Extractive Text Summarization Using Tf-Idf Model

Mohammad Nafees Bin Zaman
Dept. Computer Science
International Islamic University
Gombak, Selangor, Malaysia
zamannafees@gmail.com

Abid Ebna Saif Utsha
Dept. Computer Science
International Islamic University
Gombak, Selangor, Malaysia
abidebnasaifutsha@gmail.com

Mahfuzealahi Noman
Dept. Computer Science
International Islamic University
Gombak, Selangor, Malaysia
noman.alahi@gmail.com

Ahmad
Dept. Computer Science
International Islamic University
Gombak, Selangor, Malaysia
a.schigdar@gmail.com

ABSTRACT

With the expanding of online data and plan of action writings, text summarization has turned into a fundamental and increasingly most favorite space to save and demonstrate the primary reason for textual information. It is a challenging task for human to summarize manually substantial documents of text. Text summarization is the process of spontaneously creating and condensing form of a given record and safeguarding its information source into a shorter adaptation with by and large importance. Nowadays text summarization is one of the most favorite research territories in natural language processing and could attracted more attention of NLP researchers. Our research paper aims to summarize from large articles to give reader a clear vision about that particular article. Afterwards a review has been operated on some of the summarization approaches and their important parameters for extracting predominant sentences, distinguished the primary phases of the abridging procedure, and the most significant extraction criteria are presented. Finally, the most fundamental proposed evaluation methods are considered.

KEYWORDS

Tf-Idf, Reuters, TS, Sentence Position, Top-down approach, Extractive summary, Rouge score, Bleu score.

I. RELATED WORK

Text summarization targets on getting the core meaning of documents. Text summarization procedures are classified as Extractive and Abstractive. Extractive summaries generate a set of most important sentences from a text document whereas Abstract summaries seek the coherence among sentences [1]. At present, many scholars have proposed their own algorithms on the application to text classification. In our work as we are trying to generate extractive summaries, we follow a lot of different techniques such as tf-idf, sentence scoring, using corpus etc.

The TF-IDF algorithm is a statistical method used to assess the importance of a word for a document or a category in a file set or corpora. The primary idea is that if a word or expression shows up frequently in an article and it is once in a while found in different articles, it is viewed as that the word or expression has a decent class qualification capacity and is reasonable for characterization. The TF-IDF algorithm was first proposed by Salton [2].

Sentence scoring methods for automatic extractive text summarization algorithms relies upon the sort of text one needs to abridge, the length of archives, the sort of language utilized, and their structure [3]. Different combinations of sentence scoring algorithms yield various outcomes both in the nature of the synopses acquired and the time elapsed in creating them. In this paper we try to implement traditional tf-idf model to summarize substantial documents using sentence scoring.

II. TECHNICAL BACKGROUND

2.1 To make our summarizer we use NLTK (Natural Language Tool Kit) libraries. For substantial documents, we retrieve articles from nltk corpus namely Reuters.

2.2 **Reuters Corpus:** The Reuters Corpus contains 10,788 news documents totaling 1.3 million words. The documents have been classified into 90 topics. Unlike the other Corpus, categories in the Reuters corpus overlap with each other, simply because a news story often covers multiple topics. We can ask for the topics covered by one or more documents, or for the documents included in one or more categories. For convenience, the corpus methods accept a single field or a list of fields.

2.3 **Position of Sentence:** To determine the score of position of a sentence we use a built-in method that is already available in the web. We borrow the value from that method which is retrieved from https://github.com/xiaoxu193/PyTeaser. This method provides us values between 0 and 1 corresponding to sentence's position in the article.

2.4 **Tf-Idf:** Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = $\log_e$ (Total number of documents / Number of documents with term t in it).

For summarization we apply this technique to get more sophisticated arrangement of sentences and accurate results.

2.5 **Evaluation:** To analyze our model, we use two methods naming Bleu Score and Rouge score.

BLEU (**bilingual evaluation understudy**) is an algorithm for evaluating the quality of text which has been machine translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU. BLEU was one of the first metrics to claim a high correlation with human judgements of quality and remains one of the most popular automated and inexpensive metrics.

ROUGE, or **Recall-Oriented Understudy for Gisting Evaluation**, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. In general:

Bleu measures precision: how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries.

Rouge measures recall: how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries.

