Evaluating Extractive Summarization Techniques on News Articles

Conference Paper · July 2021

CITATIONS
0

READS
139

3 authors, including:

Beauty Tasara
Binus University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Nunung Nurul Qomariyah
Binus University
28 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Evaluating Extractive Summarization Techniques on News Articles

Sreeya Reddy Kotrakona Harinatha
Computer Science Department
Faculty of Computing and Media
Bina Nusantara University
Jakarta, Indonesia 11480
kotrakona.reddy@binus.ac.id

Beauty Tatenda Tasara
Computer Science Department
Faculty of Computing and Media
Bina Nusantara University
Jakarta, Indonesia 11480
beauty.tasara@binus.ac.id

Nunung Nurul Qomariyah
Computer Science Department
Faculty of Computing and Media
Bina Nusantara University
Jakarta, Indonesia 11480
nunung.qomariyah@binus.edu

***Abstract*—In recent years, due to the rise of deep learning and natural language processing, text summarization has become a huge topic among scholars. Text summarization derives a shorter coherent version of a longer document. There are two methods of summarization namely, abstractive and extractive. This paper focuses on extractive summarization using TextRank and BERT. These algorithms have been tested under various circumstances to determine the best and they all perform better on certain parameters. The goal of this paper is to determine which algorithm performs better as compared to human generated extractive summaries on news dataset. The same dataset was used for both these algorithms and the summaries were evaluated using ROUGE Score. The result showed that TextRank yielded a better ROUGE score as compared to BERT. TextRank showed higher F-measure and recall while BERT had higher precision.**

***Index Terms*—extractive text summarization, Natural Language Processing, BERT, TextRank, supervised machine learning**

I. Introduction

In today's society, the lives of normal civilians are generally hectic. This is due to the fast-paced nature of this technologically advanced society. Many claim that in today's society, overworking has become the new norm [1]. In this modern yet hectic society, reading has become an essential part of a person's life [2]. However, research shows that for many people, reading long texts can be stressful due to numerous causes, one of them being lack of time.

As people live hectic lives, they often do not have the time to read long texts. This frustrates them as often these long texts are essential for them to read. Therefore, in order to save time, a summarized version of these texts are needed. Obtaining a summarization version of these texts would save time as it is much easier to read and understand shorter texts.

In Natural Language Processing (NLP) domain, there is a automatic process developed by using an Artificial Intelligent (AI) based algorithm, which is called *text summarization*. It is a process of summarizing information in order to obtain a concise summary. The goal behind text summarization is to obtain an understandable and short version of a long text such as articles or reviews.

For this study, we implemented and evaluated two methods for text summarization techniques, namely TextRank algorithm and BERT model. We evaluated both methods on the news articles dataset.

II. Related Work

In this section, we reviewed some related studies in the text summarization research area. In this study we compared two algorithms, TextRank and BERT. The TextRank algorithm was introduced in 2004 by Mihalcea and Tarau [3]. It is an extractive and unsupervised text summarization technique which works by assigning a rank to every individual sentence in the document. This rank determines how important the certain sentence in the entire input text. BERT stands for Bidirectional Encoder Representations from Transformers. It was published by researchers at Google and is based on the Transformer architecture [4]. The key feature of BERT is bidirectional training of transformer architecture which is applied to language modelling. BERT has an attention mechanism which is able to identify the meaning behind words in the input text as shown in Figure 1.

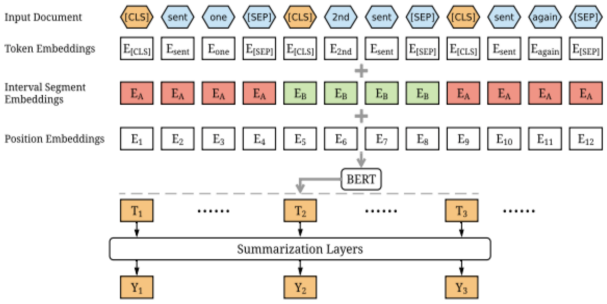

Fig. 1: BERT [5]

In the text summarization domain area, there is one common evaluation metric called ROUGE, which stands for Recall-Oriented Understudy for Gisting Evaluation. This metric has been introduced in a paper entitled "ROGUE : A Package for Automatic Evaluation of Summaries" [6]. In the published article, Lin introduced ROGUE as a method for the automatic evaluation of summaries. It determines the quality of the summary generated by comparing it to the ideal summary

created beforehand. There are four different ROUGE measures i.e. ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. Three of these measures were used in Document Understanding Conference (DUC) 2004 [6].

