Applied Mathematics and Computational Intelligence
Volume 11, No.1, Dec 2022 [174-196]

Text Summarization for News Articles by Machine Learning Techniques

Hew Zi Jian[1], Olanrewaju Victor Johnson[2], Chew XinYing[3*], Khaw Khai Wah[4]

[1,2,3*] School of Computer Sciences, 11800, Universiti Sains Malaysia, Pulau Pinang, Malaysia.
[4] School of Management, 11800, Universiti Sains Malaysia, Pulau Pinang, Malaysia.

[*] Corresponding author : xinying@usm.my

Received: 12 November 2021; Accepted: 04 July 2022; Available online (In press): 23 July 2022

ABSTRACT

Text summarizing is very instrumental in natural language text comprehension systems to constructing a text summary using more abstract, condensed knowledge structures. Extractive text summarization is therefore built on language processing to extract the essence sentences of a long text article to produce a summary. Though the known manual process had recorded achievement over time and recently, several machine learning models for extractive text summarization had also been proposed. However, there is a lack of research that benchmark the comparative performance of these machine learning models. This paper, therefore, helps to identify the champion machine learning model in text summarization for news articles and to identify the best text preprocessing method in the machine learning of text summarization. CNN/Daily Mail database is employed for the comparative study of text summarization using chosen classifiers. Random Forest (RF) classifier provides with a champion performance of Rouge-l score, Rouge-2 score and Rouge-L score as 8.2845, 2.884, and 7.9694 respectively.

Keywords: Classifier, CNN/Daily Mail, Machine Learning, News Article, Text Summarization

1 INTRODUCTION

Text summarization, a process of shortening a long piece of text into a condensed summary focusing the useful information, is gaining more attention in recent times [1],[2]. This is not far from the fact that it is time-consuming for people to read and understand long articles including news, research papers, storyline, and so on. Text summarization, therefore, provide people with understanding the main points of the articles efficiently and minimizing workload to analyze the content of the articles. It is also able to accelerate the process of content analysis in search engines, chatbots, voice-to-text technology, and so on. Manual summarization being very tedious and expensive task is suggestive to adopting a less expensive and less cumbersome approach called automatic text summarization using both Natural Language Processing (NLP) and Artificial Intelligence (AI) [3]. Although ATS being faced with undisputed challenges relating to sentence ordering, time-related dimensions, verbosity and so on as stated in [1], the strive to produce accurate summaries that are legible and coherent for users, cover all the major points in the input text, and are free of repetition and verbosity is the driving force in recent ATS system [3].

There are two approaches of text summarization in NLP based on key-phrase manipulation, on one hand, namely extractive-based summarization and abstraction-based summarization as shown in

Figure 1. Extractive-based summarization is the technique to obtain the important key phrases from the source text and combine it as a summary. Abstraction-based summarization entails paraphrasing the source text which creates new phrases and a summary that is more semantic representation [4]. In general, abstractive summarization may be seen as more advanced and closer to human interpretation, however, the research in that area may be considered in the very beginning. The extractive summarization methods are better consolidated and considered efficient in the automated text summarization [5]. On the other hand, text summarization also can be categorized into two different groups, inductive and informative as shown in Figure 1. Inductive summarization provides the main idea of the source text for the summary and the summary's length is approximately five percent of the source text, while the informative summarization provides brief information on the source text and the summary's length is approximately twenty percent of the source text [4]. A Hybrid type and a more comprehensive discussion on the classification of text summarization is provided in [3]. For the news articles, the summary provided should be an inductive summary to provide a brief idea of the article to the readers so that the readers can obtain useful information and decide to read into details from the original article.

As text summarization is becoming increasingly vital, several methods appeared for automated text summarization including statistical-based [6], machine learning (ML)-based [7], coherent-based [8], graph-based [9], and algebraic-based [10]. Other specific techniques, in recent times, for text summarization include copying mechanisms, reinforcement, multiple communicating encoders and performing pretrained models such as ELMo, GPT, Bidirectional Encoder Representations from Transformers (BERT) were mentioned in [11]. Meanwhile ML continues to attract increasingly support for text summarization due to its ability to review large volumes of data and identify patterns that are hard to be observed by a human. Several ML techniques and models are proposed for text summarization including Naïve Bayes [12], Support Vector Machine (SVM) [13], Decision Tree (DT) [14], and Random Forest (RF) [15]. These models are trained to produce a summary by identifying the sentences that have a high possibility to be a summary. To produce more efficient and accurate training in the ML model, several pre-processing techniques are proposed including text pre-processing and feature extraction. Text preprocessing is classified into tokenization, normalization, and noise removal. Tokenization is the process that split the string of the text into smaller pieces; Normalization is used to convert the text into the same level of pattern while noise removal is aimed to clean the text. In addition, a common set of metrics for evaluating the correctness of the generated summary such as Rouge-N, Rogue-L are note-mentioning under this discourse.

Consequent to the above, there available many ML models used in text summarization research. The ML models provide different learning algorithms and produce different predictions of the result for the summary generated. Meanwhile, in the text summarization by ML, it is important to choose a suitable ML model to optimize the training time, reduce the burden of the machine, and improve the accuracy. More so, there are massive text data needed to be trained in ML to obtain a satisfying accuracy of text summarization. However, there is a lack of researches related to the comparison of the performance of different ML models in this field existing. An identification of the champion ML model to perform text summarization is essential for the best decision-making in research development.

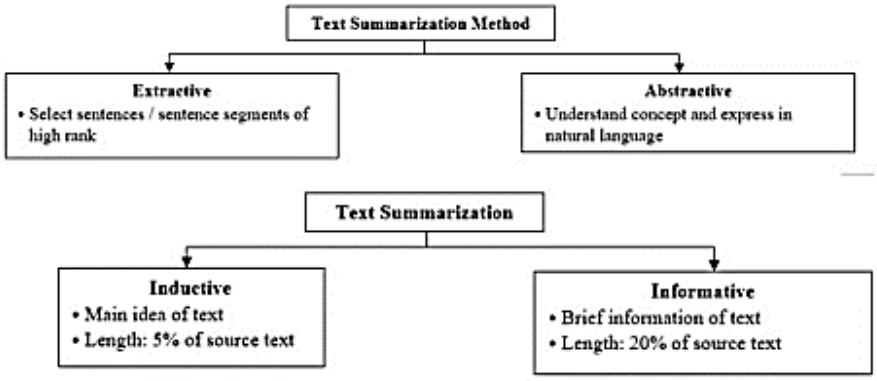

Figure 1: Categories of Text Summarization Techniques

Therefore, in this paper, we focused on finding the champion ML model in extractive text summarization for news articles and, in addition, constructing the suitable preprocessing method for the text summarization in CNN/Daily Mail dataset. Six different features are extracted from the sentences in the article to represent the sentences. These features are used to train the ML to learn the pattern with the summary result. Chosen ML models are trained to extract the important sentences from the news article in CNN/Daily Mail dataset. The sentences are combined into a summary and the accuracy of the summary with the original summary is evaluated. Obtained champion classifier in our work provides the best performance of Rouge Scores. The rest of the paper is organized as follows: Section 2 discusses literature review, materials and methods were the focus in Section 3, In Section 4, we present the results and discussion while the conclusion is presented in section 5.

