# EASESUM: an online abstractive and extractive text summarizer using deep learning technique

**Jide Kehinde Adeniyi[1], Sunday Adeola Ajagbe[2,3], Abidemi Emmanuel Adeniyi[4], Halleluyah Oluwatobi Aworinde[4], Peace Busola Falola[5], Matthew Olusegun Adigun[2]**
[1]Department of Computer Science, Landmark University, Omu-Aran, Nigeria
[2]Department of Computer Science, University of Zululand, Kwadlangezwa, South Africa
[3]Department of Computer Engineering, First Technical University, Ibadan, Nigeria
[4]College of Computing and Communication Studies, Bowen University, Osun, Nigeria
[5]Department of Computer Sciences, Precious Cornerstone University, Ibadan, Nigeria

**Article Info**

***Article history:***

Received Mar 18, 2023
Revised Aug 30, 2023
Accepted Nov 7, 2023

***Keywords:***

Abstractive summarizer
Artificial intelligence
Deep learning
Extractive summarizer
Natural language processing
Text summarizer

**ABSTRACT**

Large volumes of information are generated daily, making it challenging to manage such information. This is due to redundancy and the type of data available, most of which needs to be more structured and increases the amount of search time. Text summarization systems are considered a real solution to this vast amount of data because they are used for document compression and reduction. Text summarization keeps the relevant information and eliminates the text's non-relevant parts. This study uses two types of summarizers: extractive text summarizers and abstractive text summarizers. The text rank algorithm was used to implement the extractive summarizer, while bi-directional recurrent neural network (RNN) was used to implement the abstractive text summarizer. To improve the quality of summaries produced, word embedding was also used. For the evaluation of the summarizers, the recall-oriented understudy for gisting evaluation (ROUGE) evaluation system was used. ROUGE contrasts summaries created by hand versus those created automatically. For study, a summarizer was implemented as a web application. The average ROUGE recall score ranging from 30.00 to 60.00 for abstractive summarizer and 0.75 to 0.82 for extractive text showed an encouraging result.



***Corresponding Author:***

Sunday Adeola Ajagbe
Department of Computer Science, University of Zululand
Kwadlangezwa 3886, South Africa
Email: saajagbe@pgschool.lautech.edu.ng

## 1. INTRODUCTION

Over the years, there has been a drastic increase in the data generated daily [1], [2]. The global data sector is projected to reach 175 zettabytes by 2025, according to the International Data Corporation (IDC) in its data age 2025 analysis for Seagate [3], [4]. This increase in data has been attributed to technological advancement and datafication of the world; which resulted in the birth of big data [5]. Data are either structured or unstructured. Unlike its amorphous form, that usually includes text and multimedia, structured data are more organized (usually in a tabular form). A significant portion of the data generated is unstructured, necessitating a study in unstructured data analytics. Unstructured data contains many irregularities and ambiguities; therefore, it needs to be analyzed to draw meaningful insights. Manually manipulating and compressing unstructured data is highly time-intensive and cannot keep up with the increasing data every day, hence the introduction of electronic means [1]. Unstructured data is also easier to process using conventional methods than structured data. Hence, it must be converted into machine language, which involves long codes humans cannot understand. Text mining and natural language processing are essential in overcoming this obstacle.

Text mining, also known as knowledge discovery, involves deriving insight and looking for patterns in textual data. Information is inherent in a document, unknown, unidentified, and can hardly be derived without automatic data mining techniques. Automatic text summarization is a subfield of data mining and natural language processing concerned with extracting meaningful information from textual documents [6]. Automatic text summarization is substantially different from that human-based text summarization, as humans can identify and connect significant meanings and patterns in text documents [7].

Text summarization could be categorized as extractive or abstractive using the output of the summary procedure [8]. The extractive text summarization's output uses sentences from the original manuscript. When abstractive text summarization is applied, the resulting summary solely contains concepts from the original text. In literature, more study has been conducted in extractive text summarization [9], [10]. Text summarization could also be categorized in terms of their approaches. The approaches used in text classification include feature-based, which uses statistical methods to determine the level of importance of a sentence in a text. The latent Semantic Analysis based method also reduces sentence vector dimension using singular value decomposition [11]. The topic-based technique uses the topic in the sentence to rate the sentence's value. The relevance measure considers statistical similarity to assign levels for the inclusion of a sentence in a summary. The graph-based method generates a graph using the input text and ranks the sentence using the graph. The template-based method generates templates from the input text and uses the template for summarization. The more recent machine learning-based approach [10], [12], [13] uses machine learning algorithms for text summarization. This study examines the use graph-based approach and deep learning approach to summarize text documents online with little loss of the document's ideas.

Various techniques have been utilized for abstractive text summarization. This study contributes to the body of knowledge by using the text rank algorithm to implement the extractive summarizer. In contrast, bi-directional recurrent neural network (RNN) was used to implement the abstractive text summarizer. Furthermore, word embedding was used to improve the quality of the summaries produced. The average recall-oriented understudy for gisting evaluation (ROUGE) recall score ranging from 30.00 to 60.00 for abstractive summarizer and 0.75 to 0.82 for extractive text showed an encouraging result compared to the state-of-the-art results.

The study has the potential to provide significant benefits to users by helping them save time, improve comprehension, make better-informed decisions, and keep up with the ever-increasing amount of information available online. Also, help users make better-informed decisions by providing them with a concise overview of the information they need to consider. This can be particularly useful for businesses, policymakers, and others who need to make decisions based on large amounts of data. The motivation is to help readers understand complex material by breaking it down into more manageable chunks. This can be particularly useful for people who are not experts in a particular field or for those who have limited time to read.

