Document Summarization using NLP Techniques

LOKESWARI UMAKANTHAN
Erik Jonsson School of Engineering
University of Texas at Dallas
Richardson TX, USA
lxu190000@utdallas.edu

ROHAN VANNALA
Erik Jonsson School of Engineering
University of Texas at Dallas
Richardson TX, USA
rxv190003@utdallas.edu

REETISH CHAND GUNTAKAL PATIL
Erik Jonsson School of Engineering
University of Texas at Dallas
Richardson TX, USA
rxg190006@utdallas.edu

***Abstract*—Big Data is the combination of the 3V's, Variety, Volume and Velocity and with increase in all these aspects from different sources, it is becoming difficult to analyze the data and produce useful results which is concise and summarized. Manual summarization is also a long and difficult process for such data, and it is not efficient which is why Text Summarization has come into light and is gaining more popularity. It is becoming one of the main research fields. It is a process of choosing a set of sentences from the original text such that the selected sentences closely represent the original text without loss in meaning. It is about choosing sentences that preserve the semantic meaning of the information in the original text. There are different types of text summarization techniques but almost all of them fall into two most fundamental categories - extractive and abstractive. In this paper, we briefly describe the two categories mentioned of text summarization, and then describe the text summarization by text rank algorithm that we have implemented and lastly discuss the results obtained and analyze the results achieved.**

Keywords—text summarization, text rank algorithm, abstract method, extractive method, TF-IDF, summary, ROUGE, page rank, analysis.

I. Introduction

Summarization is the process of extracting a concise form of original information where the semantics of the text is preserved but with less sentences. This process gives out a result called the 'summary'. Reading the summary decreases the time spent in reading information by an individual and also requires less storage space on disk. Text summarization is the process of identifying the important features/sentences in the original text and compressing the document to a concise set of sentences. It is an important field in Natural Language Processing (NLP). A document contains different types of entities such as punctuation, letters, words, sentences, paragraphs etc. The purpose of text summarization is not to find the frequently occurring words but to extract those sentences that summarize the document without losing its meaning.

Let us describe the various techniques of text summarization.

A. Output-Based Summarization :

There are two categories – abstractive and extractive

1) Abstractive Method:

This method involves understanding the whole concept of the text during the comprehension phase and then making a summary in a natural language by taking the importance of the document in the production phase. In this process the original sentences might be changed, and new ones can be produced which will give a short representation of the original text. This is a difficult method as it is part of deep learning and is still being researched.

2) Extractive Method:

This method consists of selecting few sentences which represent the meaning of the document. These sentences are joined to form a summary which describes the original document. The sentences are selected using some statistical or linguistic approaches using NLP methods. This is the most popular method and can be implemented more easily than other Abstractive method.

B. Content-based Summarization :

There are two categories: Generic and Query-based.

1) Generic method:

In this method the person reading the summary might not have any prior knowledge of the topic it is related to and might not have access to the original document. This method gives out a general summary understandable for everyone and give a general meaning for the text.

2) Query-based method:

This category is opposite to the Generic Summarization, in this method the assumption is the reader knows about the specific topic and has read the original text. Any specific questions the reader has can be answered in this.

C. Number of input text-based Summarization:

Based on the number of documents given as input it can be divided into two categories:

1) Single Document Summarization:

As the name suggests only a single document is given as an input. Using the document, a summary is generated and given to the reader.

2) Multiple-Document Summarization:

This method is becoming the most used summarization method. It is an extended version to the Single-Document Summarization as the number of related documents given as input are multiple. Redundancy is a common problem between documents and the goal of this method is to identify any differences or similarities among them. There are two ways in which the task can be accomplished.

- All the input documents are combined and treated as one single document and only one summary is generated by extracting the frequent and important sentences and phrases using clustering algorithms.
- Each input document is treated as its own and summary is generated for each document separately. In the end all the summaries are compiled into one final summary.

D. Limitation-based Summarization:

Based on dependencies there can be three categories of this summarization:

1) Domain-dependent:

Summaries are generated based on domain specific input text where only particular fields are taken into consideration. The fields can be any topic like technology, sports etc.