2 LITERATURE REVIEW

2.1 Overview of Machine Learning Process

The ML process could be viewed as a lifecycle as shown in Figure 2. The lifecycle starts with understanding the research problem and the requirement of the task goal. The lifecycle then goes clockwise into data acquisition, data exploration, data preprocessing, feature extraction, model deployment, and data analysis and visualization. After an iteration end, the process and the prototype results are revised and start the next iteration for improvement. In each task, the lifecycle can adjust by moving the next or the previous step to obtain the intended result. The explanation of each main task is explained in Table 1.

2.2 Text Summarization and Related Concepts in ML

In extractive text summarization, the steps are source document preparation, preprocessing, feature extraction, sentence score calculation, sentence extraction and summary production [16]. Text summarization is taken by ML as a classification problem [17]. The sentences are classified as summary sentences and non-summary sentences based on the sentence features. It uses a training set of source documents and their corresponding extractive summaries for ML. Several components

are needed to be considered for text summarization by ML: dataset, text preprocessing, feature extraction, ML model, and evaluation metrics. Each of these is described in sections below.

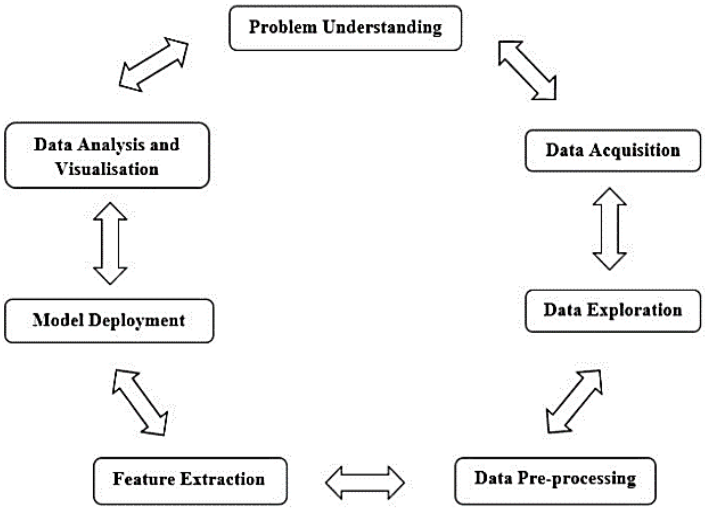

Figure 2: Iterative nature of Machine Learning process

Table 1: Explanation of Machine Learning process

Task	Explanation
Problem Understanding	Understand the underlying problem, objectives, the problem-related information and concepts, and the goals of the task.
Data Acquisition	Collect the dataset related to the task e.g. news articles.
Data Exploration	Explore the format of the dataset and identify the usability of the data.
Data Pre-processing	Clean and format the dataset for further development.
Feature Extraction	Select and transform features into a usable format e.g. convert the text data into a numerical vector.
Model Deployment	Train and identify the ML model with possible performance evaluation metrics.
Data Analysis and Visualisation	Test and analyze the model with a new dataset and identify the performance of each different ML model. Identify and evaluate the optimal machine learning model.

2.2.1 Datasets

In literature, there are a couple of standard datasets publicly available for text summarization research problems. We reviewed some of these in the section below as i) Document Understanding Conference (DUC), a series of evaluations of automatic text summarization systems running annually since 2001 [18]. The DUC dataset is not large enough for training models with a large number of parameters and usually used along with other datasets or evaluations only [19], [20], [21], ii) Gigaword dataset [22] was produced by Linguistic Data Consortium (LDC) since 2003. The dataset contains nearly ten million documents from seven news outlets. However, there are no summaries paired with the original articles in the dataset, iii) New York Times (NYT) Dataset is a large collection of articles containing over 1.8 million articles written and published by the New York Times between 1987, and 2007 with article metadata [23]. There are around 650k article summaries written manually. The data is an XML specification standard and structure of discrete news articles. It is mostly used for extractive summarization systems [24], [25] and used for the abstractive system since 2017 [26], iv) CNN/Daily Mail dataset was developed by [27] consisting of two datasets, CNN and Daily Mail. The CNN contains around 93k documents and the Daily Mail contains around 220k documents. Each dataset contains two parts which are questions and stories. The questions part contains the corpus of the document–query–answer triples for the question and answering research. It provides the link to the original document, queries with anonymized entities, and the entities' value as the answer. The recent highest Rouge-L F1-scores for the extractive text summarization research of the dataset is 40.55 with semantic matching framework approaches in extractive subsets of the dataset [28], and iv) NEWSROOM Dataset [29] is the most recent large-scale dataset for text summarization which contains 1.3 million news articles and human-written summaries. The dataset is collected from search and social media metadata between 1998 and 2017. Categorized into three extractive quantiles, abstractive, extractive, and mixed according to the summary with the measurement of extractive-ness consist an average article length of 658.6 words and summary length of 26.7 words. The recent highest Rouge-L F1-scores for the extractive text summarization research of the dataset is 28.68 with Text Rank approaches in extractive subsets of the dataset.

2.2.2 Text Summarization concepts

In the section, we discussed some concepts as an important ingredient of any natural language processing (NLP) system. [30] posited the need for these concepts for text processing in the NLP system to reduce the indexing file size of the text and to improve the efficiency and effectiveness of the information retrieval system. It also has a great impact on reducing the required time and speed resources for ML [31]. These concepts are discussed in the section below.

Sentence Segmentation: A process that converts raw text into sentences and is also a prerequisite step before the manipulation of text at the word level [32]. It can be done by separating sentences according to the dot between sentences [33]. However, this may also separate the abbreviation with a dot or not separate the sentence that ended with a question mark (?), exclamation point (!), and so on. A variant method in [34] contains a cue phrase to separate the segments that possibly convey independent meanings. Therefore, a single sentence can be separated into multiple smaller sentence segments if it conveys different meanings.

Tokenization / Lexical Analysis: Tokenization or lexical analysis splits a list of characters into a list of tokens to extract the words from the sentence while considering characters such as digits,

hyphens, punctuation marks, etc. The tokens are words, phrases, symbols, or other meaningful elements [30], [32], [33], [35].

