Efficient Text Summarization Using PSO-Optimized LSTM-Gated RoBERTa Algorithm for Social Media Content Analysis"

M.Vinitha[1], S.Vasundra[2]

1: Research Scholar, Department of Computer science and Engineering,
JNTU Ananthapuramu, India, vinitha@rguktong.ac.in
2: Professor, Department of Computer Science and Engineering,
JNTU Ananthapuramu, India, vasundras.cse@jntua.ac.in

Abstract

Nowadays, social media is an important aspect of news reading, learning, and handling digital information. Increasing detailed information on social media content leads to a vast of time to know the summary of the information. Prevailing techniques and Artificial intelligence handle the text summarization based on the importance of term frequency. Analyzing more feature dependencies like subject, nouns, key terms, and topic modeling makes it more tedious to degrade sentence fragmentation, which leads to low precision and recall rate. To resolve this problem and introduce new sentence-based frequency fragmentation, text summarization is implemented based on PSO optimized LSTM gated RoBERTa Algorithm for social media content extraction. Initially, the text preprocessing is framed to process stop word removal stemming and tokenization to normalize the document data. By reducing the sentence and extracting the important terms by evaluating based on Inverse Term quantum vector frequency evaluation (ITQCF). The sentence term frequency evaluation is carried out using best fit term frequency evaluation using PSO to select the important features. Finally, the optimized auto coders based on the LSTM Gated Robustly Optimized BERT Pretraining Approach (RoBERTa) are used to summarize the content. The Sequence Ranking sentence fragmentation (SRSF) formalizes the sentence depending on the topic to finalize the precision summarization. The proposed system improves the precision rate as well in fragmentation and sentence score to show high performance in F1 score and redundant time complexity.

Keywords: Text summarization, feature extraction and classification, key term. Document frequency, Deep learning, LSTM, social media.

1. Introduction

In the digital era, social media has emerged as a pivotal platform for various activities such as news consumption, learning, and handling digital information. With the abundance of detailed information available on social media, users often find themselves spending a significant amount of time shifting through content to grasp the essence of the information presented [1]. To address this challenge, advanced techniques and Artificial Intelligence (AI) solutions have been developed to streamline the process of text summarization, focusing on the importance of term frequency analysis [2]. Text summarization is the process of distilling a large amount of text into a concise summary that captures the main points and key information. With the rise of social media platforms such as Twitter, Facebook, and Instagram, the volume of text being generated on a daily basis is massive. In order to make sense of this vast amount of information, text summarization using NLP techniques has become essential [3].

The NLP is a branch of artificial intelligence that focuses on the interaction between computers and humans using natural language [4]. By analysing and understanding human language patterns, NLP algorithms can extract important information from text and generate summaries that are both informative and concise [5]. Efficient text summarization based on NLP social media content analysis has numerous applications in today's digital landscape [6]. For example, businesses can use text summarization to quickly analyse customer feedback and reviews on social media platforms [7], allowing them to identify trends and make data-driven decisions. Journalists and news organizations can also benefit from text summarization by quickly summarizing news articles and social media posts to keep up with the latest developments [8, 9].

Traditionally, text summarization techniques have relied on analysing key features such as subject matter, nouns, key terms, and topic modelling. However, these methods can be laborious and prone to sentence fragmentation, resulting in lower precision and recall rates. To overcome these limitations, a novel approach known as sentence-based frequency fragmentation text summarization has been introduced, leveraging a PSO-optimized LSTM gated RoBERTa algorithm specifically tailored for extracting content from social media platforms.

The process begins with text preprocessing, encompassing tasks such as stop word removal, stemming, and tokenization to standardize the document data. Subsequently, the extraction of important terms is carried out through an evaluation based on Inverse Term Quantum Vector Frequency (ITQCF). The evaluation of sentence term frequency is refined through a best-fit term frequency assessment using Particle Swarm Optimization (PSO) to identify crucial features within the text. The core of the summarization process lies in the utilization of an optimized autoencoder based on LSTM Gated RoBERTa, a robustly optimized variant of the BERT pretraining approach. This model effectively synthesizes the extracted information to generate concise summaries. To enhance the coherence and relevance of the summarized content, a Sequence Ranking Sentence Fragmentation (SRSF) technique is employed, tailoring the summarization process to the specific topic at hand.

The proposed system demonstrates significant improvements in precision rates, addressing issues related to sentence fragmentation and enhancing the overall sentence score. This enhancement is reflected in the system's high performance metrics, including F1 score and reduced time complexity. By leveraging cutting-edge AI technologies and innovative methodologies, the system offers a sophisticated solution for efficiently summarizing social media content, catering to the evolving needs of modern information consumers.