In a recent study entitled "Automatic Summary Generation using TextRank based Extractive Text Summarization Technique", Upansani et al. [7] discussed about the use of TextRank algorithm to summarize texts in an efficient manner. In the paper, they also stated the advantages of using TextRank algorithm. The advantages include the lower query-time cost of the TextRank algorithm when compared to the other extractive summarization algorithms. In addition to this, TextRank is said more feasible than other algorithms as it does not train the entire model and works intensively on the current data.

Another work entitled "Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization" published by Kumar et al. [8], compared the effectiveness and performances of different extractive summarization algorithms. The algorithms used were LexRank, TextRank and Latent Semantic Analysis (LSA). These methods were chosen as they are quite different approaches to text summarization. TextRank and LexRank follow a structured-based approach while LSA follows a semantic-based approach. They used ROUGE-1 as the metric to evaluate the summaries produced by the algorithm (we will explain this metric later in Section III-D). The experiment results showed that TextRank performed better as it obtained a score of 0.76217. However it is slower than LSA which performs the fastest whilst yielding the lowest score. LexRank, on the other hand, performed the slowest, however it still yielded better results than LSA. Therefore, they conclude that TextRank is the most ideal algorithm compared to the other algorithms.

A paper entitled "Leveraging BERT for Extractive Text Summarization on Lectures" proposed using BERT for summarization [9]. The reason for using BERT was because it is believed that deep learning algorithms would be able to improve the quality of summaries produced. In order to properly evaluate the performance of BERT for summarization, the paper compared BERT with the other common algorithms such as TextRank. The paper showed that BERT was able to capture the meaning behind the text better than TextRank. However, TextRank included more information which was more beneficial to the reader. Nevertheless, BERT was able to summarize the text in a more concise and efficient manner.

In another study entitled "Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2", BERT and OpenAI GPT-2 were used to summarize text [10]. It used ROGUE as the evaluation metric to compare the performance of the model. The paper concluded that extractive models yielded higher ROGUE scores compared to abstractive ones. However, they added that abstractive summaries are often more readable compared to extractive ones. In this study we focus on the extractive models to limit the scope of our work as it is easier for us to evaluate the result by using ROUGE score.

In another study entitled "Looking for a Few Good Metrics: ROUGE and its Evaluation", Lin [11] evaluates the credibility of ROUGE as compared to Document Understanding Conference (DUC) 2001, 2002, and 2003 evaluation data which was made by humans. It was also investigated the validity of the DUC evaluation method and it conclude that it was viable. Therefore, the expected outcome was that a good summary should have a good ROUGE score. The results showed that ROUGE score can be used effectively to evaluate the quality of generated summaries and also it has high correlation with human judgement in multi document summaries.

In a paper entitled "Extractive summarization of clinical trial descriptions", Gulden et al. [12] investigates the best algorithm for the extractive summarization of clinical trial descriptions. They conclude that the TextRank algorithm illustrates the best performance with the evaluation metrics being ROUGE score. The results from this study are in correlation with human judgement on the quality of the summaries. The results of the study also indicated that extractive summarization can be used to generate coherent summaries and ROUGE-L F1 score is the best metric for automated quality checking on the computer generated summaries.

III. Research Method

A. Dataset

For our experiment, we used the news articles dataset scraped from New York Times, CNN, Business Insider and Breitbart as available on Kaggle [1]. The original dataset did not provide any human summaries, it only offered the title of the article, while this could be used as the summary, it is not ideal as the headline title was too short. We generated the label manually by adding the human summary for the available articles. We also added another column called `theme` to the dataset, this column would state the genre of the news articles.

The dataset is ideal for summarization as the provided news articles are long and will consume lots of time to read it. Therefore, it is ideal to generate automatic summarization for the articles in the dataset. The dataset consists of 50,001 rows of data. A large dataset would result in lower estimation variance, thus the predictive ability of the model would improve. However, in this study, we limit the rows used in the experiment by only taking the first 1,000 rows due to the limitation of computing power.