Stop Word Removal: Stop words are the words that occurred frequently and do not provide important information for identifying the significant meaning of the content [30]. Articles, prepositions, and conjunctions are natural candidates for a list of stopwords such as "a", "the", "is", "are", "and", "or", etc [35]. The list of stopwords can be extended to include words other than articles, and conjunctions. This process also improves the system performance as it has reduced text data.

Part of speech (POS) tagging: POS is known as the word classes or lexical categories to classify words into their parts-of-speech and label accordingly [36]. A POS tagger analyses output of the words and assigns the appropriate part of the speech tag to each word. POS tagger works to generate tuples and it is useful in the extraction of nouns, adverbs, adjectives.

Stemming Produces the root form of the word by removing the affixes (prefixes and suffixes). [37] uses the Porter Stemming Algorithm comprising suffix stripping to produce stems by removing the commoner morphological and inflexional endings from the words in the English language. Porter Stemmer is often used in stemming due to its advantages of simplicity and speed [33].

Lemmatization converts a word to its dictionary form. Compared to stemming, lemmatization considers the context and ensures the dictionary form word belongs to the language without spelling error. The root word converted by lemmatization is called a lemma. WordNet is a large lexical database of English. It grouped nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms, each set expressing a distinct concept. It has offered lemmatization capabilities and is one of the earliest and most commonly used lemmatizers as WordNet lemmatizers [36]. [38] proved that the lemmatization algorithm should be used for text preprocessing to improve the accuracy of the NLP system.

2.2.3 Feature Extraction Concepts

Feature extraction is an important key factor for the performance of the text summarization model. The text document is represented by set, $D = \{S_1, S_2, \dots, S_i\}$, where S_i represents a sentence in document D. The sentences are subjected to the feature extraction process to generate a feature vector for each sentence, $V_i = \{F_1, F_2, \dots, F_n\}$, where V_i is the feature vector of each sentence S_1. The feature extraction can be generally classified into two classes: word-based and sentence-based. We provided a brief description of these techniques in equation 1 for word-based TF-IDF (Term Frequency Inverse Document Frequency) and Table 2 below for sentence-based feature extractions.

$$tf_i = \frac{number\ of\ i\ in\ sentence}{total\ number\ of\ words\ in\ sentence}$$

$$tf_i idf_i = \frac{tf_i log N}{n_i} \quad (1)$$

Where

tf_i is the term frequency of word i in the document

N is the total number of documents

n_i is the number of documents in which word i occurs

2.2.4 Manual News Article Summarization Technique

To get more understanding of summarization techniques and have clear summarization steps in ML, it is an interesting part to research the techniques used in summarizing news articles manually by the authors or summary writers. An article published by Summarizing Biz [39], stated some appropriate points in summarizing the news articles. The points are summarized and listed in the following:

- Find the context and the importance of the article via the quotes, arguments, and essential data.
- Find the key vocabulary and major keywords and their meaning.
- A personal pronoun should be avoided from the summary.
- The position of the main argument in the news article is mostly in the first two paragraphs.

Table 2: Summary of Text Summarization Feature extraction techniques (see [40] for details)

Sentence Feature	**Formular**
Thematic Word	$F_{thematic} = \frac{no\ of\ thematic\ words\ in\ sentences}{Max(no\ of\ thematic\ words\ in\ sentence)}$
Proper Noun	$F_{proper} = \frac{Number\ of\ proper\ nouns\ in\ sentence}{Number\ of\ words\ in\ sentence}$
Numerical Data	$F_{numerical} = \frac{Number\ of\ numerical\ data\ in\ sentence}{Number\ of\ words\ in\ sentence}$
Cue Phrase	$F_{cue} = \frac{Number\ of\ cue\ phrases\ in\ sentence}{Number\ of\ cue\ phrases\ in\ document}$
Sentence Length	$F_{sent_{len}} = \frac{Number\ of\ word\ in\ sentence}{Max(Number\ of\ words\ in\ sentence)\ in\ document}$
Sentence Position	$F_{sent_{position}} = \frac{N - P + 1}{N}$
Sentence-Sentence Similarity	$Similarity(S_i, S_j) = \frac{\sum_{t=1}^{n} w_{it} w_{jt}}{\sqrt{\sum_{t=1}^{n}(w_{it}^2) \times \sum_{t=1}^{n}(w_{ji}^2)}}$ The similarity between a sentence and the other sentences of the document. It is obtained by the cosine similarity which measures the similarity irrespective of their sizes [16].
Term Weight	$Term\ weight = \frac{Frequency\ of\ the\ term}{Total\ number\ of\ terms\ in\ document}$

2.2.5 Proposed Text Summarisation Modeling

The text summarization in ML is modeled as the classification problem which classifies the sentences into summary sentences and non-summary sentences based on the sentence features possess. News dataset were collected from CNN and Daily Mail websites [27]. The original text consisting story format is pre-processed, tabulated and store as .CSV (comma separated value) format file by conversion. At the text pre-processing stage, word tokenization is provided for to help perform sentence into word-level conversion and other redundancies removed. Thereafter, The sentences are labeled positive and negative where the positive indicates the summary sentence and negative indicates the non-summary sentence. The word-based TF-IDF is used for the feature extraction of the processed dataset. The text summarization model is trained by different machine learning algorithms such as SVM, naïve Bayes, decision tree, and random forest. The performance of the models is evaluated by the F1 score metric with validation data. The champion model obtained is used to extract the sentences from the original news story. Based on the test data, the Rouge score determines how well the model performs. This modeling procedure is as shown in Figure 3.

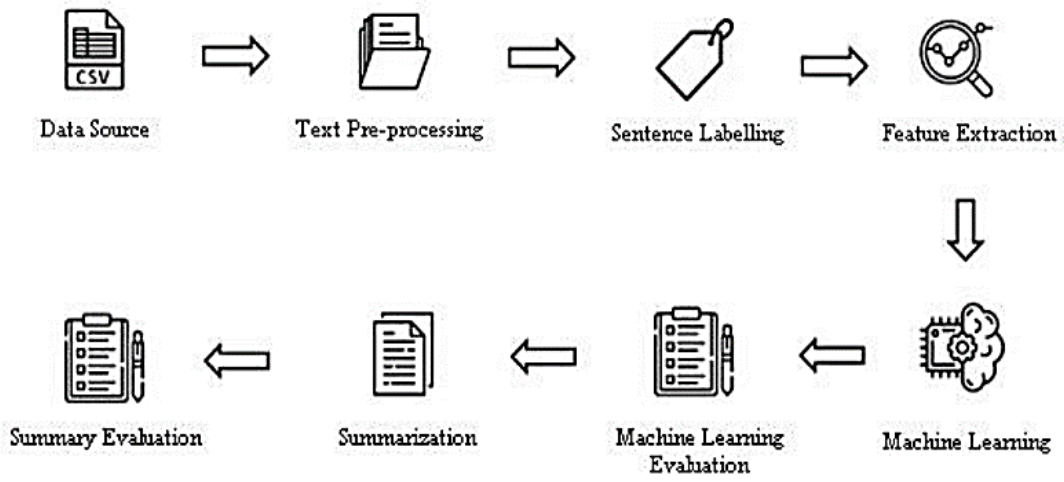

Figure 3: Architecture of the proposed Text Summarzation Modeling

3 MATERIAL AND METHODS

Under this section, we discussed the four cardinal approaches used in our text summarization by ML for News Articles.