Naturally - these results are complementing, as is often the case in precision vs recall. If there are many words from the system results appearing in the human references it will have high Bleu, and if there are many words from the human references appearing in the system results it will have high Rouge.

However, there are also available sub-sections of Rouge score which helps to measure the machine generated summaries. Those are

i) Rouge-N
ii) Rouge-L
iii) Rouge-S

- **ROUGE-N** – measures unigram, bigram, trigram and higher order n-gram overlap.
- **ROUGE-L** – measures longest matching sequence of words using LCS. An advantage of using LCS is that it does not require consecutive matches, but in-sequence matches that reflect sentence level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length.

- **ROUGE-S** – Is any pair of word in a sentence in order, allowing for arbitrary gaps. This can also be called skip-gram cooccurrence. For example, skip-bigram measures the overlap of word pairs that can have a maximum of two gaps in between words.

III. USED APPROACH

In our summarizer we use some approaches and key things such as tf-idf, position of sentence, relative corpus etc. We make our summarizer using top down approach.

A top-down approach (also known as stepwise design and in some cases used as a synonym of decomposition) is essentially the breaking down of a system to gain insight into its compositional sub-systems in a reverse engineering fashion. In a top-down approach an overview of the system is formulated, specifying, but not detailing, any first-level subsystems. Each subsystem is then refined in yet greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements. Top down approach starts with the big picture. It breaks down from there into smaller segments. In our case, we use this same approach of divide and conquer.

We take an article and split it into sentences and words. Then we measure some impact factors and score them. Based on that scores our system summarizes a whole document. That shows the divide and conquer method in our work.

Besides, Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification. Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance of tf-idf increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. For these reasons we apply tf-idf in our system.

IV. METHODOLOGY

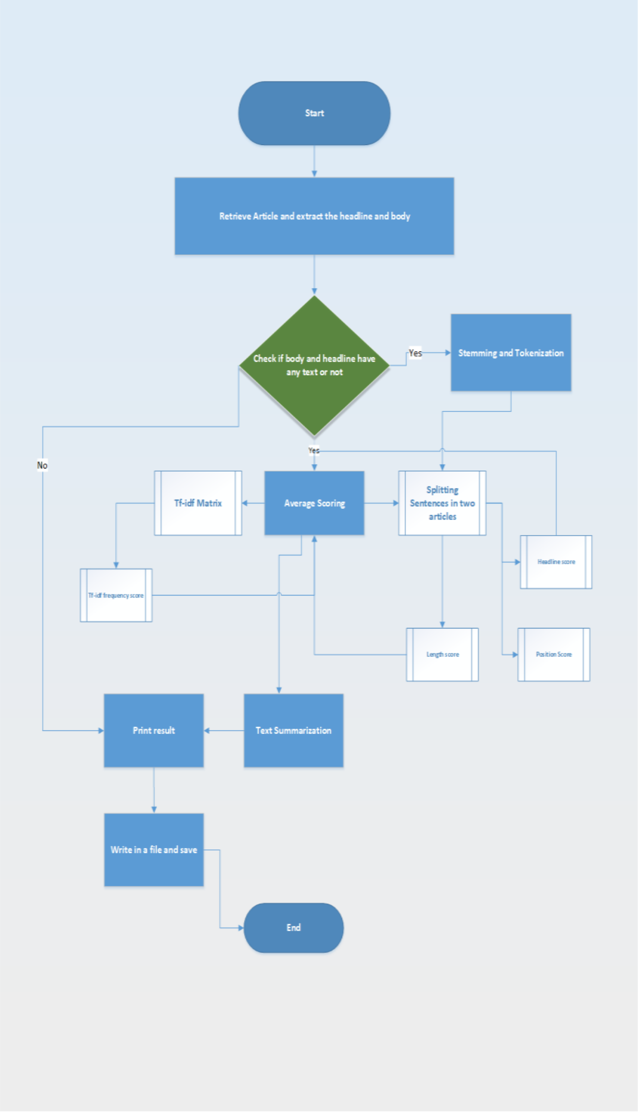


Figure: 1

To demonstrate our methodology, we use this flowchart. So, we start our work retrieving articles and after that extract headline and body from the document. Secondly, we check if there any headline or exist or not more than 5 sentences. If its below than 5 sentences the system will print the whole text as it is. If it's more than five sentences it processes two things. First one is stemming and tokenization. And the other one is average scoring means it calculates the total score and give an average score. Inside these two processes there are also some sub processes that our system considers. Those are tf-idf model, Splitting sentences, Length score, Sentence position etc. After all of these our system summarizes the whole document (make five sentences) and print the result and save in a file. Then we evaluate our system using Rogue Score and Bleu Score. Exact workflow of our process given below.