The remaining section in this paper include a literature review that examines relevant literature. The methodology comes after a literature review, and it examines the methods used in the proposed system. Results and discussion examine the result obtained in the paper and the implication of these results. The last section of this paper concludes the paper and shows the possible area of future work.

## 2. RELATED STUDIES

Several studies have been conducted on the summarization of text. They can broadly be categorized into extractive and abstractive text summarization. In this section, we examines literatures of both categories of text summarization.

### 2.1. Extractive text summarization

Researchers have examined extractive text summarization from different view using different methods in the past. Among these researchers is Li *et al.* [14]. In this article, to create extractive summaries, a deep learning data-driven method was utilized. To decide whether or not sentences should be included in the summary, paraphrasing methods were used. A convolutional layer was used to generate a feature map in the model, as well as densely connected layers of neurons. Since summary generation is a binary classification issue, two scores for each class were created for each phrase, and precision, recall, accuracy, and F-measure metrics were used to evaluate instead of ROUGE. From evaluation, it was observed that the accuracy recorded was above 90% while the other evaluation metrics were low. This was because the dataset was based on human summaries. Kumar *et al.* [15] introduced a model for building a network in which text phrases are depicted as nodes and the relationship between different sentences were represented as the weight of the edge linking them. In contrast to traditional cosine similarity, which treats words identically, a modified reversed sentence frequency-cosine similarity was constructed to assign various weights to distinct terms in the document. The graph was sparsely subdivided into various categories. It operates on the premise that sentences inside a cluster are similar to one another. The performance evaluation of the proposed summarization technique indicated it to be effective. Jang and Kang [12] examined extractive summarization using graph-based approach. The approach considered the degree to which nodes on the edges of the graph are similar. Also, weights were distributed based on the similarity with the topic. Semantic measure was also used for finding the similarity between nodes. The method proposed produced a precision, recall, and F-measure of 0.154, 0.229, and 0.445 respectively.

Liu *et al.* [16] examined multi-document text summarization with the use of firefly algorithm. Their fitness function introduced three features which include the readability, coherence, and topic relation factors. The proposed system was evaluated using the ROUGE score and a comparison with other nature inspired algorithm like particle swarm optimization and genetic algorithm. The proposed method showed produced ROUGE-1 recall, precision, and F-score of 0.43803, 0.48095, and 0.47821. ROUGE-2 result of 0.21212, 0.25012, and 0.22951 respectively. Ajagbe *et al.* [17] considered the use of common hand-crafted features for text summarization in multiple documents. The number of sentences, phrase frequency, title similarity, sentence position, sentence length, sentence-sentence frequency, and other characteristics are among these features. Two fuzzy inference systems and a multilayer perceptron were utilized for phrase extraction and document understanding after various combinations of these features were looked at. The recall, precision, and F-score of 0.409, 0.512 and 0.370 for Rouge-1. Rouge-2 also produced a recall, precision and f-score of 0.290, 0.360, and 0.264.

Bhuiyan *et al.* [18] presented a document summarization technique using quantum inspired genetic algorithm. In their method, the preprocessing steps include sentence segmentation, tokenization, removal of stop words, case folding, tagging of parts-of-speech, and stemming. Sentence scoring made use of statistical features, sentence-to-document and sentence-to-title cosine similarity, and quantum inspired genetic algorithm. The result showed a recall, precision, and F-score of 0.4779, 0.4757, and 0.4767 for ROUGE-1 respectively. A recall, precision, and F-score of 0.1289, 0.1286, and 0.1287 was also recorded for ROUGE-2 respectively. Mallick *et al.* [19] presented an approach to unsupervised extractive text summarization. The system used sentence graph, generated from each document automatically. The method was extended from single document to multi-document by using both document graph and proximity-base cross-document edges.

Mattupalli *et al.* [20] proposed an unsupervised extractive summarization model called learning free integer programming summarizer. Their approach prevents the gruesome training stage required for supervised extractive summarizing methods. In their system, an integer programming problem was formulated from pre-trained sentence embedding vectors. Principal component analysis was used to select sentences to extract from the document. The F1-score obtained for ROUGE-1, ROUGE-2, and ROUGE-L after testing with Wikihow dataset were 24.28, 5.32, and 18.69 respectively. 36.45, 14.29, and 24.56 were the F1-score obtained for ROUGE-1, ROUGE-2, and ROUGE-L for convolutional neural network (CNN) dataset.

### 2.2. Abstractive text summarization

There are methods for abstractive text summarization that have been proposed, similar to extractive summarization. Some of these studies include the study in [21]. The authors addressed the challenge of generating incorrect facts with respect to the actual text in abstractive summarization. To solve this challenge, a suite of two factual correction models called SpanFact was used. The ROUGE score obtained for CNN dataset showed result of 19.27, 41.75, and 38.81 for R-2, R-1, and R-L respectively. Mutlu *et al.* [22] proposed a topic guided abstractive summarization. Their approach ensures a level of dependency on the topic of the text. They included topic modelling with their seq2seq transformer modelling. Testing their proposed system on CNN dataset showed a result of 44.38 for R-1, 21.19 for R-2, and 41.33 for R-L. Patel *et al.* [23] described a method for abstractive text summarization that makes use of generative adversarial networks. Their designed model includes a summary generator and a discriminator. The generator generates the summary and the discriminator tries to seperate a machine generated summary from that of a human. The result obtained showed a score of 37.87, 15.71, and 39.20 for R-1, R-2, and R-L respectively. Chan and King in [24] proposed utilizing long short-term memory (LSTM)-CNN for abstractive text summarization. In their system, phrases were first extracted. After the extraction of the phrases, summary was generated using LSTM-CNN. ROUGE-1 and ROUGE-2 were used as metrics for testing and the result obtained were 34.9 and 17.8. Espino *et al.* [25] proposed a pointer-generator network for abstractive text summarization. Though the network was observed to produce out of vocabulary words, a pre-trained layer of word embedding was presented in solving this. The result showed a score of 39.06, 17.05, and 35.85 for R-1, R-2, and R-L.