3.1 Data

Having discussed a variety of text summarization datasets applicable to New Articles with a close look at their merits and demerits, CNN/Daily dataset was implemented in the paper. The content and structure of the CNN/Daily Mail dataset [27] are analyzed in this section and provided as input into our selected ML models. A total 10,000 records was read from the CNN/Daily Mail dataset for the experiments performed.

3.2 Data pre-processing

3.2.1 Text preprocessing

Text preprocessing is used to clean and normalize the text data to improve the accuracy and reduce the data redundancy and training time of the models. The text preprocessing can be done on sentence-level and word-level. 10,000 rows of the data tabulated from the CNN dataset is read. The decimals and the large numbers with commas in the text are normalized into integers by removing the commas between the numbers, the decimal point and the number behind the decimal point by the regex expressions. To produce the text data at the sentence level, a pre-trained model with the sentence segmentation method proposed in [41] is used subject to the concepts discussed in section 2.2.2. The pre-trained model uses NLTK library with the function sent_tokenize. Figure 4 shows the result of the text pre-processing process carried out on CNN/Daily Mail dataset.

Step	Description
Decimals Normalization	Remove the decimal point and the numbers behind the decimal point.
Large Number Normalization	Remove the comma between numbers.
Sentence Segmentation	Convert text in sentence-level.
News Start Removal	Remove "(CNN) -- " and its preceding words.
In-parenthesis Removal	Remove parentheses and words inside it.
Country Abbreviation Mapping	Map country's abbreviation with the original country's name.
Contractions Expansion	Map contractions with the original words.
"'s" Removal	Remove "'s".
Punctuation Removal	Remove punctuations.
Word Tokenization	Tokenize sentences to sets of words.
Non-ASCII Removal	Remove non-ASCII words.
Lowercase Conversion	Convert words to lowercase.
Stopwords Removal	Remove stopwords.
POS Tagging	Tag words with its POS.
Lemmatization	Lemmatize the words.

Figure 4: Text pre-processing process carried out on CNN/Daily Mail dataset

3.2.2 Features Extraction

Feature extraction in text processing is to transform the textual data into a real-valued vector so that to be understood by the machine. We performed features extraction on the CNN/Daily dataset using concepts provided in equation 1. Since the text data in CNN/Daily Mail dataset has no provision for the title of the article, the title word feature is ignored. The text data was stored as the raw string; therefore, the font-style feature is not sufficient for the data. The biased word feature is 51 and was also ignored because the content of the news articles is varied to each other from different fields and cases. The pronouns are usually a subset of the stop words and are removed. The uppercase word feature was overlapped with the proper noun feature to obtain specific objects such as name, city, country, etc. The proper noun feature was chosen because it refers precisely to the entities while the uppercase word may include the first word in a sentence. Hence, the features that can be obtained from the CNN/Daily Mail text data are thematic word, proper noun, numerical data, cue phrase,

sentence length, sentence position, sentence weight, and sentence-sentence similarity. The thematic words in a document are obtained as the top *n* words with the highest frequency in the cleaned text article. Figure 5 shows the cleaned summary and the top 20 words with the highest word frequency in a text article ordered in descending order. It shows that eight out of 10 of the top 10 words are contained in the summary while decrease harshly after the top 10 words. Therefore, the thematic words were set as the top 10 words in the thematic word feature extraction.

To recognize the proper nouns in the text article, the POS tagging was applied to the original text instead of the cleaned text. This is because the POS tagger needs to recognize the POS of the words according to the sentence structure and the letter case of the words. Figure 6 shows the proper nouns recognized by the POS tagger in the cleaned text and original text of a news article. It shows that the POS tagger is insufficient to recognize the POS in the cleaned text which has only the lowercase lemma and highly depends on the letter case and the sentence structure in the text.

The sentence position feature was obtained by computing the highest score to the first sentence and decreasing accordingly. The method was limited by taking only the top n sentences in the calculation. A simple method that offers value 1 to the first sentence and the last sentence of the paragraph and 0 for other sentences was also considered. Figure 7 shows the graph representing the count of the sentence position that was chosen as the summary sentence in 10,000 CNN news articles. It shows that the count is decreasing from the first sentence to the rest of the sentences. Word-based vector TF-IDF and sentence feature vector were obtained and used in our model analysis as shown in equation 1.

```
syrian official obama climbed top tree know get. obama sends letter head house senate. obama seek congressional approval milita
ry action syria. aim determine whether cw use say united nation spokesman.

[('obama', 28), ('say', 28), ('syria', 25), ('united', 25), ('state', 22), ('military', 17), ('syrian', 14), ('weapon', 14),
('action', 13), ('use', 12), ('chemical', 11), ('saturday', 11), ('congress', 9), ('nation', 9), ('president', 9), ('take', 8),
('would', 8), ('attack', 7), ('official', 7), ('debate', 6)]
```

Figure 5 : Cleaned Text Summary from the Original Text dataset

```
Cleaned Text:  ['obama']

Text:  ['nesirky', 'mccarthy', 'garden', 'majority', 'israel', 'turkey', 'u.n.', 'iran', 'council', 'friday', 'august', 'presid
ent', 'read', 'prime', 'twitter', 'coalition', 'us', 'rosecrans', 'prohibition', 'rodgers', 'safi', 'map', 'david', 'foreign',
'barack', 'rose', 'army', 'september', 'a', 'russia', 'americans', 'cathy', 'senate', 'nader', 'menendez', 'robin', 'washingto
n', 'alexei', 'white', 'lousy', 'kerry', 'eric', 'u.s.', 'mcmorris', 'sen.', 'leader', 'los', 'chair', 'minister', 'john', 'con
ference', 'constitution', 'speaker', 'congress', 'assad', 'china', 'tuesday', 'ban', 'united', 'francois', 'obama', 'italy', 'o
rganization', 'parliament', 'secretary', 'jaafari', 'kevin', 'duma', 'boehner', 'nato', 'syria', 'ki-moon', 'damascus', 'commit
tee', 'robert', 'saturday', 'angeles', 'martin', 'weapons', 'security', 'u.s.obama', 'cameron', 'whip', 'national', 'relation
s', 'bashar', 'states', 'french', 'cantor', 'house', 'pushkov', 'hollande', 'nations', 'democrats', 'state', 'cnn', 'chemical']
```

