a word to an embedding. It uses a bi directional Long Short-Term Memory trained model to create embeddings for words.

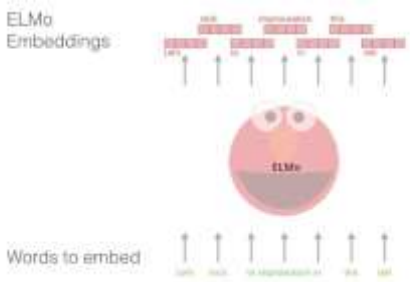

Fig 7- ELMo – [11]

Now, there was a similar model used GloVe() which was used to word embedding but there was problem with GloVe as the it has embedding based on the context but fails to have contextual understanding.so ELMo [3] was developed. The benefit of ELMo was that it can have both understanding of context as other information around the context.

ELMo examines the entire text before assigning an embedding to each word; it does not have a fixed embedding for each word. It employs two-direction Long short-term memory with a particular function for word embedding in sentences.

Embedded in Language Modelling provided an important step toward pre training of dataset in the background of natural language processing. Embedded in Language Modelling is mostly trained to classify next word in dataset. Language Modelling is defined to make the language understand to classify the next word from a sequence of word. It is helpful, as we have large and massive dataset, so models can train and give predictive results based on the labels.

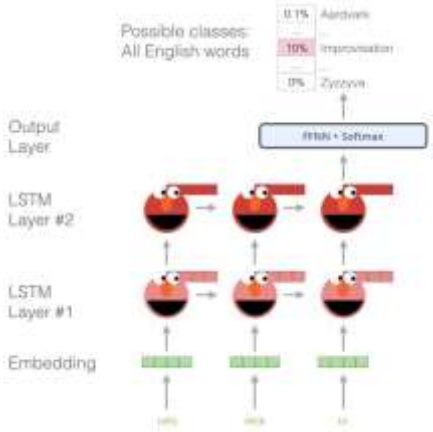

Fig 8. ElMo (LSTM) – [11]

Behind ELMo's head, we can see the concealed condition of every unrolled-Long short-term memory phase. Following the completion of this pre-training, they are useful during the embedding procedure. ELMo really goes a step ahead and trains a two-directional Long short-term memory ensuring that its language model understands both the previous and subsequent words.

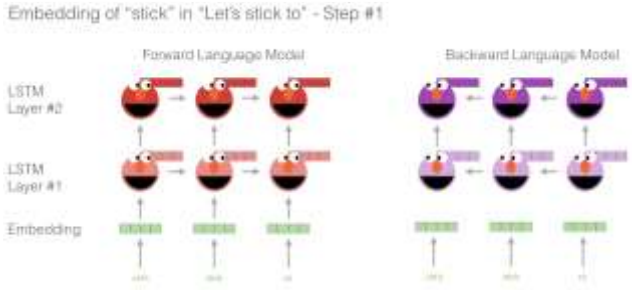

Fig 9. Elmo (Bi-directional LSTM) – [11]

ELMo in contextualize embedding has certain steps:

1. Concatenate hidden layer
2. Multiply each vector by a weight based on the task.
3. Sun the (now weighted) vector.

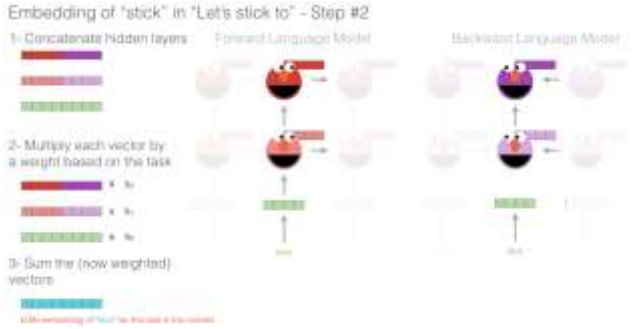

Fig 10. Elmo (Conceptual Understanding) – [11]

G. ULM-FiT

Universal Language Modelling Fine Tuning is highly used in computer vision industry. It is more than just word embedding as it is responsible for holding contextual holding. ULM-Fit fine tunes the model on various tasks.

H. OpenAI Transformer

According to the Research paper [6] , we don't require entire transformer for fine tuning of Model. We only want decoder of the transformer. The transformer is built on "mask feature taken "which is important for generating word by word. The openAI transformer contains 12 decoder layers stacked one after another. There is no presence of encoder. These are different from vanilla transformers where encoder -decoder have attention sublayer. But still, it has self-attention layers.

With their structure, we can train the data model and get next word predicted from massive dataset.

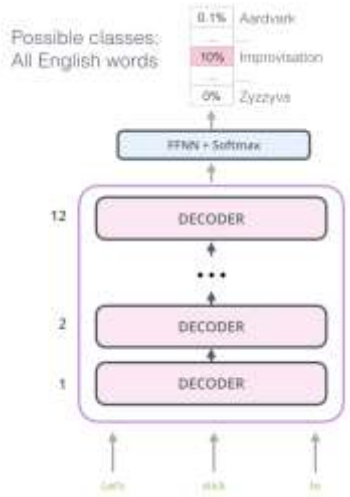

Fig 11. Decoder attention sublayer – [11]

In the above example it is written “Lets sticks to”. Now other 3 words can be predicted from the following labelled database i.e. Aardvark (0.1%), Improvisation (10%) and Zyzzyva (0%). So, we choose improvisation with 10%.

I. Transfer Learning

The same openAI transformed is pr-trained with labels. And when they get unlabeled datasets, they can predict which is slam and which is not.

The openAI paper clarifies various input transformers to handle different types to tasks for different input. Before the few structures shown in paper involve here, we list some pic above different input structure types.

