Zewail City of Science and Technology
University of Science and Technology
CIE 555 - fall 2020

Extractive
Text Summarization

BY:

Dina Adel	201601463
Alaa Hesham Mahmoud	201500638
Ahmed Ali	201701100

May 23, 2020

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

1 | Problem Definition & Motivation

Text Summarization is condensing the source text into a shorter version preserving its information content and overall meaning. Due to the huge progress that happened in computer capabilities and the internet revolution, text data has become available everywhere. In many cases, the huge data available is a problem because exploring it may take a lot of time. Besides, most of this text may not provide valuable information. So, it is a very time consuming task for humans such as researchers. Hence, text summarization will make it easier for humans to review the key points in an article, book or paper without the need to read the whole thing. Hence, text summarization is a valuable task to tackle. Therefore, we are encouraged to use deep learning architectures to come up with state of the art results on this task.

2 | Literature Review

Automatic text summarization is condensing the source text into a shorter version while preserving its information content and overall meaning [1]. The main goal is to reduce read time while keeping redundancy to the minimum and highlighting the main topics within the document. There are two types of text summarization: extractive and abstractive summarization. In extractive summarization, the main task is to extract the most important sentences from a document i.e. electing some sentences of the main document to represent it in a summarized format. In abstractive summarization, it depends on examining the document at hand and interpreting it to determine main concepts and generate a shorter text reflecting them. We will be focusing on extractive summarization as our task.

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

Before the usage of neural networks, feature extraction methods were used to determine the location of important sentences. An example of them is the TF-IDF vectorizer where tf stands for the frequency of the term in the document and idf stands for the inverted frequency of documents containing this term. Moreover, idf is a measure of whether a term is frequent or rare across all the available documents. The relevance of a term in a document was determined by tf x idf measure. So, this method tended to filter the common words [2].

After the revolution of neural networks, the network itself now can learn the important features from a sentence and whether it should be included in the summary or not. This is done by training the network on sentences within several test paragraphs and deciding whether the sentence is relevant or not. The recent advances of sequence to sequence models as well as language models has dramatically pushed the progress in the field of Natural language processing (NLP). Now, we have models which are capable of achieving great results in several tasks such as sentiment analysis and summarization.

In [3], there is a comparative study of recent methods and models used in extractive text summarization. One of the models is the Bidirectional Encoder Representations for Transformers (BERT) model introduced by Stanford research group. BERT is a bidirectional, unsupervised language representation, trained with a masked language modeling (reconstructing the output from a corrupted input) and a "next sequence prediction" task on a corpus of 3300 M words [4]. BERT uses transformer which is an attention mechanism that catches contextual relations between words (or sub-words) [18]. As we can interpret from the following figure, the BERT model is outperforming the other models in the task of extractive text summarization. So, we will focus on using it.