Figure 6: Proper nouns in Cleaned Text Summary

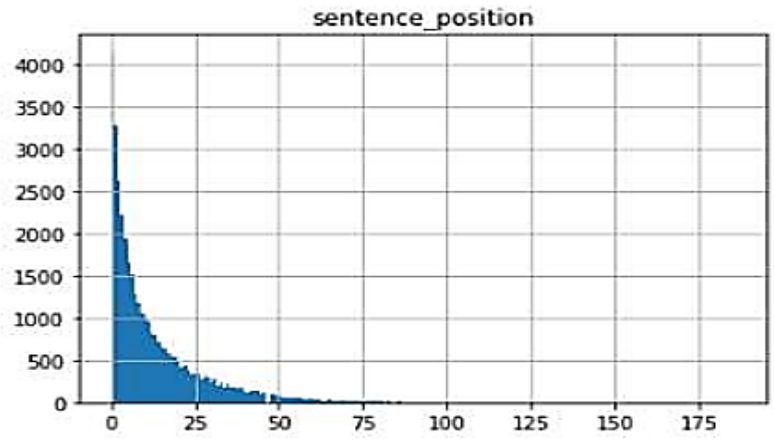

Figure 7: Summary Sentence Position

3.3 Proposed Model Evaluation Process.

Several ML models are proposed in the automated text summarization task. Each model performs different algorithms to classify the sentences into summary sentences and non-summary sentences. The Support Vector Machine (SVM) [42], Naïve Bayes (NB) [43], Decision Tree [44], and Random Forest (RF) [45] are the ML models frequently used for extractive summarization. We provide, therefore, an approach using these selected algorithms as a base classifier to underscore the optimal predictive model for text summarization as shown in Figure 8. The training of the models was carried out by the following process. The positions of the first sentence in 801st, 901st, 1001st, 8001st, and 9001st documents were obtained. These positions were used to separate the data into training data, validation data, and test data with the ratio of 80:10:10 with the total number of documents for ML which are 1000 and 10,000 documents. Both the word-based tf-idf vector dataset and sentence features dataset were split in the same order and ratio.

3.4 Evaluation Metrics

The evaluation of the ML models in the first stage is to determine the performance of selecting the correct summary sentences. The metric F1 score (equation 4) is chosen to determine this performance using a confusion matrix. The metric evaluation tools of accuracy (equation 2),

precision, recall and F1 score are imported from the metrics module in the scikit-learn library. The model evaluation has these basic concepts: true positive (TP), false positive (FP), true negative (TN), and false negative (FN) from the predicted result and validation target data as described in Table 3.

Table 3: Confusion Matrix for Performance Evaluation

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

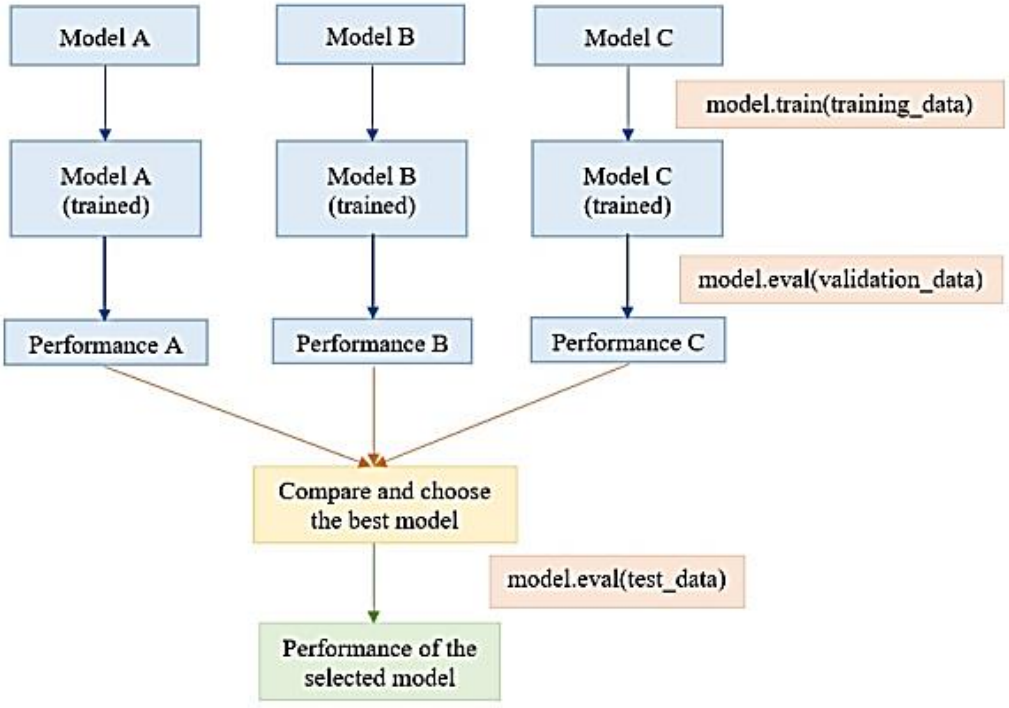

Figure 8: Framework of proposed Model Evaluation Process

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

While the precision and recall are given in equations 3 and 4 to compute the F1 score in equation 5.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Precision = \frac{TP}{TP+FN} \quad (4)$$

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

The summary sentences are extracted to form the summary based on the predicted result of the champion models. The generated summary is compared with the reference summary by using the Rouge score. The F1 scores of Rouge-1, Rouge-2, and Rouge-L are thereafter computed as the final result. The FI ROUGE Score is based on the BLEU score (bilingual evaluation understudy), an algorithm for the evaluation of machine-generated text referring to the human reference text as given in equation 6 [46], [47].

$$BLEU = BP.exp\left(\sum_{n=1}^{N} w_n\, log(p_n)\right) \tag{6}$$

Where n is the n-gram (n items from a given sample of text or speech in a continuous sequence). considered w_n is the weight assigned for n-gram precision ($w_n = \frac{1}{N}$), p_n is the n-gram precision score and BP, the brevity penalty.