J. BERT Model: From Decoder to Encoder

It is very difficult to train right task to a Transformer from the tack of encoder. This is resolved by” Masked Language Model” where fifteen of the input is masked. Masking is done to predict actual significance of words in that position.

~~K.~~

~~L.~~

K. -Two-Sentence Tasks

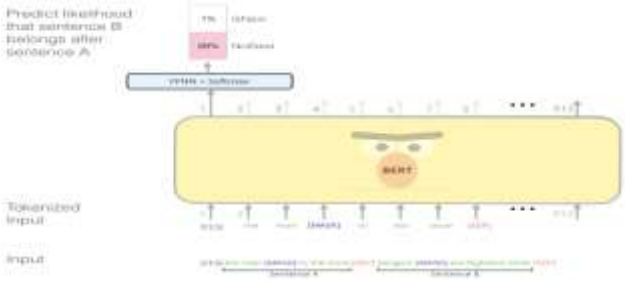

Fig. 12. Masked Language Model – [11]

There is various task where openAI has to answer intelligent about two sentences. To make this up, BERT Model is efficient in handling relationship between multiple question. Such as Predict likelihood that sentence B belongs after sentence A.

~~M.~~L. Feature Extraction (BERT Model)

Similarly, like Embedded from Language Modelling, we can use before-trained BERT model to create contextualized word embedding rather than just fine tuning. Then you can use this embedding for existing Mode. Below is the figure for contextualize word embedding.

Fig 13. BERT Language Modelling – [11]

Which is the best contextualized embedding vector is determined by the above six choices.

~~N.~~M. ~~3.13~~ ROUGE Score

Here we calculate rouge score [9] between human text summarized vs BERT model text summarized text. Rouge score determines similarity between BERT Model summary against human Summary. Important point about rouge score is that it is not case sensitive, it doesn’t able to distinguish between upper case alphabet vs lower case alphabet.

~~[7]~~ Rouge score is evaluated in 3 parts:

Rouge-1-It determine how many single words are similar in produced summary against reference summary (Unigram).

Rouge-2- It determine how many two words are similar in produced summary against reference summary (Bigram).

Rouge-L-It is also known as Least Common Sequence where rouge is determined d how many sequences of words are similar produced summary against reference summary.

Where it is Rouge 1, Rouge 2 or Rouge 3, its accuracy is defined by these 3 measures.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (1) \ \& \ (2)$$

1. Precision- Precision is defined as the ratio of the actual number of positively predicted phrases to all of the anticipated terms.

2. Recall-It is also called as sensitivity. It is defined as the proportion of accurately categorised true positive instances to all really positive phrases.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}} \quad (3)$$

~~3.~~ F-measure – It is referred as dependency between precision and recall. If anyone falls, whole value falls.

~~4.~~

3.

~~V.~~IV. EXPERIMENTAL RESULT

We have worked on the dataset ~~#Metoo and #uselection.~~ #UsElection dataset [10].

Step 1: Data cleaning. Removing column 0 and 1, non ascii values and html tag.

Step 2: Using Bert Summarization Model

Step 3: Using Rouge Score to calculate recall, precision and F-measure between model summary and 3 different human summarized summaries.

The following are the output for BERT Model for #UsElection -Dataset.

TABLE I

Rouge 1	
Recall	0.559
Precision	0.240
Frequency	0.336
Rouge 2	
Recall	0.279
Precision	0.104
Frequency	0.152
Rouge 3	
Recall	0.541
Precision	0.232
Frequency	0.325

TABLE II

Rouge 1	
Recall	0.585
Precision	0.219
Frequency	0.319
Rouge 2	
Recall	0.256
Precision	0.084
Frequency	0.126
Rouge 3	
Recall	0.557
Precision	0.209
Frequency	0.304

TABLE III

Rouge 1	
Recall	0.479
Precision	0.209
Frequency	0.291
Rouge 2	
Recall	0.216
Precision	0.076
Frequency	0.113
Rouge 3	
Recall	0.441
Precision	0.192
Frequency	0.268

The TABLE ~~I,II, ,II, shows~~II shows BERT model prediction with 3 different human summarized text.

~~VI.~~V. LIMITATIONS

- Due to the training framework and corpus, the model is big.
- Because it is large and there are several weights to update, training takes a while.
- It is high in price as it requires high computational speed and power.
- It must be adjusted for downstream activities, which can be fussy, because it is designed to be fed into other systems rather than as an independent application.

VI. CONCLUSION

From the working mechanism we could implement Rouge score between human text summarized and BERT summarized text. Based on the different dataset and effective dataset cleaning, BERT model gives efficient answer. I have shown result in "implementation Output Section". We have even shown limitations and drawback.

User must have high computing PC to implement this algorithm.

One more important BERT Model is that, it is a classifier algorithm not an extractive summarization algorithm. One has to precisely select dataset for handing the task process.

BERT is a cutting-edge NLP model with tremendous power. The pre-trained model was developed using a vast corpus of data, and you can fine-tune it using a smaller dataset based on the task and your needs.

REFERENCES

[1] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[2] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." *Advances in neural information processing systems* 28 (2015).

[3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[4] Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." *arXiv preprint arXiv:1801.06146* (2018).

[5] Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving language understanding by generative pre-training." (2018).

[6] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[7] Liu, Yang. "Fine-tune BERT for extractive summarization." ArXiv preprint arXiv:1903.10318 (2019). arXiv:1810.04805 (2018).

[8] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[9] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81. 2004.

[10]https://github.com/ad93/FairSumm/tree/master/Dataset/US-Election

[11] http://jalammar.github.io/illustrated-bert/