## 3. METHOD

### 3.1. Input dataset

For the evaluation the proposed system, two datasets were used. The two datasets are the Amazon food review dataset and the news room dataset. Abstractive text summarization techniques are supervised learning techniques; therefore, they require a labelled corpus (dataset) to be trained on. In this study, the Amazon food review dataset was used. The Amazon fine food reviews dataset is a CSV file in English language, consisting of reviews of fine foods from amazon. It includes 74258 products, 256059 users, and 568,454 reviews. The data was collected between October 1999 and October 2012. This dataset was downloaded from Kaggle and it is available at https://www.kaggle.com/snap/amazon-fine-food-reviews [5], [21].

The Newsroom dataset is a collection of summaries. It has 1.3 million stories and summaries that were written and edited by people working in the newsrooms of 38 major news organizations. This high-quality text, which was extracted from search and social media information between 1998 and 2017, shows a wide range of use in text summarization. The dataset is available at Cornell University's dataset repository [22]–[24]. Figure 1 shows the proposed system's block diagram.

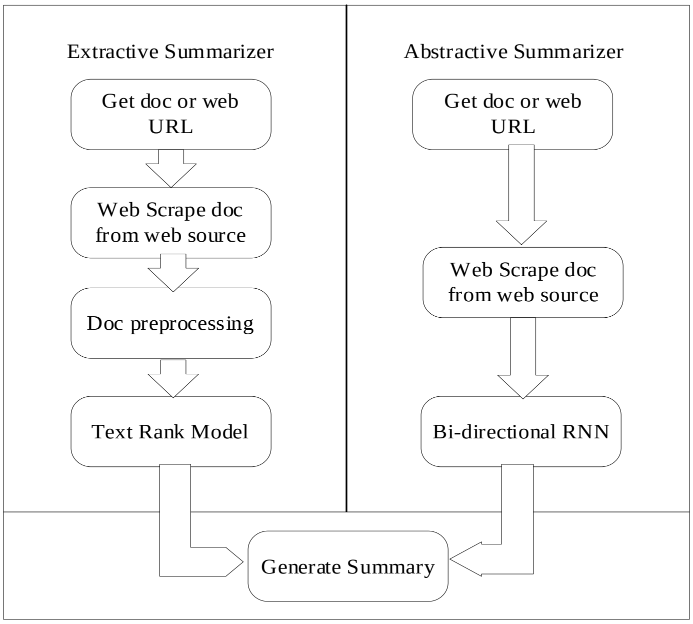


Figure 1. The proposed system’s block diagram

### 3.2. Data preprocessing

Preprocessing is a step that prepares the dataset for classification. For the proposed system, the following preprocessing tasks were carried out: data cleaning, tokenization, and word embedding. The details of each step is examined as follows.

#### 3.2.1. Data cleansing

Data cleaning is the process of preparing data for analysis by removing or altering information that is incorrect, lacking, unnecessary, redundant, or poorly structured. When it comes to natural language processing, data cleaning is usually required because it can improve the data before it is fed into the model [26]. Data cleaning aids in text normalization. In this study, the following processes were carried out to clean the data: i) converting text to lowercase, ii) text splitting (tokenization), iii) removal of punctuations in text, iv) removal of special characters in the text, and v) use of contraction mapping to replace contracted words of the language with their full form. The pseudocode for the data cleansing process is shown in Pseudocode 1.

```
Pseudocode 1: Data cleansing process
Input: sentence to summarize
For (Alphabet in the sentences)
        If (alphabet is Uppercase)
                Convert to LowerCase
        end
end
for- each (Sentence)
        extract the words
```

```
        if (word is punctuation or special character)
                remove from the sentence
        else
                add to the list of words
        end
end
Apply contraction mapping
Output: Contracted words
```

#### 3.2.2. Tokenization

Tokenization is the process of breaking down a written document into tiny components called tokens. A token can be a word, a word's fragment, or merely a character, like a period (which has been removed in the cleansing stage). It essentially divides material into little chunks of words and removes the stop word [16]. Tokenization was used to extract the words from the sentence.

### 3.3. Model development

The proposed system uses two models for summarization. The first model is for extractive text summarization and the second is for abstractive text summarization. The two models are examined as follows.

#### 3.3.1. Extractive text summarizer

For implementing this model, global vector (GloVe) word embedding was used. The model takes in word of the text as input, extracts the vector, creates its similarity matrix using cosine distance, and builds a graph. After the graph has been built, the PageRank algorithm is applied and the sentences are ranked. Sentences with a higher ranking are extracted and are included in the summary. The steps followed by the extractive text summarizer is presented as follows.

Word embedding: analysing natural language text and extracting usable information from a particular word or phrase using machine learning and deep learning approaches necessitates converting the text into a set of real integers. A natural language processing technique called word embedding, commonly referred to as word vectorization, converts words or sentences from a lexicon into a corresponding vector of real numbers. The output is then used to determine word predictions and word semantics [16], [27]. In this study, GloVe word embeddings was used.

Lexical similarity: there is a need to discover lexical similarities between words in the text after the words have been converted to vectors. Lexical similarity is a metric for comparing two texts that are based on the intersection of word sets from the same or distinct languages. A lexical similarity score of 1 indicates that the vocabularies completely overlap, whereas a score of 0 indicates that there are no shared terms in the two texts.