Zewail City of Science and Technology
University of Science and Technology
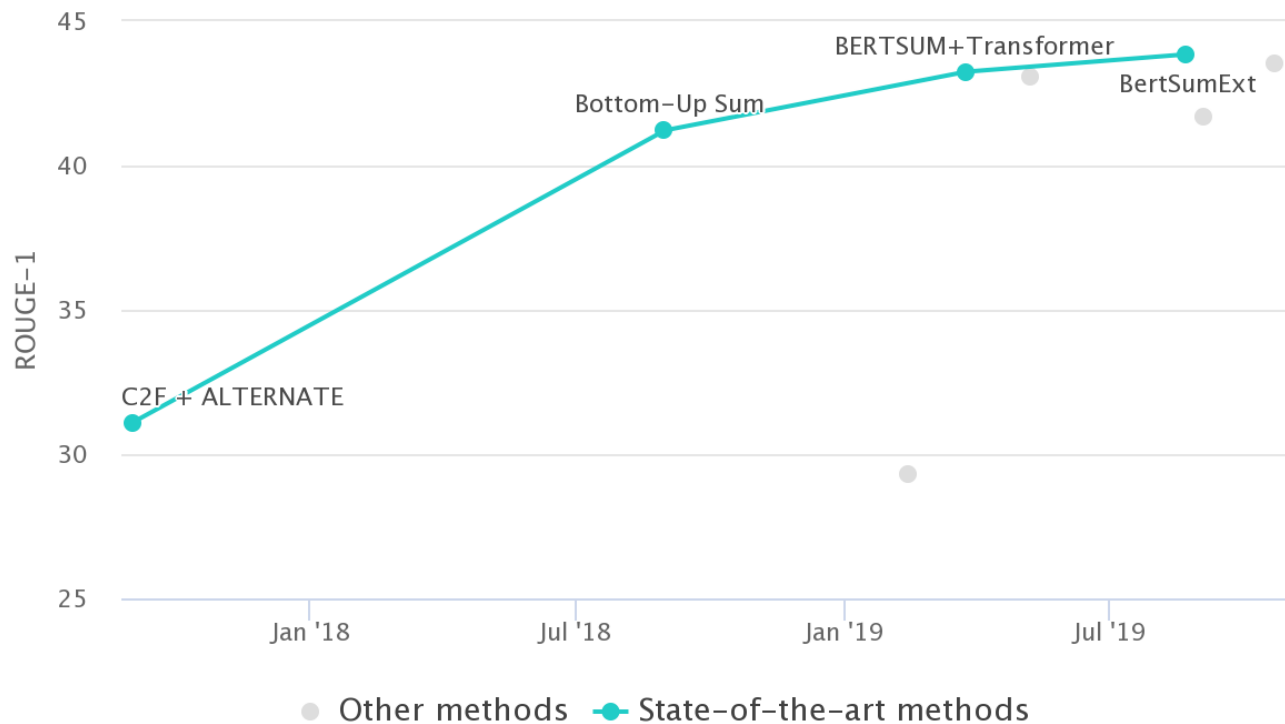CIE 555 - Spring 2020

\\

Figure 1 State of the art methods on summarization [3]

Several papers tackled the BERT model as a way for Extractive Summarization. Below we will mention some of them along with their results. The following paper represents the history of using text summarization techniques to summarize lecture contents for students. It focuses on the exerted effort to achieve this goal starting from manual text summarization and up to the usage of transformers such as Bert [5]. At first, many universities created manual text summarization for their lectures such as MIT's lecture processing project. It was good when the content was small, but when the content was large, it needed more and more effort. In 2005, researchers made automatic extractive text summarizers, but they found that its output is poor [6]. Six years later, a commercial application called "OpenEssayist" was created. Its aim was to aid students while completing their assignments or essays. It would highlight the main topics and key points in any text. In "OpenEssayist", multiple types of summarization options were implemented.

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

Moreover, they used algorithms such as TextRank for key sentences and keyword extraction [7]. TextRank is a graph-based algorithm which is based on voting. A word is considered as a vertex in this algorithm; and is voted for when another vertex leads to it. Moreover, the higher the votes to that vertex, the more relevant it is to the summary. In [8], researchers built applications that used Naive Bayes algorithms to determine which part of the slides is more descriptive to the content of the lecture. In the last few years, there are a lot of papers attempting to summarize lectures. In the book "In Recent Developments in Intelligent Computing, Communication and Devices", the author implements video subtitle extraction program that would summarize the multimedia input by utilizing TF-IDF method (this approach will be discussed in more detail in the next parts) [9]. The problem with this approach and Naive Bayes approach is that they represent the complex phrases in a bad way. The researchers are then moved towards deep learning to improve text summarization. Recurrent neural networks (RNN) using LSTM (Long Short Term Memory) become very popular in many NLP tasks, but it needs huge amounts of data and huge computational resources to reach high performance. The researcher, Vaswani, was the one who represented the architecture called "transformer" which behaves like sub-human performance in many NLP tasks. By the end of 2018, researchers at Google built an unsupervised architecture called BERT on top of the transformer architecture. That exceeded all previous models in many NLP tasks including text summarization. Also, the researchers published pre-trained models used in transfer learning. K-means clustering was used with BERT in this paper. It takes tockinze and uses the BERT model to get its embedding - convert text to a vector - then uses K-mean clustering algorithm and takes the centroid part of each cluster in the summary. To do the lecture summarization, the core of the BERT model is used with the pytorch-pretrained-BERT library from the "hugging face" organization. Pytorch-pretrained-BERT library also contains the OpenAi GPT-2 model which is another powerful model based on expanding the BERT architecture. However, when the sentence embedding coming from both the GPT-2 and BERT is examined, the BERT was more representative in the sentence. Screenshots of the results mentioned in the papers: In a lecture about semantic interoperability in health