2) Independent:

This method is like the generic summarization under content-based summarization where a general summary is provided to the reader who has no prior knowledge of the topic involved in the original text.

3) Genre-specific:

This method is more specific than Independent summarization and Domain-dependent summarization. It focuses on a specific genre in the given domain. For example, consider summarization on baseball from the sports domain.

E. Detail-Based:

The two categories in this method are Indicative and Informative summarization:

1) Indicative method:

In this summarization, only a handful of the data is used for summarizing the original data and by this means it can induce a sense of interest in the reader which makes them to read the entire original document. This summarization's perfect example would be a movie teaser or trailer which gives a small glimpse into the full movie encouraging people to watch it when it is released.

2) Informative method:

In this summarization, a little more information is used than in Indicative summarization method. Around 30% of the original data is used which highlights the important points of the information.

F. Language acceptance based Summarization:

This method is also classified into two categories Mono-language and Multi-language summarization:

1) Mono-language Summarization:

This method will accept documents in a specific language only. A document in any other language will not be summarized.

2) Multi-language Summarization:

Unlike mono-language summarization, this method will accept multiple languages and process the data to extract the summary. There are no limitations to this method until there is a language which cannot be recognized.

II. CONCEPTUAL STUDY AND IMPLEMENTATION OF THE ALGORITHM

DATASET:

The Dataset is retrieved from the Kaggle NEWS Summary data (https://www.kaggle.com/sunnysai12345/news-summary). The Dataset contains two CSV files: **news_summary.csv** and **news_summary_more.csv**. The news_summary_more.csv contains two columns headlines and text which can be used for headline extraction from the text content. We used the news_summary.csv file which contains columns: **Author, Date, Headlines, Read_more, Text, Ctext**. **Ctext** is the complete text article and **text** is the reference summary of that complete text. Some rows of Complete text do not have sufficient data to the algorithm, so used the URL column which contains relevant **URL** of the Website of each article. We extracted the article as text from each URL using the parser. We parsed the paragraphs in each website and attached the data to their respective reference summary. The extracted Dataframe is saved as CSV to the **S3 bucket** so that it can be accessed and viewed using public URL. The generated summary is grouped with their respective reference summary so that it can be evaluated using evaluation metric.

STEPS OF ALGORITHM / TECHNIQUE:

- **PARSING**: The document contains of URL, Complete text and Reference Summary. The complete article text has some null values for some rows, so we retrieved the complete article text from the URL using parser. We used beautifulSoup library for parsing the paragraphs from the Website using the URL of each row. The parsed data is saved with the reference summary text in the S3 bucket.
- **SENTENCE TOKENIZATION:** Tokenization is the process of recognizing the initial starting index and the end index of the sentence. Each sentence is made as separate entities by giving indexes to each sentence representing the tokens.
- **BAG OF WORDS:** After splitting the sentences from the document, we split the sentences into words.

These words are used to calculate the TF_IDF which gives the similarity of the sentences.

- **CALCULATE SIMILARITY USING TF_IDF**: The term frequency and the Inverse document frequency is calculated for each term and the document. Based on that the TF_IDF the similarity between the sentences is calculated as a graph.
- **PAGE RANK:** The page rank score for each sentence is calculated using PageRank Algorithm. This gives the ranking for each sentence.
- **SUMMARIZATION:** Based on the page rank score of each sentence the summary is generated. We have generated summary by using 1 sentence, 3 sentences, 5 sentences, 7 sentences, 9 sentences which has the highest page rank.
- **EVALUATION METRICS:** The reference summary and the generated summary is evaluated using the Rouge library which is the library used for evaluation of text summary and gives the precision, recall and F-statistics.

TEXT PROCESSING:

- **Tokenization:** This is also known as *Sentence Boundary Disambiguation.* This is the process of identifying where the sentences initiate and end. Each sentence is tokenized with an indexed at both the start and end points. We used NLTK library and the Punkt module for preprocessing the data.

- **Count Vectorization:** We split the sentences into words using the bag of words approach. In this approach, the words are collected in the bag of words and the occurrence for each word for each document is count in a binary format. If the word is present it is counted as 1 else 0. This output the words in each document as count vector in binary format.