For this study, cosine similarity was used. Due to its effectiveness, cosine similarity was used to compare the similarity of two vectors in an inner product space [16], [28]. By computing the cosine of the angle created by two n-dimensional vectors projected in a multidimensional space, it may identify if two vectors are moving in the same direction. A score around 0 implies less resemblance, whereas a score around 1 shows greater similarity. Figure 2 shows a graph of the cosine distance and it is expressed as shown in (1).

$$Similarity(D1, D2) = \frac{D1.D2}{||D1||||D2||} \tag{1}$$

where D1, D2 are vectors.

$$||D|| = \sqrt{{D_1}^2 + {D_2}^2 + \cdots {D_n}^2}$$

for a vector of size n.

PageRank algorithm: websites are ranked in search engine results using the PageRank algorithm developed by Google. PageRank was inspired by one of Google's original founders, Larry Page. Using PageRank, one may assess the significance of website pages. By calculating the quantity and caliber of links pointing to a website, PageRank generates an approximate evaluation of its importance. The underlying assumption is that websites with greater authority are more likely to receive links from other websites. Let's say that pages T1 through Tn all point to page A. (i.e., are citations). A variable called the damping factor d has a range of 0 to 1 (usually set around 0.85). The next section contains more information about d. C(A) also refers to the number of links that leave page A. A page's PageRank is calculated using (2) [29]:

$$PR(A) = (1 - d) + d\left(\frac{PR(T1)}{C(T1)}\right) + \cdots + PR(Tn)/C(Tn)) \quad (2)$$

where $Tk = page\ pointing\ to\ the\ page\ A$; $PR(Tk)$=PageRank of the page $Tk$; $d$=a damping factor; and $C(Tk)$=a number of outgoing links of the page $Tk$, k=1,…,n.

Because Page Rank is a probability distribution over online pages, the total PageRank of all web pages will be one. PageRank, or PR(A), is the primary eigenvector of the web's normalized link matrix and may be determined using a simple iterative process. In this study, Page Ranking algorithm was used to rank the sentences not webpages. The algorithm ranks each sentence in order of importance in the text using the number of words in the sentence that appear in the topic of the article.

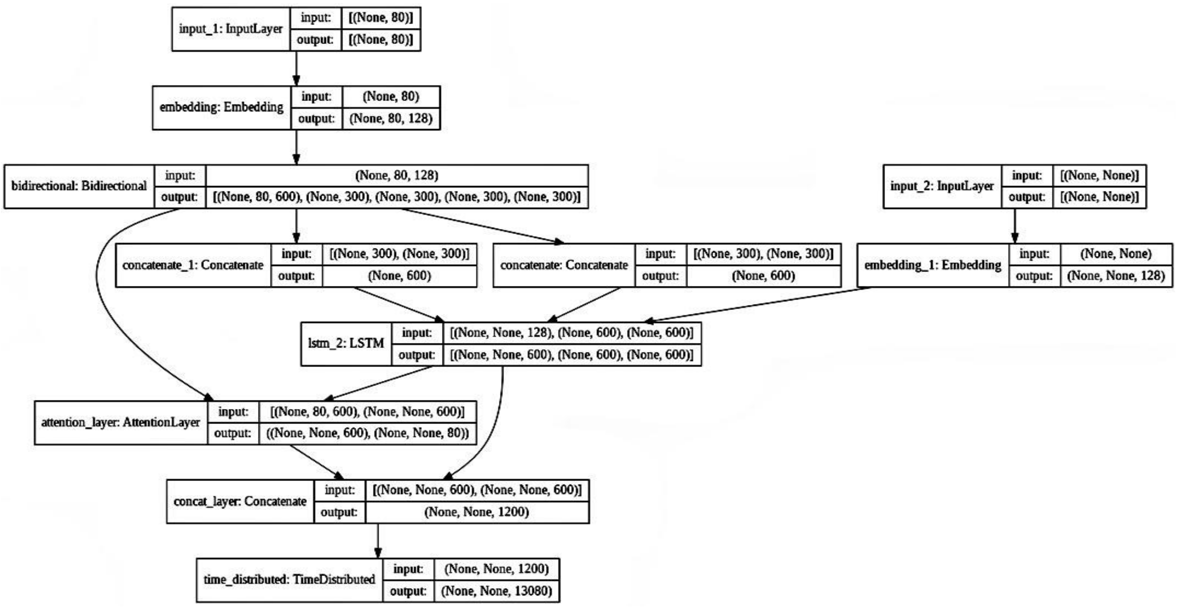


Figure. 2. Abstractive text summarizer model

### 3.3.2. Abstractive text summarizer

For abstractive text summarizer, the hyperparameters that will be used in building the model were tuned. The model is a sequence-to-sequence model using bidirectional RNN. It is made up of the following layers:

- Encoding layer: the encoding layer reads an input token sequence and encodes it into a vector with fixed length for processing. A concept is represented by more than one neuron in the vector form, and one neuron encodes many concepts. It is therefore dense, as opposed to sparse representation, which requires a new dimensionality each time a new idea is added. In this study, the encoder is a bidirectional RNN composed of two separate LSTM; one encodes the information from left-to-right, forward encodes, while the other encodes from right-to-left (backward encoder). Bidirectionality in RNN on the encoder side gives a better document understanding and representation.
- Dense layer: it is a standard neural network layer with many connections. Each neuron gets information from all neurons in the preceding layer, resulting in a highly linked network. It is the most popular and often used layer.
- Attention layer: the attention layer is used to carefully choose important information while eliminating irrelevant information. This layer achieves this by conceptually mapping the produced sentences with the encoder layer's inputs. The Bahdanau attention was utilized in this study.
- Dropout layer: input and recurrent connections to LSTM units are eliminated from activation and weight adjustments made during network training by the normalization technique known as dropout. In this layer, overfitting is reduced while model performance is improved. The dropout layer randomly sets input units to 0 with a rated frequency at each phase of the training process. Inputs that are not set to 0 have their size raised by 1/(1-rate), such that the sum of all inputs remains constant.
- Decoding layer: for the summary, the decoder decodes the text sequence and turns the numeric data into an intelligible word sequence. The likelihood of each target token is simulated for each decoder using a

SoftMax, which converts the decoder outputs into a probability distribution across a fixed-size vocabulary. This likelihood is projected based on the recurrent decoder state and the previously produced token. The encoded interpretations of the source article are sent into the decoder together with a vector called the context vector from the attention layer. Figure 3 shows the abstractive model structure. The model was gotten after tunning several hyper-parameters.

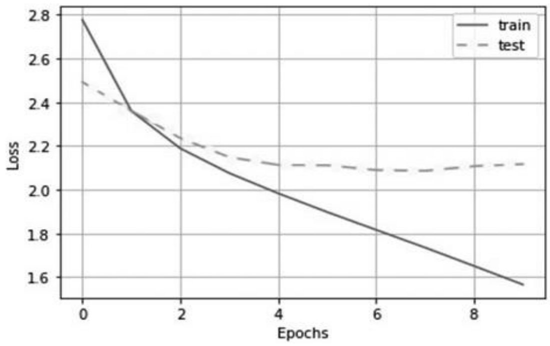


Figure 3. Model loss plot graph

## 4. RESULTS AND DISCUSSION

In supervised machine learning, a machine learning algorithm creates a model by evaluating examples of data supplied to it and generating a model that minimizes loss. The loss parameter reflects how inaccurate the model's prediction was on a given data sample. The model’s prediction is correct if the loss is 0; it becomes otherwise if the loss is greater. Cross-entropy was used as the loss function for the proposed model. A batch size of 128 and an epoch of 50 was used but the training stopped early at the 10th epoch. The model training and model loss are shown in Figures 4 respectively. The implementation was done on a dell with dual core processor i5, each running at 2.3 GHz, 500 GB HDD and 4 Gb of RAM.

**Text:** I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.

**Reference Summary:** The quality of the Vitality canned dog products is good.

**Predicted Summary:** My Labrador is finicky and she appreciates this product better than most.

Figure 4. Test case I

### 4.1. Model evaluation

The evaluation of summary is a subjective task and hence evaluation is a very difficult task. It is a subject of debate as to what makes a summary a good summary. Intrinsic evaluation was utilized in this study. The produced summary is compared to the original text or a reference summary in intrinsic assessment. When compared to a reference summary, it is possible to quantify how effective the system is against humans. Methods for evaluating text quality aim to validate linguistic characteristics of the produced summary such as correct grammatical, reference clarity, and coherence. In this system, the ROUGE evaluation measure was utilized for evaluation. Here are the four quadrants of a confusion matrix that was used to compute the recall, precision, and F1 scores [7], [30], [31]. The abstractive text summarizer model and model for training are presented in Figures 2.

- True positive (TP) is the result of the model’s correct prediction of the positive class.
- True negative (TN) is the result of the model’s accurate prediction of the negative class.
- False positive (FP) is the product of the model’s inaccurate prediction of the positive class.
- False negative (FN) is the outcome of the model incorrectly predicting the negative class.

Recall: Recall is the proportion of right information recovered by a system versus the proportion of erroneous information recovered. The mathematical expression in (3) shows how the recall R is obtained using the TF, TP, and FN [6]. In Figure 3, the model loss plot graph.

$$R = \frac{TF}{TP+FN} \quad (3)$$

Precision: precision is the size of accurate information retrieved by a system in comparison to the amount of incorrect information recovered. The precision P is obtained using (4) [6].

$$P = \frac{TP}{TP+FP} \quad (4)$$

F-score: F-score is a metric that combines accuracy and memory by calculating the harmonic mean of recall and precision. The F1-score, which is an exchange between recall and accuracy, is the most commonly used F-score. F-score is obtained using (5) [6].

$$F = \frac{2TP}{2TP+FP+FN} \quad (5)$$

ROUGE: it is a collection of measures for assessing machine translation and automatic text summarization. The objective is to compare the quality of the resulting summary to a standard document automatically. The goal is to determine the recall by counting the number of units (N-grams) in both the summary and reference systems. Because a text may include numerous summaries, this method enables the usage of multiple reference summaries. ROUGE compares an autonomously generated summary to a collection of pre-set or golden summaries. Many ROUGE variants have been proposed, including ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. For this study, ROUGE-N and ROUGE-L were used for summary evaluation [32], [33].

### 4.2. Extractive summarizer evaluation

The extractive summarizer was evaluated on the newsroom dataset. Test case I is as shown in Figure 4. The performance of the extractive text summary for test case I is shown in Table 1 while the result for test case II is shown in Figure 5. R-1, R-2, and R-L refer to the the ROUGE-1, ROUGE-2, and ROUGE-L score. From the results obtained in Tables 1 and 2, in summary 1 the ROUGE approach doesn't tend to give high scores to the generated summary due to the less common words between the generated summary and the reference summary. To guarantee a good evaluation by ROUGE, the reference result must be taken literally from the corpus with the exact word, or at least to contain the same word style as in the summarized corpus.