The original dataset contains nine columns as shown in Table I. We added the new columns namely `human_summary` and `theme` as shown in Table II. In the `human_summary` column, we placed the summaries we made, and in the `theme` column we defined the genre of the news articles. The `content` column has been renamed into `articles` for simplicity. Amongst all the columns, the ones we used for this research is `human_summary`, `theme`, and `content`.

Afterwards, a *catplot* was made to show the general theme of the articles in the dataset.

As seen from Figure 2, most of the news articles discussed about politics or business. In addition to that, barely any

[1] https://www.kaggle.com/snapcrack/all-the-news?select=articles1.csv

TABLE I: Samples Dataset in Its Original Columns

	id	title	publication	author	date	year	month	url	content
0	17283	House ...	New Yo...	Carl H...	2016-1...	2016	12	NaN	WASHIN...
1	17284	Rift B...	New Yo...	Benjam...	2017-0...	2017	6	NaN	After ...
2	17285	Tyrus ...	New Yo...	Margal...	2017-0...	2017	1	NaN	When W...
3	17286	Among ...	New Yo...	Willia...	2017-0...	2017	4	NaN	Death ...
4	17287	Kim Jo...	New Yo...	Choe S...	2017-0...	2017	1	NaN	SEOUL,...

TABLE II: Samples Dataset with Two Additional Columns

	id	human_summary	theme
0	17283	In successfully ...	politics
1	17284	Officers put her...	crime
2	17285	The film strikin...	entertainment
3	17286	The year was onl...	entertainment
4	17287	If North Korea c...	politics

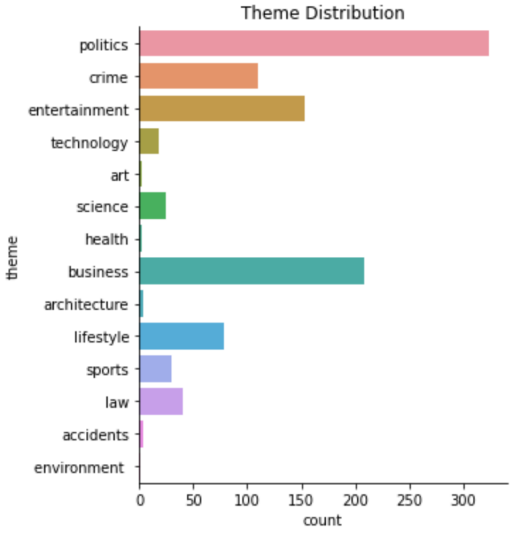

Fig. 2: Dataset Article Count based on Theme Category

articles talked about health, art, architecture, accidents or environment.

We also created a histogram to determine the average word count of the data in the `human_summary` and `articles` columns as shown in Figure 3.

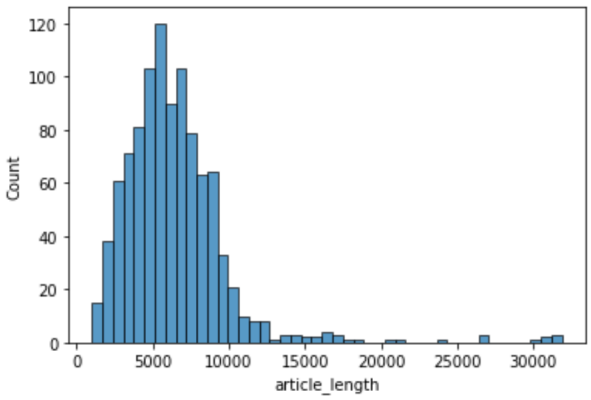

Fig. 3: Article Length

This histogram shows that the length of 120 articles were around 5,000. Most of the articles fell under the range of 5,000 words to 7,000 words. Only a few articles were more than 10,000 words.

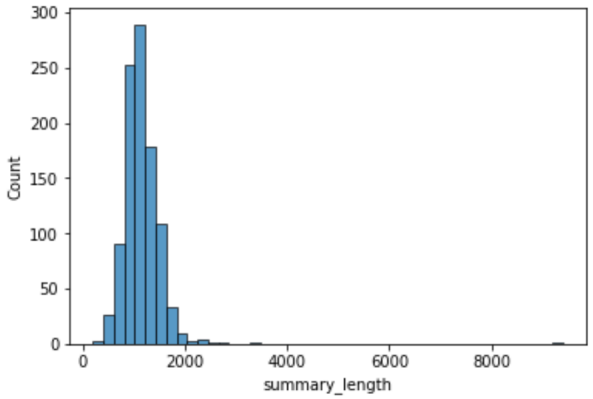

Fig. 4: Summary Length

This histogram in Figure 4 shows that the length of the summaries fell under the range of less than 2,000. Only a few summaries exceeded that length. We also tested if the length of the articles were related to the length of the summaries by fitting a linear regression line as shown in Figure 5.