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

exchanges, both the BERT model and the TextRank model were examined .According to [5], the BERT model overall had better results than the TestRank approach. Moreover, the BERT model summary was more able to add a broader context and sentences that flowed together than the one generated by the TextRank model. BERT was also used in [4] where both extractive and abstractive summarization were performed. However, we are only interested in the Extractive Summarization part. They used the BERT pre-trained model. Even though the BERT model has been fine-tuned on several NLP tasks, its application on summarization was not straightforward since the model's output vectors are based on tokens instead of sentences. So, the BERT model was modified as shown to accommodate for individual sentences:
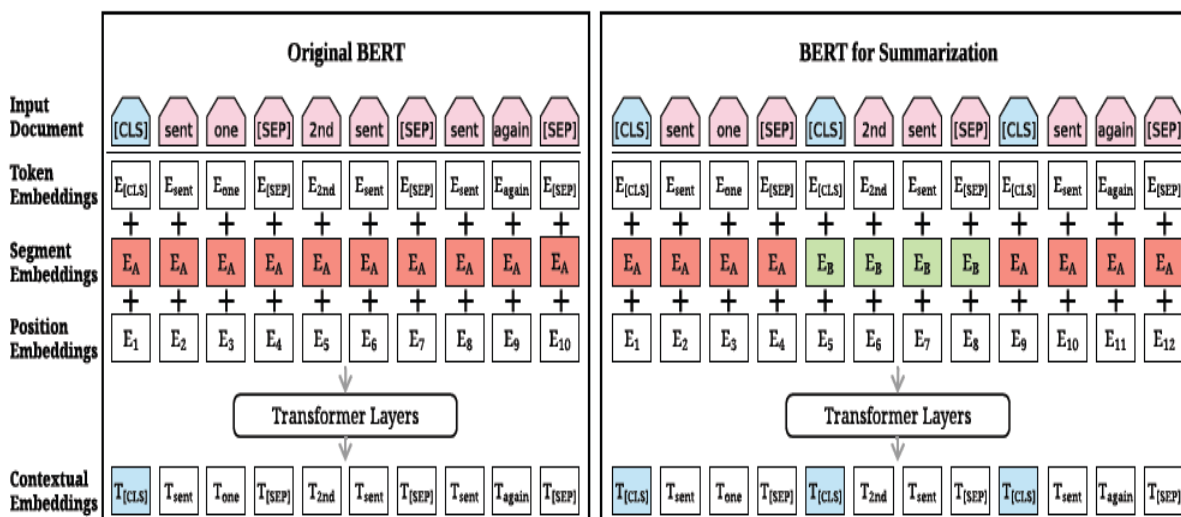

Figure 2 Original BERT and BERT for Summarization. Source: [4]

Vector t-i which is a vector of the i-th [CLS] is used to represent each sentence-i from the document. CLS is the first token in each sequence and is used as a representation for every input sequence in classification tasks. To continue, several inter-sentence Transformer layers are then stacked on top of BERT outputs, to capture document-level features for extracting summary. The transformer was mentioned earlier. However, a transformer, explained simply, uses attention-mechanism which is taking

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

each sequence and deciding at each step what is important and what is not. Moreover, a transformer transforms an input sequence into another one by using an encoder/decoder method without the use of recurrent networks such as LSTM.

What the authors of the paper used?

- The output is a sigmoid classifier.

- The loss of the model is the binary classification entropy.

- Adam Optimizer was used ($\beta1=0.9, \beta2=0.999$).

- The extractive models were trained for 50,000 steps.

- They tested their model on 3 datasets: CNN/DailyMail news highlights dataset, the NewYork Times Annotated Corpus, and XSum.

- The summarization was automatically evaluated by Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric.

- Those were the results obtained from the authors of [4] compared with the work of other authors; where R-1 and R-2 are means of assessing how informative the summary is and the longest common sequence; RL is a means of assessing fluency. For more information about the authors results, please refer to [4].