Table 1. Extractive text summarizer test case I result

| | Precision | Recall | F1-score |
|---|---|---|---|
| R-1 | 0.100 | 0.0833 | 0.0909 |
| R-2 | 0.000 | 0.0000 | 0.0000 |
| R-L | 0.100 | 0.0833 | 0.0909 |

**Text:** Truffle oil quite good prefer brand France Urbani Italy expensive oh delicious tried black white good black bit stronger pungent event healthy alternative butter enjoy.

**Reference Summary:** Delicious but not the best

**Predicted Summary:** Delicious

Evaluation Result:

Test Case I

**Text:** My husband and I drink about 3 pots of coffee per day and we both have a liking for French Vanilla Creamer. I used to buy it in the powdered, but then the liquid came out and we us ed that. Then now they have this concentrated creamer which is the best yet, we believe it tastes better than the liquid and the powdered. Also, it gives you more for your dollar. The only downside is the pump. Really cool idea if you could only remove it after putting it on. I could understand at a business where you wouldn't want people tampering with it, but the pump leaves a lot of creamer left in the bottom yet. We leave it off and just pour it into the coffee, and when there is only a little left, we pour some coffee in it and get the rest out. Also another plus, the con tainer is recyclable in a lot of areas. Go CoffeeMate!

**Reference Summary:** Great for your money!

**Predicted Summary:** Great.

Test Case IV

Figure 5. Test case extractive test summarization

Table 2. Case II

| | Precision | Recall | F1-score |
|---|---|---|---|
| R-1 | 0.7619 | 0.9697 | 0.8533 |
| R-2 | 0.7000 | 0.9459 | 0.8046 |
| R-L | 0.7619 | 0.9697 | 0.8533 |

### 4.3. Abstractive summarizer evaluation

The abstractive summarizer was evaluated on the amazon food review dataset. The results are shown in Tables 3 and 4. Test case IV shown in Figure 5.

Table 3. Abstractive text summarizer test case I results

| | Precision | Recall | F1-score |
|---|---|---|---|
| R-1 | 0.250 | 1.000 | 0.399 |
| R-2 | 0.000 | 0.000 | 0.000 |
| R-L | 0.250 | 1.000 | 0.399 |

Table 4. Abstractive text summarizer test case II results

| | Precision | Recall | F1-score |
|---|---|---|---|
| R-1 | 0.200 | 1.000 | 0.333 |
| R-2 | 0.000 | 0.000 | 0.000 |
| R-L | 0.200 | 1.000 | 0.333 |

### 4.4. Average ROUGE scores

The average ROUGE score comprises of extractive summarizer and comparison of result. In the extractive summarizer, Evaluation carried out on 50 articles from cornel newsroom dataset showed the result given in Table 5. In comparison of result however, Tables 6 and 7 shows the comparison of the result obtained from this study with other similar studies. The comparison is primarily focused on the ROUGE metrics. This is because it is the most popular in literatures. The comparison in Tables 7 show an improved performance in the ROUGE scores obtained. The ROUGE recall score obtained for the abstractive text summarization shows an improvement when compared with similar systems. It should however be noted that the dataset for which the comparison is based is not the same for the compared papers

Table 5. Extractive text summarizer text case III results

| | Precision | Recall | F1-score |
|---|---|---|---|
| R-1 | 0.650 | 0.823 | 0.739 |
| R-2 | 0.700 | 0.750 | 0.800 |
| R-L | 0.650 | 0.823 | 0.739 |

Table 6. Extractive text summarizer comparison

| Paper | ROUGE-1 | | | ROUGE-2 | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| [12] | 0.229 | 0.154 | 0.445 | - | - | - |
| [16] | 0.43803 | 0.48095 | 0.4784 | 0.212 | 0.25012 | 0.2295 |
| [17] | 0.409 | 0.512 | 0.370 | 0.290 | 0.360 | 0.264 |
| [20] | | | 0.3645 | | | 0.1429 |
| Our system | 0.650 | 0.823 | 0.739 | 0.700 | 0.750 | 0.800 |

Table 7. Abstractive text summarizer comparison

| Paper | ROUGE-1 (%) | ROUGE-2 (%) | ROUGE-L (%) |
|---|---|---|---|
| [21] | 41.74 | 19.27 | 38.81 |
| [22] | 44.38 | 21.19 | 41.33 |
| [23] | 37.87 | 15.71 | 39.20 |
| [25] | 39.06 | 17.05 | 35.85 |
| Our system | 60.00 | 30.00 | 60.00 |

## 5. CONCLUSION AND FUTURE SCOPE

In this study, extractive and abstractive summarizers were implemented as web application. For the extractive text summarizer, the Text rank algorithm was used. For the abstractive text summarizer, a sequence-to-sequence model with a bidirectional RNN was used. For the encoder to understanding the document, word embedding was used. To generate better results, an attention mechanism was also added to the decoder. According to the results of the evaluation, automatically produced summaries are not as logical and intelligent as human summaries, since humans can think about and choose the best option. However, most readers cannot grasp the summary and put them together by applying basic logic. So, if a suitable summarizing approach is employed, automatically generated summaries may be a good substitute for human summaries. It can also make dealing with vast amounts of data much easier and faster. Providing this summarization approach online as done in this study would provide easier access to text summarization. For future studies, the comparison could be made between machine learning techniques. Other ranking algorithms could also be compared with page-rank algorithm to see which is more efficient.