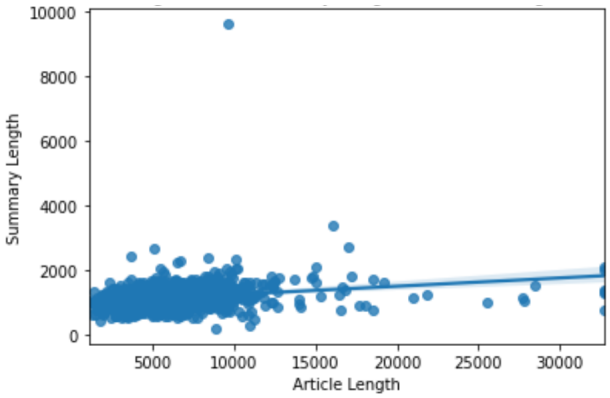

Fig. 5: Linear Regression

There is barely any relationship between the length of the `human_summary` and `content` as even though the r-squared was positive, it is too close to 0 to be considered relevant. It is only 9% (the model's r-squared is 0.093).

B. Data Preparation and Processing

Data preparation or data cleaning as it is sometimes referred to, is one of the most crucial parts in any data science problem. Data is often extremely messy, it may contain columns that

TABLE III: Data Cleaning Steps

Steps	Description

1	Remove the unnecessary columns which are:
	`id`
	`publication`
	`author`
	`date`
	`year`
	`month`
	
	This leaves us with only the the following columns
	`human_summary`
	`articles theme`

2	Remove missing data if present
3	Expand contractions
4	Remove special characters such as "■" and words in brackets
5	Remove all punctuations
6	Remove excess whitespace
7	Remove stopwords

are not necessary. Additionally, it could also have rows with incomplete data which will not be useful. Thus, action must be taken in order to clean up these data.

TextRank and BERT are both unsupervised machine learning algorithms, thus it does not require labelled data. The basic steps to prepare the data are shown in Table III.

We have explained the first step in detail in the previous section. For the second step, we made use of the `isnull.sum()` function in Python. This function checks if there is any missing data in our dataset. However, we found that the data is complete, and no missing value found.

In the third step, we expanded the contractions as we wanted to standardize the data and simplify it. Contractions increase the dimensionality of the document-term matrix as there would be separate columns for the words "I" and "I'll". We expanded the contractions by defining a dictionary which shows the contractions and their expanded form. If the dataset contained any of the contractions, it would expand it to its expanded form. The contractions we expanded are shown in Figure 6.

```
[8] contraction_expander = {
    "ain't": "am not", "aren't": "are not", "can't": "cannot", "can't've": "cannot have", "'cause": "because",
    "could've": "could have", "couldn't": "could not","couldn't've": "could not have", "didn't": "did not",
    "doesn't": "does not", "don't": "do not", "hadn't": "had not", "hadn't've": "had not have", "hasn't": "has not",
    "haven't": "have not", "he'd": "he would", "he'd've": "he would have","he'll": "he will", "he's": "he is",
    "how'd": "how did", "how'll": "how will","how's": "how is","i'd": "i would","i'll": "i will","i'm": "i am",
    "i've": "i have", "isn't": "is not","it'd": "it would","it'll": "it will","it's": "it is","let's": "let us",
    "ma'am": "madam","mayn't": "may not","might've": "might have","mightn't": "might not","must've":
    "must have","mustn't": "must not","needn't": "need not","oughtn't": "ought not","shan't": "shall not",
    "sha'n't": "shall not","she'd": "she would", "she'll": "she will","she's": "she is","should've": "should have",
    "shouldn't": "should not","that'd": "that would", "that's": "that is", "there'd": "there had","there's": "there is",
    "they'd": "they would","they'll": "they will", "they're": "they are","they've": "they have","wasn't": "was not",
    "we'd": "we would", "we'll": "we will","we're": "we are","we've": "we have","weren't": "were not",
    "what'll": "what will","what're": "what are","what's": "what is", "what've": "what have","where'd": "where did",
    "where's": "where is","who'll": "who will","who's": "who is", "won't": "will not","wouldn't": "would not",
    "you'd": "you would","you'll": "you will","you're": "you are"
    }
```

Fig. 6: Contractions

On the fourth step, we noticed while summary the articles that the dataset contained special characters such as "■ ". Thus, we fixed this by replacing those special characters with a blank space. Additionally, we also noticed there were lots of unnecessary words inside brackets. Thus we used a regex function to remove those words.