MAJOR APPROACHES / STEPS OF THE ALGORITHM:

- **TF_IDF CALCULATION FOR SIMILARITY GRAPH:**

The similarity of the sentences is normalized before ranking based on the TF-TDF score. The output from the tf-idf function will be in the form of matrix which gives the similarity of words and documents based on the term frequency and the inverse document frequency. We transformed the similarity matrix into graph where the relations between the sentences are linked based on the similarity. This is a mirrored matrix, where both the rows and columns correspond to sentences, and the elements describe how similar the two sentences are. The scores are in the binary format either 1 or 0 based on similar or dissimilar sentences.

- **PAGE RANK APPROACH:**

PageRank Algorithm developed by Larry Page is used for the ranking of web documents by the Google Internet search engine.
In this approach, we assign weights to each element of a hyperlinked set of documents to measure its relative importance within the set. Based on the inlink and the outlinks of each documents the rank score is calculated. A hyperlink to a page counts as a vote of support. The document that has the highest inlinks and outlinks are considered as the highest page and it's page rank has the highest rank value. This requires several iterations to calculate the page rank for each sentence which yields the sentences with more accurate page rank. The more the number of iterations the more accurate will the page rank and there is a threshold for the iterations. Based on the rank of each sentences, the highest ranking sentences are summarized into the generated summary. The number of sentences can be changed and observed for the best model. Some documents yield best model with particular number of sentences. In our case model3 which has the number of sentences as 5 is the best model. This evaluation is calculated based on the precision, recall and f-statistics.

Steps of the Algorithm:
Step 1: Convert the paragraph into sentences
Step 2: Text processing (Removal of stop words)
Step 3: Tokenizing the article into sentences
Step 4: Evaluate the weighted occurrence frequency table for the words
Step 5: Substitute words with their weighted frequencies
Step 6: Finding the weighted frequencies of the sentences
Step 7: Calculating the threshold of the sentences
Step 8: Getting the summary
Step 9: Repeat these steps for all articles to get the summary for each document
Step 10: Evaluation of results and generating metrics for the performed process

EVALUATION METRICS:

ROUGE – Recall-Oriented Understudy for Gisting Evaluation is a library available in PySpark which is being used for evaluating the performance of our algorithm. The main purpose of this evaluation is to compare the original information with the summary generated by the algorithm.

The three types of techniques available to use are ROUGE-N, ROUGE-W and ROUGE-L. They are the granularity of texts which are compared between the initial summaries and the algorithm generated summaries. Further details are given below:

1. ROUGE-N:

This method measures unigram (ROUGE-1), bigram (ROUGE-2) and trigram (ROUGE-3) and more higher order n-grams. It checks for the number of words overlapping while calculating each case.

2. ROUGE-L:
This measures the Longest Common Sequence of words in the sentences. There are no consecutive matches happening but considers the sequence of matches which will take place.

3. ROUGE-W:
It is like ROUGE-L but considers Weighted Longest Common Subsequence. Checking for consecutive LCS's is the primary mode of calculation.

Language and Libraries used for the process:
We have implemented our algorithm using PySpark in our local machine and also in the databricks notebook and installed the following libraries from PyPi which are required to run the code:

- py-ROUGE – v1.1
- NetworkX – v2.4
- NLTK – v3.4.5
- S3 File System

The input and output text files were retrieved from and stored in a bucket on AWS S3.

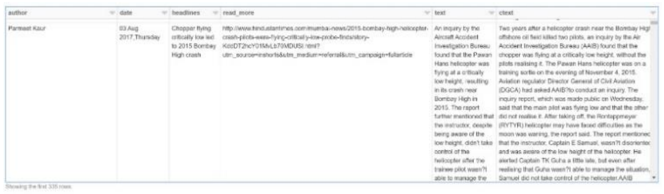
III. Analysis of results

After performing the above procedure, we have the following:

1. Original news article is shown in 'Fig. 1' and their summaries extracted from the Kaggle website.
2. Analysis of the generated summary and the reference summary Algorithm generated summary of the news articles in the two tables labeled as 'Table 1'.
3. Generated summary is displayed in 'Fig. 2'.
4. Histogram representation in 'Fig. 3' showing Recall, Fstatistics and Precision based on the number of sentences in text.
5. Bar plot representation in 'Fig. 4' showing Recall, Fstatistics and Precision based on the number of sentences in text.
6. Bar plot representation in 'Fig. 5' of Precision based on number of sentences in text.