And Rouge (Recall-Oriented Understudy for Gisting Evaluation) [47], the metric used for the evaluation of the machine-generated in both automated summarization and machine translation tasks. It is mostly the same as BLEU but has different variants. The Rouge-N and Rouge-L (measures the longest matching sequence of words using the Longest Common Subsequence (LCS)) are given in equations 7 and 8 respectively.

$$Rouge - N = \frac{\sum_{c \in Reference} \sum_{n-gram \in c} Count_{match}(n-gram)}{\sum_{c' \in Reference} \sum_{n-gram' \in c} Count(n-gram')} \tag{7}$$

Where $Reference$ is the human reference summary, $Count_{match}$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries while the Denominator is the total number of n-grams in the human reference summary.

$$Recall_{LCS} = \sum_{i=1}^{u} \frac{LCS_{\cup}(r_i, C)}{m}, \qquad Precision_{LCS} = \sum_{i=1}^{u} \frac{LCS_{\cup}(r_i, C)}{n}$$

$$F1 - Score_{LCS} = \frac{1 + \beta^2 Recall_{LCS} Precision_{LCS}}{Recall_{LCS} + \beta^2 Precision_{LCS}} \tag{8}$$

Where u is the number of sentences of the reference summary, m is the total number of words of the reference summary, n is the total number of words of the candidate summary, C is the candidate summary, r_i is the reference sentence, β is the parameter that is set to a large number ($\rightarrow \infty$) in DUC and $LCS\cup$, the union LCS.

4 RESULTS AND DISCUSSION

The experiment setup for text summarization was implemented on Google Colab with GPU capability using the following packages: Numpy, Pandas, Scikit-learn Scipy and the NLTK in python 3.6 and Rpy2 for R language (C50 package). Our results are presented in the section below.

4.1 The sentence Feature vector for N=1000

The first approach was training the selected ML models (SVM, NB, and DT) with 800 documents and 100 documents for validation with the sentence feature vector as the feature extraction method and the rsults are as presented in Tables 4, 5, and 6. Table 4 shows the SVM model performance with the sigmoid kernel as the champion model since it obtained the highest score in recall and F1 score. The training time is also relatively less than other kernels except for the linear kernel which used the classifier based on the LIBLINEAR. The SVM models that obtained 100% precision may be due to the overestimation of the sentences as summary sentences.

Table 4: Results of SVM using different kernels

SVM Kernel	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
Linear	0.3167	93.75	3.2189	6.2241
Polynomial (Degree = 2)	20.374	**100**	3.2189	6.237
Polynomial (Degree = 3)	426.15	**100**	3.2189	6.237
Radial Basis Function (RBF)	17.139	**100**	5.5794	10.569
Sigmoid	**8.8**	16.241	**15.022**	**15.608**

Table 5: Results of DT using different Tree methods

DT	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
C5.0	0.8161	**90.4255**	18.2403	30.3571
CART	**0.2740**	30.2231	**31.9742**	**31.0740**

Table 5 shows the CART tree as the champion model as it obtained the highest recall and F1 score. It also used less training time compared to the C5.0 decision tree. Figure 9 shows the training time, accuracy, precision, recall, and F1 score of the Bernoulli NB corresponding with the binarize threshold. The highest recall and the F1 score are obtained by the threshold value of 0.09. Therefore, the threshold of 0.09 is chosen for the Bernoulli NB. The performance of the NB model as

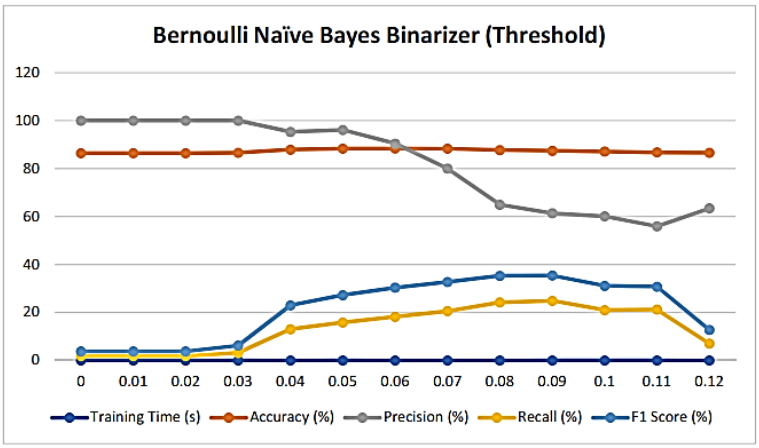

Figure 9: Threshold graph for Bernoulli Naïve Bayes Binarizer, N=1000

indicated from Table 6 shows Bernoulli naïve Bayes to outperform the other models. The multinomial NB obtained 0.00 for the measured metrics because it classified all the sentences into negative and not suitable for the classification task in text summarization which is an imbalance class task.

Table 6: Results of NB using different Distributions

NB	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
Gaussian NB	0.0331	34.1317	12.2318	18.0095
Multinomial NB	**0.0321**	0.00	0.00	0.00
Bernoulli NB	0.0466	**61.3757**	**24.8927**	**35.4198**

4.2 The sentence Feature vector for N=10000

We further trained the models with 8000 documents and validated with 1000 documents with the sentence feature vector as the feature extraction method as presented in Tables 7, 8 and 9. SVM (sigmoid) obtained the highest score in recall and F1 score in Table 7. The smallest training time is obtained by the LIBLINEAR model however the training time of the LIBSVM model with sigmoid kernel still owns the second-least training time.

Table 8 shows the C5.0 decision tree as the champion model with the highest F1 score and precision, however, its training time is high. The CART decision tree still owns its advantage in its recall which covers the actual summary sentences higher than the C5.0 decision tree. Figure 10 shows the same Bernoulli NB with a corresponding binarize threshold of 0.09 while its performance nalysis is presented in Table 9.

Table 7: Results of SVM using different kernels, N=10000

SVM Kernel	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
Linear	**6.7329**	86.9110	3.4764	6.6855
Polynomial (Degree = 2)	7241.6230	**95.3757**	3.4555	6.6694
Radial Basis Function (RBF)	3527.9263	88.0315	11.7068	20.6654
Sigmoid	978.7918	26.3484	**26.7016**	**26.5238**

Table 8: Results of DT using different Tree methods, N=10000

DT	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
C5.0	10.0586	**82.6087**	24.2723	**37.5202**
CART	**3.7237**	34.8721	**36.5445**	35.6887

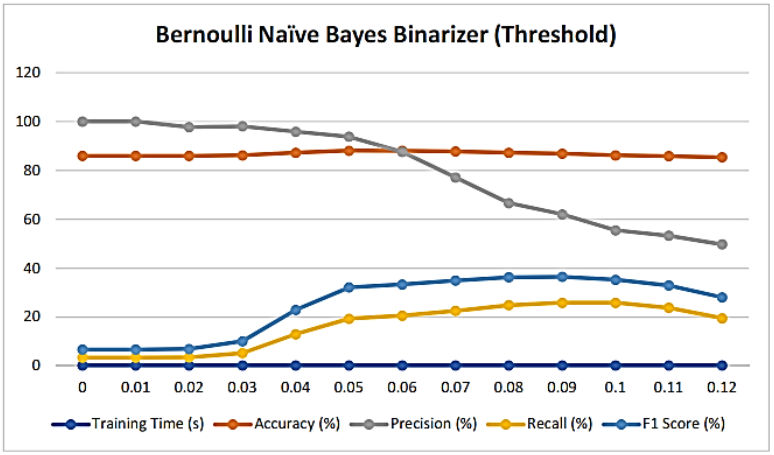

Figure 10: Threshold graph for Bernoulli Naïve Bayes Binarizer, N=10000

In addition, the result of training the ML models with 800 documents and validated with 100 documents using word-based tf-idf vector feature extraction method is discussed as follows. Table 10 shows that the SVM model with the RBF kernel is the champion. The training time of the RBF and sigmoid kernel is high since it needs a lot of computational force to transform the data points to higher dimension space. CART model outperforms C5.0 in Table 11. C5.0 is not suitable for this feature extraction as it obtained a value of zero. The champion model NB algorithm is the gaussian NB in Table 12 as it owns the highest recall and F1 score and the least training time.