On the fifth step, we removed the punctuations by defining all the punctuations we wanted to remove. If any of those punctuation marks were present in the dataset, it would replace them with a blank space.

On the seventh step, we removed all the stopwords located in the dataset. Stopwords are words that are most common in a language and do not have much value in understanding the meaning of a document. For example, in the sentence "There is a cat on the roof", the words "is", "a", "the" and "on" does not add any meaning to the sentence. On the other hand, the words "there", "cat" and "roof" are the key to understanding what the sentence means. Therefore, only those words are necessary.

C. Model and Techniques

There are two techniques used to summarize texts, extractive summarization and abstractive summarization [13]. The concept behind extractive summarization can be compared to a highlighter. It involves highlighting the most crucial lines or phrases in the text. Afterwards, the final result is the summary which consists of the compilation of the important message.

Extractive summarization is also considered can produce more reliable results compared to abstractive summarization. Unlike extractive summarization, the concept of abstractive summarization can be compared to a pen instead of a highlighter. In abstractive summarization, the summary output can contain words that did not exist in the original text. Abstractive summarization is considered to be more inline to how humans summarize text as it is able to rewrite the entire text. However, this research focuses on extractive summarization instead of abstractive summarization.

1) TextRank: TextRank is an algorithm that is derived from the well-known PageRank algorithm [3]. PageRank is an algorithm that is used by Google Search in order to rank the web pages that appear when a user uses it to search for any information. TextRank is similar to PageRank as both algorithms are graph based, however they have their differences. Unlike the graph in PageRank, the graph in TextRank is undirected and the weight of each edge is different.

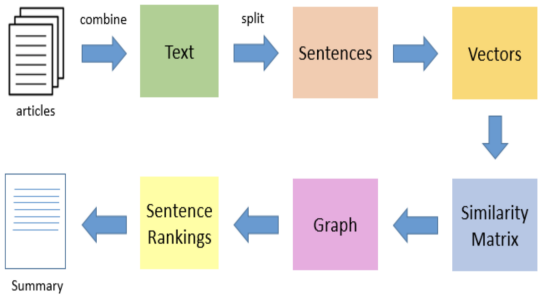

Fig. 7: TextRank

In TextRank technique, the text would first be gathered then split into individual sentences. Afterwards, the similarity between each sentence would be calculated and then stored into a matrix, otherwise known as the similarity matrix. Afterwards, this matrix would then be converted into a graph, the calculated similarity score would be the edges and the sentences itself would be the vertices. Finally, the algorithm

selects the highest ranked sentence or sentences to create the summary.

2) Word Embeddings: In this experiment, we used GloVe word embeddings [14] to capture global and local statistics of a corpus to come up with word vectors. GloVe is built on the idea that it is possible to deduce the semantic relationships between words from the co-occurrence matrix. A co-occurrence matrix is a matrix that shows how often a particular pair of words occur together. The rows of the matrix are the words that occur in the corpus whilst the columns are how frequently we see the word in the corpus.

For example, in the sentence "I play cricket, I love cricket, I love cricket", the co-occurrence matrix would look like the matrix in the table below.

TABLE IV: Co-occurrence Matrix Example

	play	love	football	I	cricket
play	0.0	0.0	0.0	1.0	1.0
love	0.0	0.0	1.0	2.0	1.0
football	0.0	1.0	0.0	0.0	0.0
I	1.0	2.0	0.0	0.0	0.0
cricket	1.0	1.0	0.0	0.0	0.0

Through this matrix, we are able to find the probability of a pair of words. For example, focusing on the word cricket, we can count $P(\frac{cricket}{play}) = 1$ and $P(\frac{cricket}{love}) = 0.5$. Therefore the ratio of the probabilities is $\frac{1}{0.5} = 2$.