We used Precision, Recall and F1-Score to measure the performance of our model. The following are the result we got from 5 models:

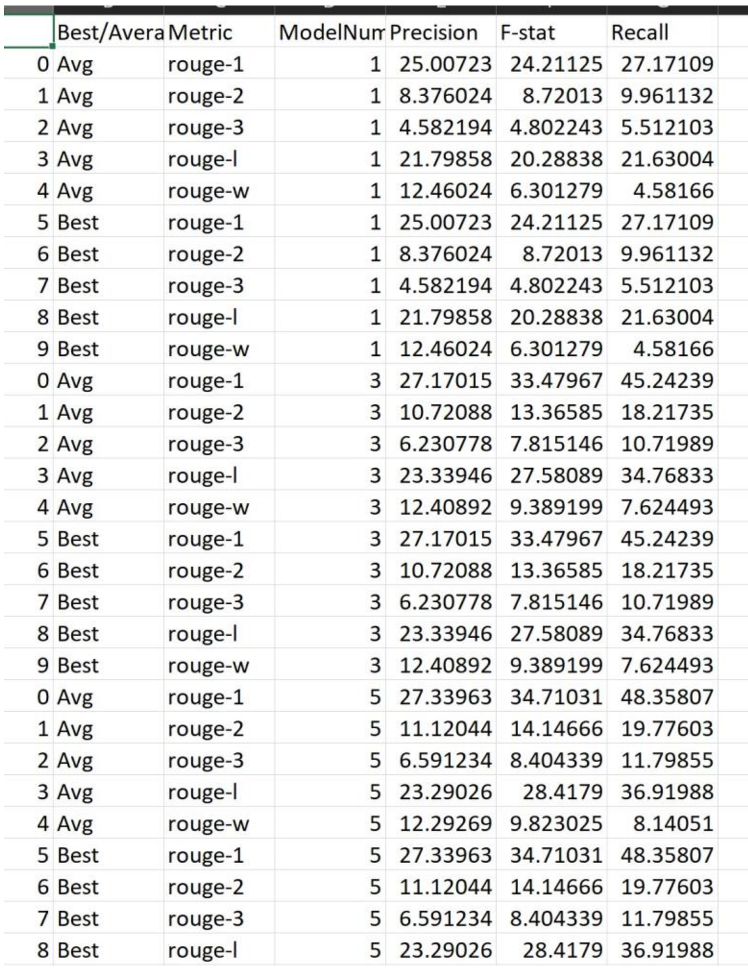
- ➢ Model 1: Number of sentences in summary = 1
- ➢ Model 2: Number of sentences in summary = 3
- ➢ Model 3: Number of sentences in summary = 5
- ➢ Model 4: Number of sentences in summary = 7
- ➢ Model 5: Number of sentences in summary = 9