Table 9: Results of NB using different Distributions, N=10000

NB	Training Time (s)	Precision (%)	Recall (%)	F1 Score (%)
Gaussian NB	0.3519	40.3261	15.5393	22.4339
Multinomial NB	0.3666	0.0000	0.0000	0.0000
Bernoulli NB	**0.3285**	**62.0482**	**25.8848**	**36.5302**

4.3 Word-Based IF-IDF Vector for N=1000

Table 10: Results of SVM using Word-Based IF-IDF

SVM Kernel	Training Time(s)	Precision(%)	Recall(%)	F1 Score(%)
Linear	**0.1822**	**96.9231**	13.5193	23.7288
Polynomial (Degree = 2)	179.3675	**96.9231**	13.5193	23.7288
Polynomial (Degree = 3)	178.1661	**96.9231**	13.5193	23.7288
Radial Basis Function (RBF)	186.8500	94.7368	**15.4506**	**26.5683**
Sigmoid	201.1984	**96.9231**	13.5193	23.7288

Table 11: Results of DT using Word-Based IF-IDF

DT	Training Time(s)	Precision(%)	Recall(%)	F1 Score(%)
C5.0	86.1557	0.00	0.00	0.00
CART	**9.9879**	**96.9231**	**13.5193**	**23.7288**

Table 12: Results of NB using Word-Based IF-IDF

NB	Training Time(s)	Precision(%)	Recall(%)	F1 Score(%)
Gaussian NB	**0.2215**	96.9697	**13.7339**	**24.0602**
Multinomial NB	0.2851	**100.0**	0.8584	1.7021
Bernoulli NB	0.3328	96.875	13.3407	23.3962

Moreover, RF model had alongside been carried out on the previous experiments (for N=1000, N=10000 and Word-Based IF-IDF). Since the idea is to compare with overall performances of each model adopted, its preferable RF results is presented alongside others as shown in Table 13 for better comparison. The overall performance of each model in each category of sentence/word length used shows Bernoulli NB has the overall best F1 Score with the least training time for N=1000, while RF obtained the best performance for N=10000. Though the SVM (sigmoid) obtained the highest recall shows that the actual summary is mostly covered in its predicted summary. For word-based IF-IDF, N=1000, SVM with RBF kernel gains the highest performance. The gaussian naïve Bayes is also a good model to perform since it uses the least time to compute with a relatively good result.

Table 13: Overall performance of each Model in each Category

ML Models	Training Time(s)	Precision(%)	Recall(%)	F1 Score(%)
N=1000				
SVM (Sigmoid)	8.8	16.241	15.022	15.608
NB (Bernoulli)	**0.0466**	61.3757	24.8927	**35.4198**
DT (CART)	0.2740	30.2231	**31.9742**	31.0740
RF	4.4134	**80.8000**	21.6738	34.1794
N=10000				
SVM (Sigmoid)	978.7918	26.3484	**26.7016**	26.5238
NB (Bernoulli)	**0.3285**	62.0482	25.8848	36.5302
DT (C5.0)	10.0586	**82.6087**	24.2723	37.5202
RF	67.7462	80.1061	25.2984	**38.4530**
Word-Based IF-IDF Vector for N=1000				
SVM (RBF)	186.8500	94.7368	**15.4506**	**26.5683**
NB (Gaussian)	**0.2215**	96.9697	13.7339	24.0602
DT (CART)	9.9879	96.9231	13.5193	23.7288
RF	24.7293	96.9231	13.5193	23.7288

4.4 Model Evaluation on Test Data

The evaluation of each champion model performance with the extractive summarization and feature extraction is presented in table 14. The table shows that the word-based IF-IDF feature extraction is poor for text summarization in ML. The Bernoulli Naïve Bayes performs well in the test data even it only trained with 1000 training data.

Table 14: Model Evaluation on Test Data

ML Model	Rouge-1	Rouge-2	Rouge-L	Test Data (N)	Features Extract
Random Forest	8.2845	2.884	7.9694	10,000	Sentence Feature
Naïve Bayes (Bernoulli)	2.8015	0.2206	0.2855	1000	Sentence Feature
SVM (Rbf)	1.1009	0.0003	1.1753	1000	Word-based TF-IDF

5 CONCLUSION

In this paper, text summarization and its concepts are explored and performed with CNN/Daily Mail dataset. The analysis of the dataset for news articles is visualized and tabulated. Further to this, the performance of each chosen ML model in the classification task for text summarization has been investigated. The champion model for the extractive summarization for the CNN/Daily Mail dataset is identified as RF classifier having the quality of the generated summary score for Rouge-l, Rouge-2 and Rouge-L as 8.2845, 2.884, 7.9694 respectively. In addition, sentence feature vector representation provides with best sentence representation for feature extraction. For future work, an improvement in feature extraction can be conducted for the text summarization in CNN/Daily Mail dataset with a different combination of features and text preprocessing to improve on pronouns, conjunction words, and so on. Also worthy of note is the use of deep learning model or encoders for pretraining the text summarisation earlier mentioned [11]. This could further be considered in future work.

ACKNOWLEDGEMENT

This work is supported by the Universiti Sains Malaysia, Short Term Grant [Grant Number: 304 / PKOMP / 6315616], with the project entitled “New Coefficient of Variation Control Charts based on Variable Charting Statistics in Industry 4.0 for the Quality Smart Manufacturing and Services”.

REFERENCES

[1] M. Gambhir and V. Gupta, “Recent automatic text summarization techniques: A survey,” *Artif Intell Rev*, vol. 47, pp. 1–66, 2017, doi: https://doi.org/10.1007/s10462-016-9475-9

[2] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D., J. B., and K. Kochut, “Text summarization techniques: A brief survey,” International Journal of Advanced Computer Science and Applications, vol. 8, no. 10, 2017.

[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A comprehensive survey,” Expert Systems with Applications, vol. 165, p. 113679, 2021, doi: https://doi.org/10.1016/j.eswa.2020.113679

[4] O. Tas and F. Kiyani, “A survey automatic text summarization,” PressAcademia Procedia, vol. 5, no. 1, pp. 205-213, 2017, doi:10.17261/Pressacademia.2017.591.