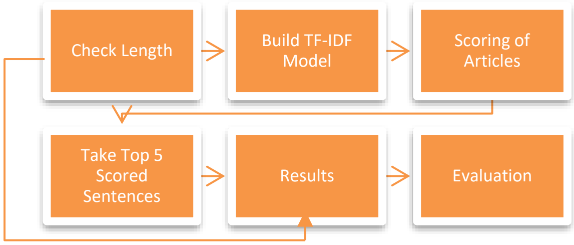


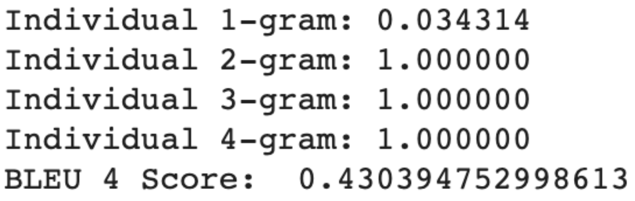
Figure 2

V. Result Analysis

To evaluate our results, we compare our summary with other summary results. We use two websites to generate summaries namely:

i) http://textsummarization.net/text-summarizer

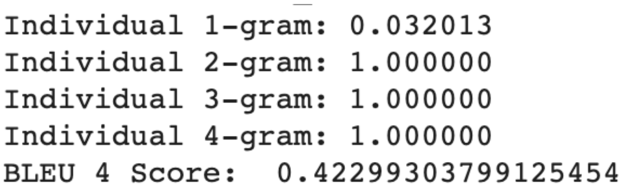
ii) https://www.tools4noobs.com/summarize/



```
Individual 1-gram: 0.034314
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.430394752998613
```

Figure: 3

In figure 2, it shows the bleu score for our summary. For the 2-gram, 3-gram and 4-gram our summarizer gives an outstanding output of exact 1 that means the accuracy is the same whereas in individual 1-gram score is very low compare to others. Consequently, the final outcome of Bleu scores results 0.4303. This happens because of the website uses machine learning algorithm in their text summarizer.

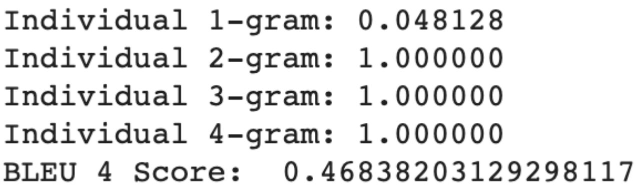


```
Individual 1-gram: 0.032013
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.42299303799125454
```

Figure: 4

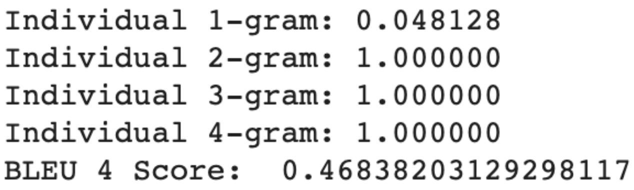
In figure 3, same as figure 2 the Bleu score results barely 0.423. Although 2-gram, 3-gram and 4-gram gives expected output, absence of machine learning algorithm in our summarizer results in overall average scoring.

So, we try to change the length of the summary to 3 and 8.



```
Individual 1-gram: 0.048128
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.46838203129298117
```

Figure: 5

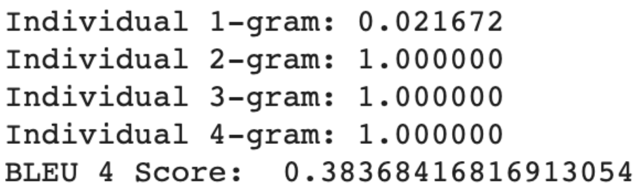


```
Individual 1-gram: 0.048128
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.46838203129298117
```