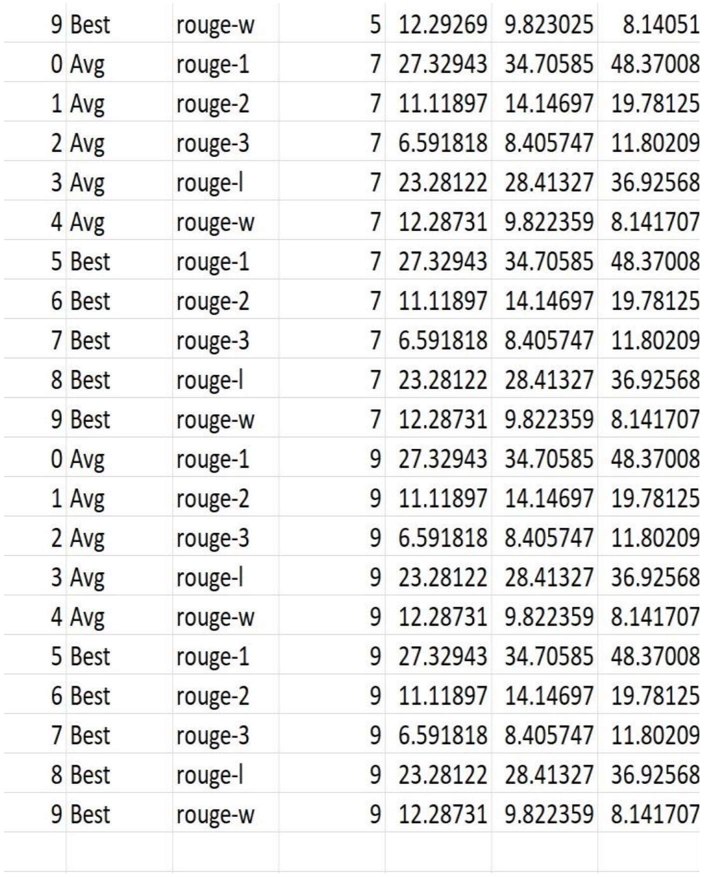
author	date	headlines	read_more	text	ctext
Parmeet Kaur	03 Aug 2017,Thursday	Chopper flying critically low led to 2015 Bombay High crash	http://www.hindustantimes.com/mumbai-news/2015-bombay-high-helicopter-crash-pilots-were-flying-critically-low-probe-finds/story-KddDT2hcY01MvLb70MDUSI.html?utm_source=inshorts&utm_medium=referral&utm_campaign=fullarticle	An inquiry by the Aircraft Accident Investigation Bureau found that the Pawan Hans helicopter was flying at a critically low height, resulting in its crash near Bombay High in 2015. The report further mentioned that the instructor, despite being aware of the low height, didn't take control of the helicopter after the trainee pilot wasn?t able to manage the	Two years after a helicopter crash near the Bombay Hig offshore oil field killed two pilots, an inquiry by the Air Accident Investigation Bureau (AAIB) found that the chopper was flying at a critically low height, without the pilots realising it. The Pawan Hans helicopter was on a training sortie on the evening of November 4, 2015. Aviation regulator Director General of Civil Aviation (DGCA) had asked AAIB?to conduct an inquiry. The inquiry report, which was made public on Wednesday, said that the main pilot was flying low and that the other did not realise it. After taking off, the Rontappmeyer (RYTYR) helicopter may have faced difficulties as the moon was waning, the report said. The report mentionec that the instructor, Captain E Samuel, wasn?t disorientec and was aware of the low height of the helicopter. He alerted Captain TK Guha a little late, but even after realising that Guha wasn?t able to manage the situation, Samuel did not take control of the helicopter.AAIB

Showing the first 335 rows.

Fig. 1. Initial dataset after loading.