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

Model	R1	R2	RL
ORACLE	49.18	33.24	46.02
LEAD-3	39.58	20.11	35.78
Extractive			
COMPRESS (Durrett et al., 2016)	42.20	24.90	—
SUMO (Liu et al., 2019)	42.30	22.70	38.60
TransformerEXT	41.95	22.68	38.51
Abstractive			
PTGEN (See et al., 2017)	42.47	25.61	—
PTGEN + COV (See et al., 2017)	43.71	26.40	—
DRM (Paulus et al., 2018)	42.94	26.02	—
TransformerABS	35.75	17.23	31.41
BERT-based			
BERTSUMEXT	46.66	26.35	42.62
BERTSUMABS	48.92	30.84	45.41
BERTSUMEXTABS	49.02	31.02	45.55

Figure 3 BERT performance on summarization tasks. Source [4]

3| Methodology

The project baseline is composed of three pillars:

1. Model to perform summarization
2. Dataset
3. Evaluation metric

- **Model**

  Based on our literature review, we conclude that Bert is the best candidate to use as a model for text summarization. The process of using it has exposed us to technical challenges that are mentioned in details in the appendix section.

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

- **Dataset**

    As previously mentioned in the proposal, we would choose between the CNN/DailyMail dataset and the WikiHow dataset. We chose to use the WikiHow dataset. WikiHow is a dataset of 230,000 articles and summary pairs from an online knowledge base. Each article in the dataset is composed of multiple paragraphs. Any paragraph has a sentence summarizing its content. Combining article's paragraphs together forms the main document. Combining the sentences together form the summary. We have chosen to use it for various reasons. First, WikiHow is a dataset built for summarization tasks specifically according to [16]. Second, WikiHow has more than 230,000 pairs of articles written by journalists. Third, there was a comparison between the CNN/DailyMail and the WikiHow dataset shown below:
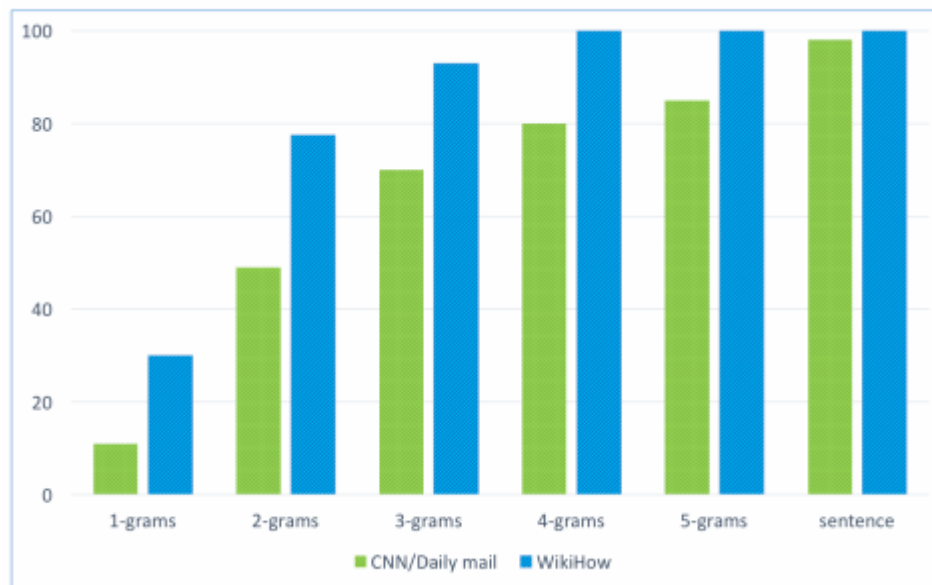

Figure 4 Uniqueness of n-grams in CNN/Daily mail and WikiHow dataset .Source [16]

    The abstraction of a dataset is measured by the uniqueness of its n-grams.As illustrated in this comparison, WikiHow has more abstraction than the CNN/DailyMail dataset. Another comparison was made in highlighting the compression ratio of each dataset. The compression ratio is the ratio between the average length of sentences

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

and the average length of summaries. Having a higher compression ratio indicates that the summarization task is also more challenging [16].