## REFERENCES

[1] I. Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand, and P. K. Soni, "Natural language processing (NLP) based text summarization - a survey," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 1310–1317, doi: 10.1109/ICICT50816.2021.9358703.
[2] N. Alami, M. Meknassi, N. En-nahnahi, Y. El Adlouni, and O. Ammor, "Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling," *Expert Systems with Applications*, vol. 172, 2021, doi: 10.1016/j.eswa.2021.114652.
[3] D. T. Anh and N. T. T. Trang, "Abstractive text summarization using pointer-generator networks with pre-trained word embedding," in *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*, 2019, pp. 473–478, doi: 10.1145/3368926.3369728.
[4] R. Bhargava and Y. Sharma, "Deep extractive text summarization," *Procedia Computer Science*, vol. 167, no. 2019, pp. 138–146, 2020, doi: 10.1016/j.procs.2020.03.191.
[5] R. C. Belwal, S. Rai, and A. Gupta, "A new graph-based extractive text summarization using keywords or topic modeling," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 8975–8990, 2021, doi: 10.1007/s12652-020-02591-x.
[6] Y. Dong, S. Wang, Z. Gan, Y. Cheng, J. C. K. Cheung, and J. Liu, "Multi-fact correction in abstractive text summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9320–9331, doi: 10.18653/v1/2020.emnlp-main.749.
[7] 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, Mar. 2021, doi: 10.1016/j.eswa.2020.113679.
[8] I. D. Oladipo, M. AbdulRaheem, J. B. Awotunde, A. K. Bhoi, E. A. Adeniyi and M. K. Abiodun, "Machine learning and deep learning algorithms for smart cities: a start-of-the-art review," IoT and IoE driven smart cities, pp. 143-162, 2021.
[9] R. Rawat, O. Oki, R. K. Chakrawarti, T. S. Adekunle, J. M. Lukose and S. A. Ajagbe, "Autonomous artificial intelligence systems for fraud detection and forensics in dark web environments," *Informatica*, vol. 47, no. 9, pp. 51-62, 2023, https://doi.org/10.31449/inf.v47i9.4538.
[10] 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, Volume 1 (Long Papers)*, 2018, pp. 708–719, doi: 10.18653/v1/N18-1065.
[11] S. L. Hou *et al.*, "A survey of text summarization approaches based on deep learning," *Journal of Computer Science and Technology*, vol. 36, no. 3, pp. 633–663, 2021, doi: 10.1007/s11390-020-0207-x.
[12] M. Jang and P. Kang, "Learning-free unsupervised extractive summarization model," *IEEE Access*, vol. 9, pp. 14358–14368, 2021, doi: 10.1109/ACCESS.2021.3051237.
[13] B. Jing, Z. You, T. Yang, W. Fan, and H. Tong, "Multiplex graph neural network for extractive text summarization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 133–139, doi: 10.18653/v1/2021.emnlp-main.11.
[14] J. Li, C. Zhang, X. Chen, Y. Hu, and P. Liao, "Survey on automatic text summarization," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 58, no. 1, pp. 1–21, 2021, doi: 10.7544/issn1000-1239.2021.20190785.
[15] Y. Kumar, K. Kaur, and S. Kaur, "Study of automatic text summarization approaches in different languages," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 5897–5929, 2021, doi: 10.1007/s10462-021-09964-4.
[16] J. Liu, D. J. D. Hughes, and Y. Yang, "Unsupervised extractive text summarization with distance-augmented sentence graphs," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2313–2317, doi: 10.1145/3404835.3463111.
[17] S. A. Ajagbe, A. A. Adegun, A. B. Olanrewaju, J. B. Oladosu, and M. O. Adigun, "Performance investigation of two-stage detection techniques using traffic light detection dataset," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 4, pp. 1909–1919, 2023, doi: 10.11591/ijai.v12.i4.pp1909-1919.
[18] M. R. Bhuiyan, M. H. Mahedi, N. Hossain, Z. N. Tumpa, and S. A. Hossain, "An attention based approach for sentiment analysis of food review dataset," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225637.
[19] C. Mallick, A. K. Das, M. Dutta, A. K. Das, and A. Sarkar, "Graph-based text summarization using modified TextRank," in *Soft Computing in Data Analytics*, Singapore: Springer, 2019, pp. 137–146, doi: 10.1007/978-981-13-0514-6_14.
[20] S. Mattupalli, A. Bhandari, and B. . Praveena, "Text summarization using deep learning," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 1, pp. 2663–2667, 2020, doi: 10.35940/ijrte.a3056.059120.
[21] M. Mojrian and S. A. Mirroshandel, "A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA," *Expert Systems with Applications*, vol. 171, 2021, doi: 10.1016/j.eswa.2020.114555.
[22] B. Mutlu, E. A. Sezer, and M. A. Akcayol, "Multi-document extractive text summarization: A comparative assessment on features," *Knowledge-Based Systems*, vol. 183, p. 104848, 2019, doi: 10.1016/j.knosys.2019.07.019.
[23] D. Patel, S. Shah, and H. Chhinkaniwala, "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique," *Expert Systems with Applications*, vol. 134, pp. 167–177, 2019, doi: 10.1016/j.eswa.2019.05.045.
[24] H. P. Chan and I. King, "A condense-then-select strategy for text summarization," *Knowledge-Based Systems*, vol. 227, 2021, doi: 10.1016/j.knosys.2021.107235.
[25] J. Q.- Espino, R. M. Romero-González, and A.-M. Herrera-Navarro, "A deep look into extractive text summarization," *Journal of Computer and Communications*, vol. 9, no. 6, pp. 24–37, 2021, doi: 10.4236/jcc.2021.96002.
[26] M. M. Rahman and F. H. Siddiqui, "Multi-layered attentional peephole convolutional LSTM for abstractive text summarization," *ETRI Journal*, vol. 43, no. 2, pp. 288–298, 2021, doi: 10.4218/etrij.2019-0016.