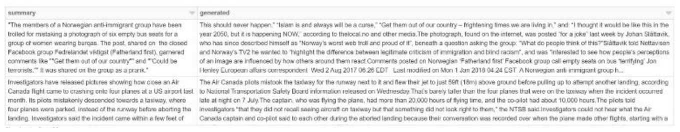


	Best/Avera	Metric	ModelNum	Precision	F-stat	Recall
0	Avg	rouge-1	1	25.00723	24.21125	27.17109
1	Avg	rouge-2	1	8.376024	8.72013	9.961132
2	Avg	rouge-3	1	4.582194	4.802243	5.512103
3	Avg	rouge-l	1	21.79858	20.28838	21.63004
4	Avg	rouge-w	1	12.46024	6.301279	4.58166
5	Best	rouge-1	1	25.00723	24.21125	27.17109
6	Best	rouge-2	1	8.376024	8.72013	9.961132
7	Best	rouge-3	1	4.582194	4.802243	5.512103
8	Best	rouge-l	1	21.79858	20.28838	21.63004
9	Best	rouge-w	1	12.46024	6.301279	4.58166
0	Avg	rouge-1	3	27.17015	33.47967	45.24239
1	Avg	rouge-2	3	10.72088	13.36585	18.21735
2	Avg	rouge-3	3	6.230778	7.815146	10.71989
3	Avg	rouge-l	3	23.33946	27.58089	34.76833
4	Avg	rouge-w	3	12.40892	9.389199	7.624493
5	Best	rouge-1	3	27.17015	33.47967	45.24239
6	Best	rouge-2	3	10.72088	13.36585	18.21735
7	Best	rouge-3	3	6.230778	7.815146	10.71989
8	Best	rouge-l	3	23.33946	27.58089	34.76833
9	Best	rouge-w	3	12.40892	9.389199	7.624493
0	Avg	rouge-1	5	27.33963	34.71031	48.35807
1	Avg	rouge-2	5	11.12044	14.14666	19.77603
2	Avg	rouge-3	5	6.591234	8.404339	11.79855
3	Avg	rouge-l	5	23.29026	28.4179	36.91988
4	Avg	rouge-w	5	12.29269	9.823025	8.14051
5	Best	rouge-1	5	27.33963	34.71031	48.35807
6	Best	rouge-2	5	11.12044	14.14666	19.77603
7	Best	rouge-3	5	6.591234	8.404339	11.79855
8	Best	rouge-l	5	23.29026	28.4179	36.91988



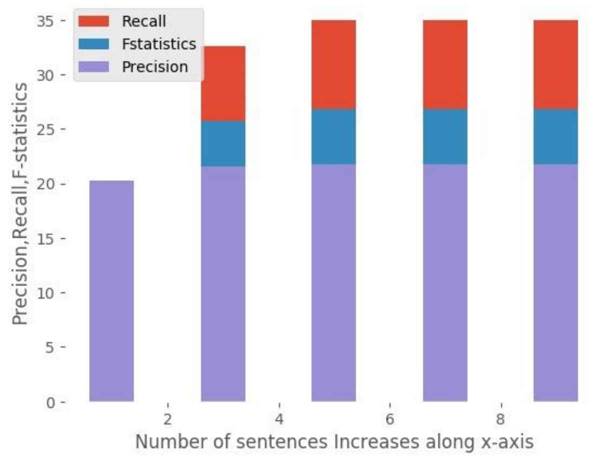
9	Best	rouge-w	5	12.29269	9.823025	8.14051
0	Avg	rouge-1	7	27.32943	34.70585	48.37008
1	Avg	rouge-2	7	11.11897	14.14697	19.78125
2	Avg	rouge-3	7	6.591818	8.405747	11.80209
3	Avg	rouge-l	7	23.28122	28.41327	36.92568
4	Avg	rouge-w	7	12.28731	9.822359	8.141707
5	Best	rouge-1	7	27.32943	34.70585	48.37008
6	Best	rouge-2	7	11.11897	14.14697	19.78125
7	Best	rouge-3	7	6.591818	8.405747	11.80209
8	Best	rouge-l	7	23.28122	28.41327	36.92568
9	Best	rouge-w	7	12.28731	9.822359	8.141707
0	Avg	rouge-1	9	27.32943	34.70585	48.37008
1	Avg	rouge-2	9	11.11897	14.14697	19.78125
2	Avg	rouge-3	9	6.591818	8.405747	11.80209
3	Avg	rouge-l	9	23.28122	28.41327	36.92568
4	Avg	rouge-w	9	12.28731	9.822359	8.141707
5	Best	rouge-1	9	27.32943	34.70585	48.37008
6	Best	rouge-2	9	11.11897	14.14697	19.78125
7	Best	rouge-3	9	6.591818	8.405747	11.80209
8	Best	rouge-l	9	23.28122	28.41327	36.92568
9	Best	rouge-w	9	12.28731	9.822359	8.141707