	WikiHow	CNN/Daily Mail
Article Sentence Length	100.68	118.73
Summary Sentence Length	42.27	82.63
Compression Ratio	2.38	1.44

Compression Ratio of both CNN/Daily Mail and WikiHow dataset

Technicalities of preprocessing the dataset is mentioned in the appendix section.

- **The Evaluation Metric**

As an evaluation metric, we had two options: the ROUGE or the BLEU metric. We chose to use ROUGE as we found that papers that tackle the Extractive Summarization usually use it as a metric. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation; and is used for evaluating automatic summarizations of texts or machine translation. It works by comparing a model's generated summary with reference summaries. It evaluates the quality of these summaries based on calculating the precision, recall and F-Measure. There are multiple types (of evaluation metrics) of ROUGE. However, we will only focus on 2 of them: ROUGE-N and ROUGE-L. ROUGE-N measures the n-gram overlap between the model's summary and the reference summary. Moreover, ROUGE-L measures the longest matched sequence between them. In short, we evaluated our model based on ROUGE-1, ROUGE-2 and ROUGE-L.

Fine tuning:

We have attempted to fine tune the BERT model. However our attempts have not succeeded for three main reasons. Unlike other NLP tasks, there is no literature to review considering this task except for one paper that is poorly written [17]. Second, the BERT model provided on google research repository is only a base model and not specified for any NLP Task. Third, Hugging Face created customised

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

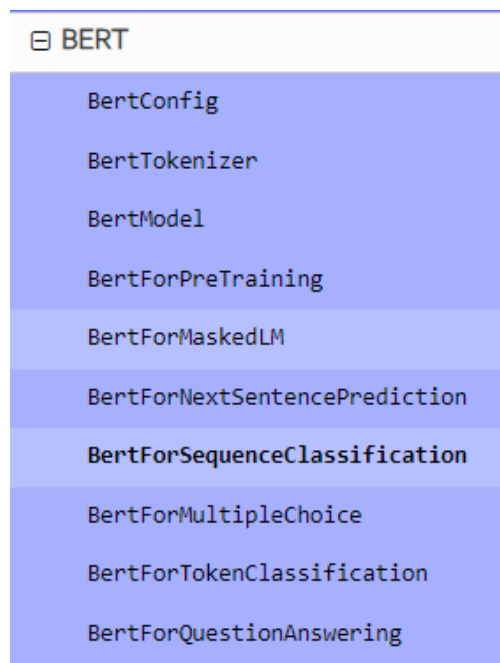interfaces for fine tuning BERT on different tasks but summarization was not one of them.

Figure 5 BERT fine-tuning interface for other NLP tasks.

4| Results

We used the Rouge metric to evaluate BERT model performance on the WikiHow dataset. We are concerned with R-1, R-2 and R-L metrics. Any metric of them has three numbers to represent its value: Recall, precision, and Score/F-Measure. Recall equals the number of overlapping words between system summary (BERT summary) and reference summary (generated by humans) divided by the number of words in reference summary. Precision equals the number of overlapping words between system summary and reference summary (generated by humans) divided by the number of words in system summary .F-measure combines both recall and precision together in one number to indicate the score of the model/BERT in the task .Our results are illustrated in the following table

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

Metric percentage	R-1	R-2	R-L
Score	27.070805	19.640298	22.238973
Precision	23.718564	18.614623	17.925122
Recall	39.440298	22.485895	34.7645

Figure 6 our results with BERT model on Wiki-How dataset using Rouge

We could not find a paper that tested the Wiki-How dataset on BERT. The reason is that they are both recently published .However, we managed to find other popular models and their scores on the Wiki-How dataset to compare our results with [16]. We can see that BERT has achieved the highest score on Rouge-2 and achieved near optimum results on both Rouge1 and Rouge-L. In Rouge-1, BERT has achieved 27.07 while the highest score is 28.53. In Rouge -L, BERT has achieved 22.23 while the highest score was 26.54.