The purpose behind using GloVe word embeddings is to reduce computational time. A corpus could have millions of words. Therefore, the time needed to train it could be extremely long. However, with GloVe's pre-trained word vectors, the time needed would shorten as it is already trained.

3) BERT: In the BERT model, there is an inbuilt extractive text summarizer. The main goal behind this is to summarize text. The name of the module is *summarizer*, it takes the articles as the input and then provides the summary in an efficient manner. This research focuses on BERT's inbuilt summarizer module.

D. Evaluation Method

The evaluation method used in our experiment was ROUGE which stands for Recall-Oriented Understudy for Gisting Evaluation [6]. It is a set of metrics that is used to evaluate summaries produced by machines by comparing with a set of predetermined summaries. It measures *recall* or how much the words proportions (and or n-grams) in the human reference summaries are captured in the machine generated summaries. If there are many words from the human references appearing in the system, then the ROUGE score will be high. Therefore, the larger the ROUGE score the more accurate the model.

There are different types of ROUGE measures such as ROUGE-N and ROUGE-L. ROUGE-N measures the unigram, bigram, trigram and higher order of n-gram overlap. ROUGE-L measures the longest matching sequence of using Longest Common Subsequence (LCS). For this research, we utilized the ROUGE-N measure, more specially, ROUGE-1 which measures unigram due to the size of our dataset that was not large. Thus, ROUGE-1 is a sufficient metric.

There are three metrics used to be considered in ROUGE score, namely precision, recall and F-measure. We use these metrics to evaluate our models.

1) Recall
Recall measures how much of the manmade summary is used by the computer-generated summary. It can be calculated by using the following formula:

$$\text{Recall} = \frac{\text{number of overlapping words}}{\text{total words in reference summary}} \quad (1)$$

2) Precision
Precision measures how much of the computer-generated summary is necessary or relevant. It can be counted by using the following formula:

$$\text{Precision} = \frac{\text{number of overlapping words}}{\text{total words in predicted summary}} \quad (2)$$

3) F-measure
F-Measure, also known as F1-Score, is the measure of a model's accuracy on a dataset. It combines both recall and precision and is considered as the harmonic mean of the model's precision and recall. It can be counted by using the following formula:

$$\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

ROUGE is one of the most popular metrics for measuring the performance of text summarization models [6]. However, it also has flaws which affects the quality of measuring the performance. The essence of ROUGE is that it checks if the words generated by the computer overlaps with the words of the man-made summary. In theory, this seems like an ideal way to measure the accuracy of the summary of the text the computer generated. However, the first problem of this is that man-made summaries tend to be abstractive in nature whereas the summary generated by the model would be extractive. This summary generated by computer would not include any new words, so there is a good possibility that the ROUGE score calculated would not be that high.

Furthermore, another problem of ROUGE is that it is unable to measure fluency. As stated previously, ROUGE measures if the words generated by the computer overlaps with the words of the man-made summary. Therefore, the summary generated could have all the words the man-made summary has but presented in an incoherent order. For example, the man-made summary could be "cat is on the roof" and the generated summary is "roof cat the on is". ROUGE would give this summary a perfect score, however, to humans, it is clear that this is not an ideal summary.

IV. Results and Discussion

A. Stopwords

In our experiment, we used Google Colab to test and evaluate TextRank and BERT. The processor used by Google Colab by the time this paper was written was Intel(R) Xeon(R)

CPU @ 2.20GHz, with the available GPU (Nvidia Tesla T4) RAM is 15GB and the memory that we can use was 13GB. As stated in the data preparation and processing section, we removed all the stopwords presented in the dataset. The stopword removal helped make our data less complex for the model to understand. However, we observed that removing the stopwords made the computer-generated summary of the model worsen, eventhough it run a little bit faster. The two paragraphs below show the difference between the summaries generated when the stop words were removed and when they were kept:

- Summary generated after removing stopwords

```
mr. trump private club florida spent past two weeks
away home new york city likely eclipse 45th  president
winter white house. mr. trump given cadre white
house reporters cover access club  grudgingly eluded
reporters covering saturday slipping away without
warning play golf another  clubs nearby jupiter. robin
bernstein club member nearly 25 years said club members
might  express frustration thought important keep
donald family safe. like aspects mr. trump business
 interests party generated controversy tickets made
available club members guests little 500. mr.  trump
aides rejected questions. mr. trump frequently dines
patio central point action night singer  plays small
band sometimes belting requests mr. trump guests.
```