TABLE 1. RESULTS SHOWING PERFORMANCE OF OUR ALGORITHM IN TERMS OF PRECISION, RECALL, F1 MEASURE FOR ALL MODELS USING ROUGE 1, 2, L, W



summary	generated
"The members of a Norwegian anti-immigrant group have been trolled for mistaking a photograph of six empty bus seats for a group of women wearing burqas. The post, shared on the closed Facebook group Fedrelandet viktigst (Fatherland first), garnered comments like ""Get them out of our country"" and ""Could be terrorists."" It was shared on the group as a prank."	This should never happen." "Islam is and always will be a curse," "Get them out of our country – frightening times we are living in," and: "I thought it would be like this in the year 2050, but it is happening NOW," according to thelocal.no and other media.The photograph, found on the internet, was posted "for a joke" last week by Johan Slåttavik, who has since described himself as "Norway's worst web troll and proud of it", beneath a question asking the group: "What do people think of this?"Slåttavik told Nettavisen and Norway's TV2 he wanted to "highlight the difference between legitimate criticism of immigration and blind racism", and was "interested to see how people's perceptions of an image are influenced by how others around them react.Comments posted on Norwegian 'Fatherland first' Facebook group call empty seats on bus 'terrifying' Jon Henley European affairs correspondent Wed 2 Aug 2017 06.26 EDT Last modified on Mon 1 Jan 2018 04.24 EST A Norwegian anti immigrant group h...
Investigators have released pictures showing how close an Air Canada flight came to crashing onto four planes at a US airport last month. Its pilots mistakenly descended towards a taxiway, where four planes were parked, instead of the runway before aborting the landing. Investigators said the incident came within a few feet of	The Air Canada pilots mistook the taxiway for the runway next to it and flew their jet to just 59ft (18m) above ground before pulling up to attempt another landing, according to National Transportation Safety Board information released on Wednesday.That's barely taller than the four planes that were on the taxiway when the incident occurred late at night on 7 July.The captain, who was flying the plane, had more than 20,000 hours of flying time, and the co-pilot had about 10,000 hours.The pilots told investigators "that they did not recall seeing aircraft on taxiway but that something did not look right to them," the NTSB said.Investigators could not hear what the Air Canada captain and co-pilot said to each other during the aborted landing because their conversation was recorded over when the plane made other flights, starting with a

Fig. 2. Given summary and algorithm-generated summary compared.



0

Fig. 3. Bar plot showing Recall, Fstatistics and Precision based on number of sentences

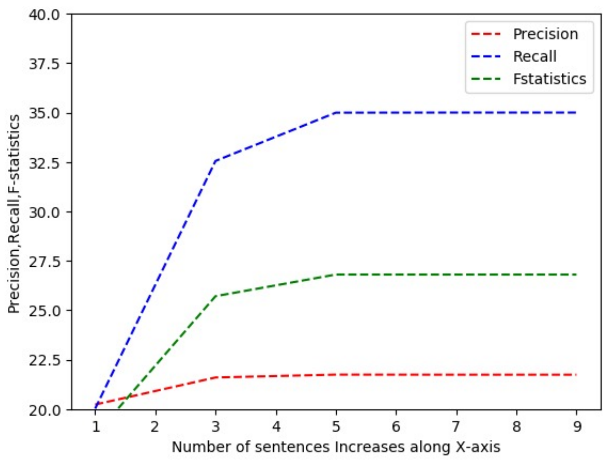


Fig. 4. Line plot showing Recall, Fstatistics and Precision based on number of sentences

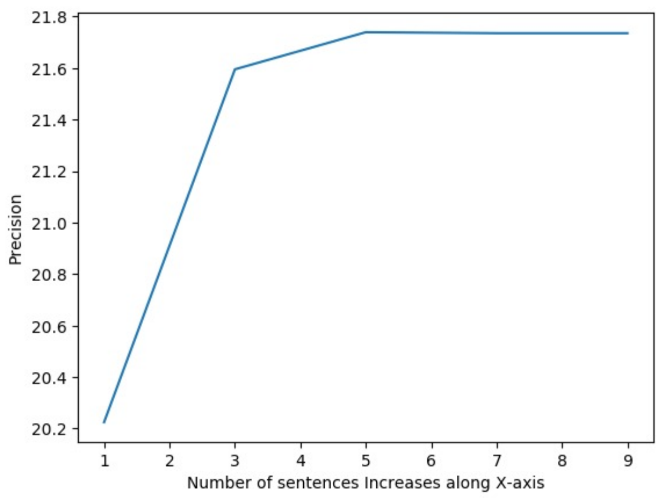


Fig. 5. Line plot showing Precision based on number of sentences

Observations:

1. The number of sentences in the summary is directly proportional to the recall as there is a chance that the generated summary may contain unigrams, bigrams or some other similar words.
2. The number of sentences in the summary is inversely proportional to the precision because more the content in the summary the more it is equivalent to the original text.
3. As the occurrence of two or more consecutive words in the reference summary and generated summary is less the bigram results (rouge-2) are lower than Unigram (ROUGE-1).
4. F1 score = Average (Precision, Recall). This increases till Model 3 and then reduces for the

following models. Model 3 has the highest F1 score – 33.47.

5. There is a slight tradeoff between recall and precision in all the models.
6. Unigram outputs are much higher because it is identifying all single words in both the reference and algorithm generated summary.

Hence, by looking at the F1 scores we can say that Model 2 with summary sentence size=5 is the best model in our case.

We observe from the graphs ("Fig. 4", "Fig. 5") that there are fewer high values present in the recall curve than the precision curve. Precision curve follows a Gaussian distribution where most of the values are centered around 0.5 while in recall curve, there are also a high number of points with less than 0.2 recall value. On the whole, their weighted average which is the F1 score proves out to be the best among all the models.

IV. Conclusion and Future Work

For the algorithm, the most efficient model is Model Number 3 where number of sentences in summary is equal to 5. The most effective and nifty methods used so far in document summarization rely on extractive methods which aim at selecting the most relevant sentences from the original documents. Moreover, most of the time only a part of a sentence is relevant but extracting only sub-sentences is still far from being operational. Finally, extracting sentences from different documents may produce an inconsistent and/or hard-to-read summary. Getting a good number as the number which maximizes the score of the summary details is important to achieve a balance of precision and recall and hence produces an informative and well-formed summary. Recently, an approach using Restricted Boltzmann Machine and Fuzzy Logic to select important sentences from the text keeping the summary meaningful and lossless.

Research for text summarization goes a lot further than just evaluation, since evaluating a summary involves having an accurate description of the content of the documents. Using Convolutional neural networks and Recurrent neural networks for this might increase the overall precision and accuracy of the project. Deep learning and usage of neural networks might require additional hardware and computing resources.

V. References

[1] Kavita Ganesan, "What is ROUGE and how it works for evaluation of summaries?".
https://kavita-ganesan.com/what-is-rouge-and-how-it-works-for-evaluation-of-summaries/#.XqzsumhKhPZ

[2] "Text Summarization Evaluation – BLEU vs ROUGE"
https://stackoverflow.com/questions/38045290/text-summarization-evaluation-bleu-vs-rouge

[3] Rachael Tatman, "Evaluating Text Output in NLP: BLEU at your own risk".
https://towardsdatascience.com/evaluating-text-output-in-nlp-bleu-at-your-own-risk-e8609665a213

[4] Anukarsh Singh and Divyanshu Daiya "A Qualitative Introduction to Automatic Text Summarization".
https://medium.com/the-ai-herald/a-qualitative-introduction-to-automatic-text-summarization-30f025c853c0

[5] Praveen Dubey, "Understand Text Summarization and create your own summarizer in python".
https://towardsdatascience.com/understand-text-summarization-and-create-your-own-summarizer-in-python-b26a9f09fc70

[6] S. A. Babar, Pallavi D. Patil, "Improving performace of Text Summarization".
https://www.sciencedirect.com/science/article/pii/S1877050915000952

[7] Pranay, Aman and Ayush, "Text Summarization inn Python: Extractive vs Abstractive techniques revisited".
https://rare-technologies.com/text-summarization-in-python-extractive-vs-abstractive-techniques-revisited/

[8] Desh Raj, "Metrics of NLG Evaluation".
https://medium.com/explorations-in-language-and-learning/metrics-for-nlg-evaluation-c89b6a781054

[9] Sciforce, "Towards Automatic Text Summarization: Extractive Methods".
https://medium.com/sciforce/towards-automatic-text-summarization-extractive-methods-e8439cd54715