[27] D. Reinsel, J. Gantz, and J. Rydning, *The digitization of the world - from edge to core*. Needham, Massachusetts: Framingham: International Data Corporation, 2018.
[28] B. Rekabdar, C. Mousas, and B. Gupta, "Generative adversarial network with policy gradient for text summarization," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, 2019, pp. 204–207, doi: 10.1109/ICOSC.2019.8665583.
[29] R. K. Roul, "Topic modeling combined with classification technique for extractive multi-document text summarization," *Soft Computing*, vol. 25, no. 2, pp. 1113–1127, 2021, doi: 10.1007/s00500-020-05207-w.
[30] N. K. Sirohi, D. M. Bansal, and D. S. N. R. Rajan, "Text summarization approaches using machine learning & LSTM," *Revista Gestão Inovação e Tecnologias*, vol. 11, no. 4, pp. 5010–5026, 2021, doi: 10.47059/revistageintec.v11i4.2526.
[31] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 857–875, 2019, doi: 10.1007/s11042-018-5749-3.
[32] T. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translation Journal on Magnetics in Japan*, vol. 2, no. 8, pp. 740–741, 1987, doi: 10.1109/TJMJ.1987.4549593.
[33] S. A. Ajagbe and M. O. Adigun, "Deep learning techniques for detection and prediction of pandemic diseases: a systematic literature review," *Multimedia Tools and Applications*, vol. 83, no. 2, pp. 5893–5927, 2024, doi: 10.1007/s11042-023-15805-z.

## BIOGRAPHIES OF AUTHORS

**Jide Kehinde Adeniyi, Ph.D.** is a lecturer in the Department of Computer Science, College of Pure and Applied Sciences, Landmark University, Omu-Aran, Kwara State, Nigeria. He holds Doctor of Philosophy in Computer Science at the University of Ilorin, Nigeria. His interest's range in various topics in biometrics, computer vision, security, mobile computing, artificial intelligence, and machine learning. He can be contacted at email: adekitos@gmail.com.

**Sunday Adeola Ajagbe** is a Ph.D. candidate at the Department of Computer Science, University of Zululand, South Africa and a lecturer at First Technical University, Ibadan, Nigeria. He obtained M.Sc. and B.Sc. in Information Technology and Communication Technology respectively at the National Open University of Nigeria (NOUN), and his Postgraduate Diploma in Electronics and Electrical Engineering at Ladoke Akintola University of Technology (LAUTECH), Ogbomoso, Nigeria. His specialization includes applied artificial intelligence, natural language processing, information security, data science, and internet of things (IoT). He is also licensed by The Council Regulating Engineering in Nigeria (COREN) as a professional electrical engineer, a student member of the Institute of Electrical and Electronics Engineers (IEEE), and International Association of Engineers (IAENG). He has over ninety (90) publications to his credit in reputable academic databases. He can be contacted at email: 230015266@stu.unizulu.ac.za.

**Abidemi Emmanuel Adeniyi** is currently a lecturer and researcher in the Department of Computer Science, Bowen University, Iwo, Osun State and a Ph.D. student at the University of Ilorin both in Nigeria. His area of research interest is information security, the computational complexity of algorithms, the internet of things, and machine learning. He has published quite a number of research articles in reputable journal outlets. He can be contacted at email: abidemi.adeniyi@bowen.edu.ng.

**Halleluyah Oluwatobi Aworinde, Ph.D.** holds a teaching position with the College of Computing & Communication Studies, Bowen University, Iwo, Nigeria and as well, a research fellow at Computing and Analytics Research Laboratory. He is currently the Director of Digital Services at Bowen University and as well, serve on several committees in the university. He won the 2020 AI Commons Best Problem Documentation Award worth USD1,000 and his poster presentation got the best poster award (2nd) at the 2020 Data Science Nigeria AI Bootcamp. He is the winner of Poster Prize at 2022 Deep Learning Indaba held at Tunis, Tunisia. He is a recipient of Google Africa Scholarship Grant. He can be contacted at email: aworinde.halleluyah@bowen.edu.ng.

**Peace Busola Falola** is a lecturer in the Department of Computer Science, Precious Cornerstone University, Ibadan. She is currently a Ph.D. student at the University of Ibadan, Nigeria. She can be contacted at email: falolapeace@pcu.edu.ng.

**Prof. Matthew Olusegun Adigun** retired in 2020 as a senior professor of Computer Science at the University of Zululand. He obtained his doctorate degree in 1989 from Obafemi Awolowo University, Nigeria; having previously received both Masters in Computer Science (1984) and a Combined Honours degree in Computer Science and Economics (1979) from the same University (when it was known as University of Ife, Nigeria). A very active researcher in software engineering of the wireless internet, he has published widely in the specialised areas of reusability, software product lines, and the engineering of on-demand grid computing-based applications in mobile computing, mobile internet and ad hoc mobile clouds. Recently, his interest in the wireless internet has extended to wireless mesh networking resources and node placements issues, as well as software defined networking issues which covered performance and scalability aspects arising from software defined data centres and cloud/fog/edge computing milieu. He has received both research and teaching recognitions for raising the flag of excellence in historically disadvantaged South African Universities as well as being awarded a 2020 SAICSIT Pioneer of the Year in the Computing Discipline. Currently, he works as a temporary senior professor at the Department of Computer Science, University of Zululand, Kwadlangezwa, South Africa to pursue his recent interest in AI-enabled pandemic response and preparedness. He can be contacted at email: adigunm@unizulu.ac.za.