- Summary generated without removing stopwords

```
mr. trump has given the cadre of white house reporters
who now cover him some access to the  club but
grudgingly so he once again eluded the reporters
covering him on saturday slipping  away without any
warning to play golf at another of his clubs nearby
in jupiter. but mr. trump  private club in florida
where he has spent the past two weeks away from his
home in new york  city is likely to eclipse them all
as the 45th president winter white house. mr. trump
later delivered  remarks according to a guest who said
he thanked his family and the club members for their
support over the years.like most aspects of mr. trump
business interests the party generated  controversy
as tickets to it were made available to club members
and guests for a little more than  500. mr. trump
aides rejected the questions.and that was always the
intention of marjorie  meriweather post the cereal
heiress and the property original owner who left to
the federal  government when she died in 1973 hoping it
would serve as a home for presidents.
```

TABLE V: Experiment with Stopwords

Measurement	Removed Stopwords	Kept Stopwords
Exec. time (mins)	11.5	12.8
Precision	0.4825	0.6125
Recall	0.3229	0.5978
F-Measure	0.3841	0.6004

In addition to that, Table V and Figure 8 show how deleting stopwords affected the accuracy of the TextRank model. The model performs better when the stopwords were not removed, as can be seen by their ROUGE scores in the table and Figure 8. The execution time needed by the model increases if the stopwords were kept, however, the difference was not significant. Therefore, it would be ideal to keep the stopwords for the training instead of removing them. Stopwords are generally removed because it can help reduce computational time of the model's training. However, in the text summarization, removing these stopwords could negatively impact the quality

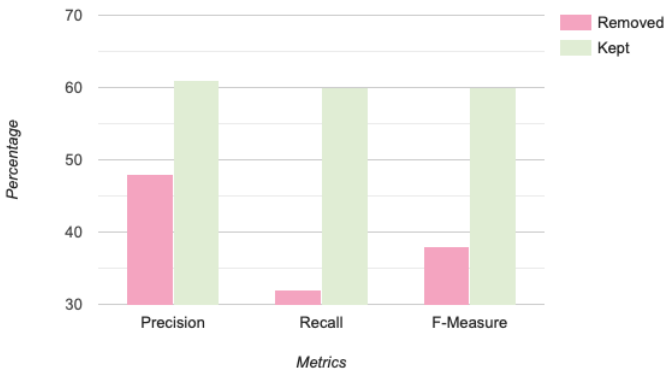

Fig. 8: Stopwords

of the summary produced as shown above. Removing the stopwords reduces the coherency of the generated summary.

B. TextRank and BERT

This section shows the information obtained during this research. From our experiment, we found that the number of rows in the dataset did not greatly affect the performance of the model as shown in Table VI. However, it greatly affected the execution time as shown in Figure 9.

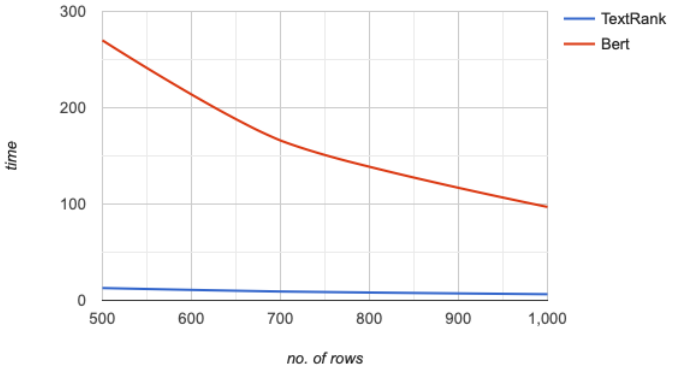

Fig. 9: TextRank and BERT Time (in second)

The computation time for TextRank was significantly lower than the computational time for BERT. For 1000 rows, BERT took 4.5 hours to compute whereas TextRank accomplished it in 12.8 minutes. The same trend follows regardless of the number of rows.

In terms of performance, we have evidence that TextRank performed better than BERT as shown visually in Figure 10. It shows that BERT has a better precision value compared to TextRank. However, TextRank has better Recall value as well as F-Measure value compared to BERT. Therefore, TextRank yielded a better ROUGE score compared to BERT.