Figure: 6

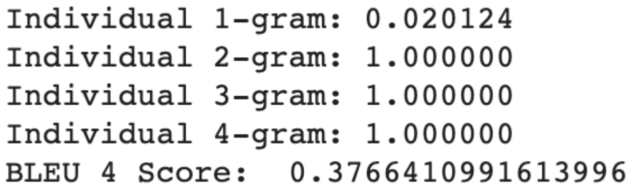
Both the articles have the same result for BLEU score when the summary length is 3 (Figure 5 & Figure 6)

Furthermore, we get the result of summary length of 8 in Figure 7 & Figure 8 which is slightly lesser than the previous. One thing in common of these phases is that only the individual 1-gram differs from each other whereas the rest are the same.



```
Individual 1-gram: 0.021672
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.38368416816913054
```

Figure: 7



```
Individual 1-gram: 0.020124
Individual 2-gram: 1.000000
Individual 3-gram: 1.000000
Individual 4-gram: 1.000000
BLEU 4 Score:  0.3766410991613996
```

Figure: 8

For Rouge Score we use Rouge-1 for unigram, Rouge-2 for bigram and Rouge-L for longest sequence of matching words.

ROUGE-1			
	F-1 Score	Precision	Recall
Web 1	0.573	0.468	0.739
Web 2	0.420	0.273	0.905

Table: 1.1

ROUGE-2			
	F-1 Score	Precision	Recall
Web 1	0.50	0.372	0.60
Web 2	0.330	0.202	0.905

Table: 1.2

ROUGE-L			
	F-1 Score	Precision	Recall
Web 1	0.514	0.460	0.727
Web 2	0.290	0.273	0.905

Table: 1.3

From the above, three tables show an average result as the F-1 score fails to score expected result which is below 60%. As F-1 score is the combination of precision and recall thus these two give also an average score. Precision is number of overlapping words divided by total words in system summary. Recall is number of overlapping words divided by total words in reference summary.

ROUGE-1			
	F-1 Score	Precision	Recall
Web 1	0.530	0.458	0.628
Web 2	0.398	0.264	0.828

Table: 2.1

ROUGE-2			
	F-1 Score	Precision	Recall
Web 1	0.384	0.331	0.459
Web 2	0.276	0.196	0.828

Table: 2.2

ROUGE-L			
	F-1 Score	Precision	Recall
Web 1	0.494	0.448	0.614
Web 2	0.279	0.264	0.828

Table: 2.3

In table 2.1, 2.2 and 2.3 we get the result of Rouge score where the summary length of those articles is 3.

ROUGE-1			
	F-1 Score	Precision	Recall
Web 1	0.443	0.313	0.761
Web 2	0.372	0.226	0.878

Table: 3.1

ROUGE-2			
	F-1 Score	Precision	Recall
Web 1	0.332	0.230	0.598
Web 2	0.311	0.154	0.878

Table: 3.2

ROUGE-L			
	F-1 Score	Precision	Recall
Web 1	0.332	0.303	0.738
Web 2	0.287	0.202	0.878

Table: 3.3

The results of table 3.1, 3.2 and 3.3 are the Rouge score of our summary when the length of summary is 8.

VI. Evaluation

In order to evaluate our model, we plan to execute some comparative bar graphs and line graphs.

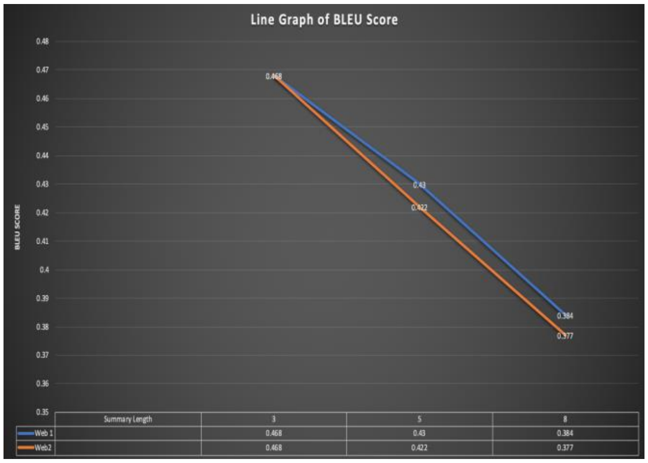
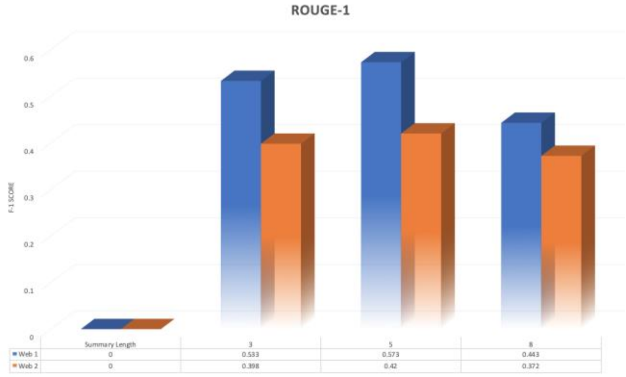