Model	CNN/Daily Mail				WikiHow			
	ROUGE			METEOR	ROUGE			METEOR
	1	2	L	exact	1	2	L	exact
TextRank	35.23	13.90	31.48	18.03	27.53	7.4	20.00	**12.92**
Seq-to-seq with attention	31.33	11.81	28.83	12.03	22.04	6.27	20.87	10.06
Pointer-generator	36.44	15.66	33.42	15.35	27.30	9.10	25.65	9.70
Pointer-generator + coverage	39.53	17.28	36.38	17.32	**28.53**	**9.23**	**26.54**	10.56
Lead-3 baseline	**40.34**	**17.70**	**36.57**	**20.48**	26.00	7.24	24.25	12.85

Figure 7 Other Models Scores on Wiki-How dataset

Also, we have also evaluated it on the only first 70 articles and calculated the average score as the processing was slow and consumed a lot of time.

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

5| Conclusion

We conclude that the need for text summarization has emerged due to the huge content of text available today. Besides, the fact that text summarization task is necessary but time consuming. It has two types: Extractive and abstractive text summarization. Extractive text summarization is about electing some sentences from the document to be included in the summary. These sentences are expected to represent the document in a summarized format .BERT model has outperformed other previous suggested models in the task of summarization .That is why we have used it. Regarding dataset used, we have encouraged to use WikiHow dataset to apply BERT model on it for many reasons; it is made especially for summarization. We could not find papers in literature use it along with BERT so we were tempted to figure out how BERT performs on it. We have used Rouge-1, Rouge-2, and Rouge-L to evaluate BERT summarization and its results were satisfying. It has achieved the highest score on Rouge-2 and achieved near optimum results on both Rouge1 and Rouge-L. In Rouge-1, BERT has achieved 27.07 while the highest score is 28.53. In Rouge-L, BERT has achieved 22.23 while the highest score was 26.54. Fine tuning BERT for text summarization is a challenging task. However, we are now much more capable of tackling it now than previously. So it is considered as future work.

6| Appendix

Model Technicalities:

Trying to deal with the BERT model directly from google research github was pretty overwhelming as it only contains the base model .It is not tailored for any specific NLP task with no easy to use interface for developers . That has motivated us to search for other technical solutions . The best choice was to use Pypi v 0.42 tool which utilizes hugging face interface and coreference techniques altogether . Hugging Face is a company which has raised 15 million dollar to push the field of NLP forward by making easy to use

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

interfaces for most effective deep learning architectures for developers and other services . NeuralCoref is a pipeline extension for spaCy 2.1+ which makes preprocessing text an easier task by providing many tools for doing this . Pypi tool combines both Hugging Face interface and NeuralCoref to give a combat interface for dealing with the BERT model for extractive text summarization . Here is a guideline on how to use it:

P.s. We find it is useful to document steps in the report as there were no tutorials guiding us to how to do it. Hopefully, we will write a blog post about it:

- First, create a python virtual environment with the aid of the following video .
- Second, use pypi v 0.42 tool which utilizes hugging face interface and coreference techniques, to resolve words in summaries that need more context.
- Third , in your virtual environment run the commands mentioned in pypi v0.42 tool
- Fourth, write your main script to use bert model at python notebook.

Install the following dependencies in your virtual environment

- pip install bert-extractive-summarizer
- pip install spacy==2.1.3
- pip install transformers==2.2.2
- pip install neuralcoref

Now, you have finished installing the tools In your virtual environment, type :
- Jupyter notebook
- Go to Lib folder

Name	Date modified	Type	Size
.ipynb_checkpoints	5/5/2020 3:50 PM	File folder	
Include	5/5/2020 2:20 PM	File folder	
Lib	5/5/2020 2:20 PM	File folder	
Scripts	5/5/2020 5:26 PM	File folder	
pip-selfcheck	5/5/2020 2:30 PM	JSON File	1 KB
pyvenv	5/5/2020 2:20 PM	Text Document	1 KB

Directory to insert your Python Notebook

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

- Go to site-packages

| site-packages | 5/6/2020 8:48 PM | File folder |
| tcl8.6 | 5/5/2020 2:20 PM | File folder |

Directory to insert your Python Notebook

- Insert your python notebook at site-packages
- from summarizer import Summarizer
- Here an error where appear trying to tell you that pytorch dependency is missing so write this command : pip3 install torch==1.3.1+cpu torchvision==0.4.2+cpu -f https://download.pytorch.org/whl/torch_stable.html
- body ='''' enter article to be summarized here"
- model = Summarizer()
- result = model(body, min_length=60)
- full = ''.join(result)
- print(full)   // full is the summarized text