V. Conclusion and Future Work

This paper showed the evaluation of the two methods in extractive summarization, TextRank and BERT, to predict the summary of the news articles. The result of our experiment

TABLE VI: TextRank and BERT Results

Measurement	TextRank			BERT		
	1000 rows	700 rows	500 rows	1000 rows	700 rows	500 rows
Execution time (mins)	12.8	9.2	6.4	270	166	97
Precision	0.6125	0.601	0.5911	0.5712	0.5623	0.5551
Recall	0.5978	0.5778	0.5444	0.2296	0.2236	0.2279
F-Measure	0.6004	0.5892	0.5668	0.3276	0.3066	0.3087

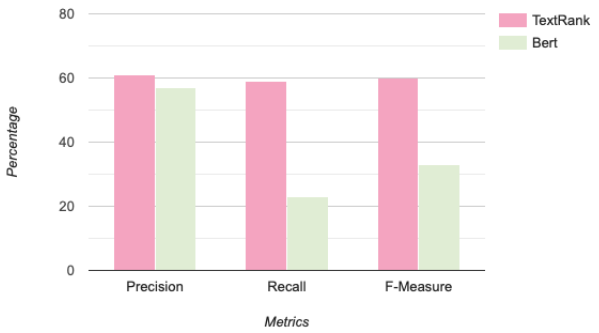

Fig. 10: TextRank and BERT Performance

showed that TextRank yielded a better ROUGE score compared to BERT. The BERT model had better precision, however TextRank had a better recall and F-measure. Additionally, in terms of the computational complexity, TextRank did not take as much time as BERT to process the data. For a thousand rows, TextRank only needed 12.8 minutes whereas BERT required 240 minutes (about 19 times longer). The difference in achitecture between Textrank and Bert results in Textrank performing better in terms of speed under limited resources due to its lightweight. Additionally, the result of our experiment suggest that the stopwords should not be removed as it reduces the accuracy and restricts the fluency of the computer-generated summary. Further research is needed in order to improve this study. A bigger dataset should be used so that there will be more training, improving the model's accuracy. LSTM and RNNs could be used to improve the model as this would create abstract summaries which are more in line with how humans summarize articles. This study provides a general NLP pipeline to perform text summarization for English language. For the future work, we plan to implement the same pipeline for other language, i.e. Bahasa.

Supplementary Material

The implementation of the algorithms in Python and the pre-processed dataset are available at https://github.com/reddzzz/DataScience_FP under GNU General Public License v3.0. Please cite this paper to reuse the dataset and/or modify the code.

References

[1] G. Belli. (2017) Most american workers are stressed most of the time. [Online]. Available: https://www.cnbc.com/2017/03/29/most-american-workers-are-stressed-most-of-the-time.html

[2] N. Dunlevy. (2013) Pinpointing signs and causes of reading-related stress. [Online]. Available: http://evancedkids.com/pinpointing-signs-and-causes-of-reading-related-stress/

[3] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," vol. 1, 2019.

[5] R. Horev. (2018, Nov) Bert explained: State of the art language model for nlp. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[6] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," 2017.

[7] D. S. J. A. J. A. Upansani A, Amin N, "Automatic summary generation using textrank based extractive tear summarization technique," *International Research Journal of Engineering and Technology*, vol. 7, 2020.

[8] S. S. K. S. Kumar A, Sharma A, "Performance analysis of keyword extraction algorithms assessing extractive text summarization," *International Research Journal of Engineering and Technology*, 2017.

[9] D. Miller, "Leveraging bert for extractive text summarization on lectures."

[10] Y. N. Bowen Tan, Virapat Kieuvongngam, "Automatic text summarization of covid-19 medical research articles using bert and gpt-2," June 2020.

[11] C.-Y. Lin, "Looking for a few good metrics: Rouge and its evaluation," 2004.

[12] C. Gulden, M. Kirchner, C. Schüttler, M. Hinderer, M. Kampf, H.-U. Prokosch, and D. Toddenroth, "Extractive summarization of clinical trial descriptions," *International journal of medical informatics*, vol. 129, pp. 114–121, 2019.

[13] M. Maybury, *Advances in automatic text summarization*. MIT press, 1999.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.