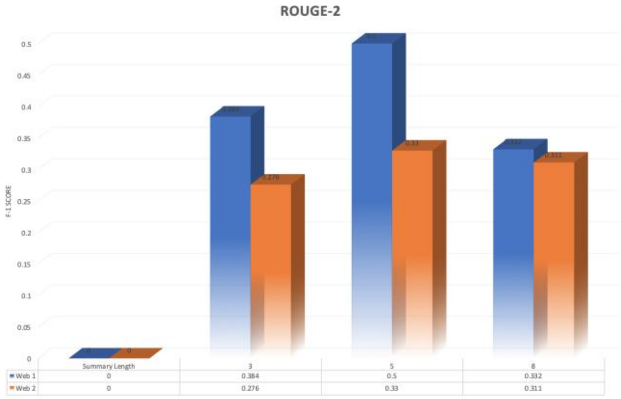
Figure: 9

We know that, the higher the BLEU score the better model is. Our model performs well when the summary length is 3. Both Web 1 and Web 2 have a higher value compare to summary length 5 and 8 shows in figure 9.



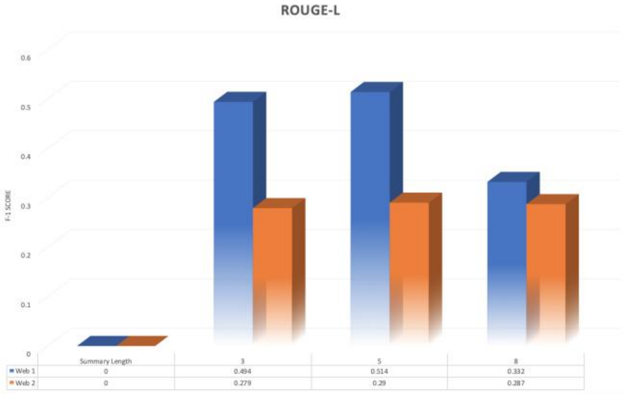
	Summary Length	3	5	8
Web 1	0	0.533	0.573	0.443
Web 2	0	0.398	0.42	0.372

Figure: 10



	Summary Length	3	5	8
Web 1	0	0.384	0.5	0.332
Web 2	0	0.276	0.33	0.311

Figure: 11



	Summary Length	3	5	8
Web 1	0	0.494	0.514	0.332
Web 2	0	0.279	0.29	0.287

Figure: 12

In Figure 10, 11 & 12 it is clearly seen that for both Web 1 and Web 2, summary length 5's bar at the peak. Although summary length 5 scores most in all Rouge Scores, summary length 3 reaches almost same as 5 compare to summary length 8.

VII. Conclusion

According to above analysis of BLEU Scores and Rouge Scores we approve our model best for summary length three. However, in the Rouge scores summary length five is a bit ahead of summary length three. Thus, we can consider our model for five line summary though it would not be that much sophisticated. In future, we will try to implement machine learning algorithm in our model to upgrade our model as well as performance.

Acknowledgement

We would like to thank Dr. Suriani Binti Sulaiman for her guiding in this project.

References

[1] Kågebäck, M., Mogren, O., Tahmasebi, N., & Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)* (pp. 31-39).

[2] Liu, C. Z., Sheng, Y. X., Wei, Z. Q., & Yang, Y. Q. (2018, August). Research of Text Classification Based on Improved TF-IDF Algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)* (pp. 218-222). IEEE.

[3] Ferreira, R., Freitas, F., de Souza Cabral, L., Lins, R. D., Lima, R., França, G., ... & Favaro, L. (2014, April). A context based text summarization system. In *2014 11th IAPR International Workshop on Document Analysis Systems* (pp. 66-70). IEEE.

[4] Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, *7*(1), 61-70.

[5] Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.