Dataset Technicalities:

Note: The mentioned preprocessing steps are done in the process.py file included in the submission. The dataset needed multiple steps to be ready for our task:

- The data set came in a .csv file. We had to extract each paragraph and its summary from the .csv file and ignore any paragraph that does not have a summary.
- The text is formatted as tag @summerizer
- The next step was to process the .txt files to make all the content to be summarized in one paragraph.
- Then, we go through all the files in the directory, open each file and split it to the article and its summary.
- After that the article is now sent to the summarizer.
- Lastly, both the summarizer's output and the original summary are sent to the ROUGE model to be evaluated.

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

How to reproduce our work:

1. Visit this link to download the dataset.
2. Run the process.py file.
3. Upload the dataset files to your Drive.
4. Run the model's script using Google Colab.

7| Team task distribution

.٥

Team Member	Role
Dina Adel	Performance metric , Report writing
Ahmed Ali	Dataset handling , code integration
Alaa Hesham	BERT Model ,Report writing

Figure 8 Team members and task distributions

8 | References

[1] V. Gupta and G. Lehal, JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, NO. 3, vol. 2, no. 3, pp. 104 -116, 2010. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.348.3840&rep=rep1&type=pdf#page=104 .

[2] K. Ježek and J. Steinberger, "Automatic Text Summarization", 2007. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.456.58&rep=rep1&type=pdf .

[3] "CNN / Daily Mail Leaderboard | Papers with Code", Paperswithcode.com, 2020. [Online]. Available: https://paperswithcode.com/sota/document-summarization-on-cnn-daily-mail

[4] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders", 2019. Available: https://www.aclweb.org/anthology/D19-1387.pdf

Zewail City of Science and Technology
University of Science and Technology
CIE 555 - Spring 2020

[5] D. Miller, "Leveraging BERT for Extractive Text Summarization on Lectures", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/1906.04165v1

[6] Murray, G., Renals, S., & Carletta, J. (2005). Extractive summarization of meeting recordings

[7] Van Labeke, N., Whitelock, D., Field, D., Pulman, S., & Richardson, J. T. (2013, July). What is my essay really saying? Using extractive summarization to motivate reflection and redrafting. In AIED Workshops.

[8] V. Balasubramanian, S. Doraisamy and N. Kanakarajan, "A multimodal approach for extracting content descriptive metadata from lecture videos", Journal of Intelligent Information Systems, vol. 46, no. 1, pp. 121-145, 2015. Available: 10.1007/s10844-015-0356-5

[9] S. Garg, "Automatic Text Summarization of Video Lectures Using Subtitles", Advances in Intelligent Systems and Computing, pp. 45-52, 2017. Available: 10.1007/978-981-10-3779-5_7

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners", 2020. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[11] J. Devlin, M. Chang, K. Lee and K. Toutanova, "Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018. Available: https://arxiv.org/abs/1810.04805.

[12] M. Koupaee and W. Yang Wang, "WikiHow: A Large Scale Text Summarization Dataset", 2018. Available: https://arxiv.org/abs/1810.09305

[13]V. Chen, E. Montaño, L. Puzon and D. Chen, "An Examination of the CNN/DailyMail Neural Summarization Task", 2020. Available:

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762104.pdf

[14] Ccs.neu.edu, 2020. [Online]. Available:

http://www.ccs.neu.edu/home/vip/teach/DMcourse/5_topicmodel_summ/notes_slides/What-is-ROUGE.pdf.

[15] "CNN / Daily Mail Leaderboard | Papers with Code", Paperswithcode.com, 2020. [Online]. Available: https://paperswithcode.com/sota/document-summarization-on-cnn-daily-mail

17

[16] "Papers with Code - WikiHow: A Large Scale Text Summarization Dataset", Paperswithcode.com, 2018. [Online]. Available:

https://paperswithcode.com/paper/wikihow-a-large-scale-text-summarization

[17] Y. Liu and M. Lapata, "Text Summarization with Pretrained Encoders", 2019.

[18]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.