[5] G. Silva, R. Ferreira,R. D. Lins, L. Cabral, H. Oliveira, S. J. Simske, and M. Riss, “Automatic text document summarization based on machine learning,” in Proceedings - 2015 ACM Symposium on Document Engineering, 2015, Lausanne, Switzerland, doi: https://doi.org/10.1145/2682571.2797099.

[6] M. I. Hashem, "Improvement of Email Summarization Using Statistical Based Method," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 3, no. 2, pp. 382-388, 2014.

[7] M. Patel, A. Chokshi, S. Vyas, and K. Maurya, "Machine Learning Approach for Automatic Text Summarization Using Neural Networks," International Journal of Advanced Research in Computer and Communication Engineering, vol. 7, no. 1, 2018.

[8] D. Parveen, M. Mesgar, and M. Strube, “Generating coherent summaries of scientific articles using coherence patterns,” in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.

[9] W. S. El-Kassas et al.,, “Edgesumm: Graph-based framework for automatic text summarization,” Information Processing & Management, vol. 57, no. 6, p. 102264, 2020.

[10] M. G. Ozsoy, N. A. Ferda, and C. Ilyas, "Text summarization using latent semantic analysis," Journal of Information Science, vol. 37, no. 4, pp. 405-417, 2011.

[11] Y. Liu and M. Lapata, “Text summarization with pretrained encoders,” 2019, arXiv preprint, arXiv:1908.08345.

[12] M. Bashir, R. Azilawati, and W. I. Wan Malini, "Automatic Hausa language text summarization based on feature extraction using Naive Bayes Model," World Applied Science Journal, vol. 35, no. 9, pp. 2074-2080, 2017.

[13] S. Parthasarathy and T. Hasan, "Automatic broadcast news summarization via rank classifiers and crowdsourced annotation," in Proceedings - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5256-5260, doi: 10.1109/ICASSP.2015.7178974.

[14] X. Wu et al., "News filtering and summarization on the web," IEEE Intelligent Systems, vol. 25, no. 5, pp. 68-76, 2010.

[15] A. John and M. Wilscy, "Random forest classifier based multi-document summarization system," in 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2013, pp. 31-36, doi: 10.1109/RAICS.2013.6745442.

[16] O. Tas and F. Kiyani, "A survey automatic text summarization," PressAcademia Procedia, vol. 5, pp. 204–213, 2017.

[17] M. Desai and J. Baxi, "A Survey on Various Techniques to build Extractive Text Summarizer," International Journal of Emerging Technologies and Innovative Research (www.jetir.org), vol. 5, no. 12, pp. 495-500, 2018, doi:http://www.jetir.org/papers/JETIR1812970.pdf.

[18] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, 2012.

[19] D. Harman and P. Over, "The effects of human variation in DUC summarization evaluation," in Proceedings of the Text Summarization Branches Out Workshop, Barcelona, SP, 2004.

[20] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

[21] A. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," Comput. Sci, 2015, doi:10.18653/v1/D15-1044.

[22] D. Graff and C. Cieri, "English Gigaword LDC2003T05," Philadelphia: Linguistic Data Consortium, 2003. [Online]. Available: https://doi.org/10.35111/0z6y-q265.

[23] E. Sandhaus, "The New York Times Annotated Corpus LDC2008T19," Philadelphia: Linguistic Data Consortium, 2008. [Online]. Available: https://doi.org/10.35111/77ba-9x74.

[24] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-Based Single Document Summarization with Compression and Anaphoricity Constraints," 2016, arXiv:1603.08887.

[25] K. Hong, and A. Nenkova, "Improving the Estimation of Word Importance for News Multi-Document Summarization," in Proceedings - 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL) 2014, 2014, pp. 712-721. doi:10.3115/v1/E14-1075.

[26] R. Paulus, C. Xiong and R. Socher, "A Deep Reinforced Model for Abstractive Summarization," 2017, arXiv:1705.04304.

[27] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in Proceedings - of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, vol. 1, 2015.

[28] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive Summarization as Text Matching," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[29] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL, 2018, pp. 708-719.

[30] V. Gurusamy and S. Kannan, "Preprocessing Techniques for Text Mining," 2014. [Online]. Available:https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining.

[31] A. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," International Journal of Computer Science and Information Security, vol. 16, pp. 22-32, 2018.

[32] S. M. Patel, V. K. Dabhi, and H. B. Prajapati, "Extractive Based Automatic Text Summarization," Journal of Computers, vol. 12, pp. 550-563, 2017, doi:10.17706/jcp.12.6.550-563.

[33] S. Kadhem and Z. Ali, "Multi-Document Text Summarization Based on Multiple Linear Regression," Al-Mansour Journal, 2018.

[34] D. Marcu, "The Rhetorical Parsing of Natural Language Texts," in Proceedings of the 35th annual meeting on Association for Computational Linguistics, 1997, doi:10.3115/976909.979630.

[35] Y. Ledeneva, "Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences," in MICAI 2008: Advances in Artificial Intelligence, in Lecture Notes in Computer Science, vol 5317, A. Gelbukh and E. F. Morales, Eds, Berlin, Heidelberg: Springer, 2008, doi: https://doi.org/10.1007/978-3-540-88636-5_11.

[36] S. Bird, E. Klein, and E. Loper, Natural language processing with python. Sebastopol: O'reilly, 2009.

[37] M. F. Porter, "An algorithm for suffix stripping," Program: Electronic Library And Information Systems, vol. 40, no. 3, pp. 211-218, 2006, doi:10.1108/00330330610681286.

[38] L. Skorkovská, "Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering," Text, Speech and Dialogue , vol. 7499, 2012.

[39] "Expert Guide on How to Summarize Newspaper Articles." 2017. Slideshare. https://www.slideshare.net/summarizing-Biz/expert-guide-on-how-tosummarize newspaper-articles.

[40] S. A. Babar and P. D. Patil, "Improving performance of text summarization," Procedia Computer Science, vol. 46, pp. 354–363, 2015.

[41] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," Comput. Linguist., vol. 32, no. 4, pp. 485–525, 2006, doi:10.1162/coli.2006.32.4.485.

[42] C. -C. Chang and C. -J. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 21-27, 2011.

[43] C. D. Manning, P. Raghavan, and H. Schuetze, Introduction to Information Retrieval. Cambridge University Press, 2008, pp. 234-265.

[44] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[45] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001. doi:10.1023/A:1010933404324.

[46] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, 2002, doi:10.3115/1073083.1073135.

[47] C. -Y. Lin, "Rouge: A Package for Automatic Evaluation of summaries," in Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 2004.