2. Literature survey

Social media platforms have revolutionized the way we consume news and information, offering a plethora of content that can be overwhelming to navigate efficiently [10]. Text summarization techniques have been developed to condense this information into concise summaries, aiding users in extracting key insights quickly [11]. However, traditional methods based solely on term frequency may not capture the nuances of the content effectively, leading to fragmented sentences and reduced summarization accuracy. In this paper, proposed a novel approach that leverages the power of AI and optimization algorithms to enhance text summarization for social media content [12].

Amidst the outbreak of the novel coronavirus disease (COVID-19), numerous research institutions, including the Allen Institute for AI, have curated extensive datasets related to the virus. The primary objective behind this effort is to aid the research community and the general public in delving into valuable insights derived from COVID-19 datasets on a broader scale [13]. The Covid-19 Open Research Database (CORD-19) serves as a valuable search engine, offering a semantic search platform dedicated to the CORD-19 dataset. Furthermore, covidex presents a multi-step search feature designed to refine various attributes of the COVID-19 dataset. Notably, researchers referred to as authors in a particular study integrated a Natural Language Processing (NLP)-based clinical inference engine, specifically Well in AI [14], along with a well-suited ranking mechanism to establish a framework with exceptional precision and recall scores for conceptual notions.

In the past, LSTM-based methods have been extensively utilized by research communities across various applications such as image captioning, text classification, entity classification, and speech recognition. LSTM, a variant of recurrent neural networks (RNN), has been a cornerstone in effective text summarization, particularly in abstract summarization tasks [15]. It has demonstrated strong performance in extractive summarization. Building upon this foundation, the authors introduced an innovative approach that leverages a focus mechanism integrated with transformers to predict and summarize the salient aspects of the input accurately [16]. This methodology enhances the summarization and translation processes significantly. The summarization technique predominantly follows an extractive approach, where key elements within the input are efficiently identified through advanced weighting and ranking algorithms [17]. Similarly, in reference [18], the author utilized an ensemble random forest method for software fault prediction using the PROMISE dataset.

In most cases, a multidimensional pruning strategy is used for automatic text summarization. The difficulties presented by opinion texts' vastness and complexity, as well as the complexity of K-means algorithms for opinion text aggregation, are addressed by manifold learning [19]. ROI-1 is improved by 11% and ROI-L by 9% using the recommended MOOTweetSumm approach. A Fuzzy Evolutionary Algorithm for Extracted Text Summary. Learning-based optimization calculates the weighted average of the text pieces, and a human-generated FIS is used to assess the recommended approach [20], yielding the sentence total score. The evaluation uses the CNN, DUC 2001, and DUC 2002 datasets.

The word-based attentional system and deep learning techniques to extract data from text. This method uses the Convolutional Bi-GRU to extract the text's syntactic and semantic associations [21]. The CNN/Daily Mail and DUC 2002 datasets are used for evaluation. With Daily Mail scores of 55.9%, 24.8%, and 53.9% using DUC 2002 dataset and 42.9%, 19.7%, and 39.3% as F1 scores, the suggested technique scores R-1, R-2, and R-L measures as 32.8%, 11.0%, and 27.5%. A deep auto-encoder-based unsupervised extraction text summary framework. The following three criteria form is the basis of SummCoder's summaries [22]. An advanced automatic coding network is used to evaluate sentence content's uniqueness, accuracy, placement, and significance. The similarity between two sentences determines a sentence's uniqueness. The SummCoder approach is assessed by using the TIDSumm dataset.

The Bat Butterfly Optimization (BBO) and layered recurrent neural network (L-RNN) methods were introduced by [23]. These methods improve the classification accuracy, performance for software fault identification.

An unconscious text summarization prototypical that encompasses outdated sequence-to-sequence neural text summarization models using a heading-aware decoder and syntactic enrichment encoder [24]. An encoder within the sentence embedding encodes the syntactic structure and the word information of the sentence. Investigational consequences shown that the deployed methodology beats the summarization base model regarding the ROUGE evaluation methodology and attains instant group presentation analogous to the extraction standard technique. Deep learning approach for text extraction. An attention-layered reinforced learning model serves as the foundation for this approach. The recommended method performs well on significant texts in a standardized dataset, with BLEU and ROUGE values of 0.4 and 0.6, respectively [25].

The convolutional neural networks or CNNs, are used with complex generative algorithms, or CGA to aggregate data. It's a new way to select phrases, in contrast to previous models that often used greedy algorithms [26]. A study on medical datasets demonstrates that the proposed method outperforms competitive models by a factor of two. 5% on average, more similar to the orientation instant. A text extraction method in this text summary that uses the topic modeling methodology based on tagging-located domain awareness (LDA) [27]. A smoothing technique is utilized to create diverse and stimulating summaries. Following this, the diversity of the summary, retention rate, and ROI of the produced summaries are assessed to determine their efficacy. The suggested approach is evaluated by using the English dataset.

In ref [28], the author focused on providing an efficient overview of extensive text collections using two valuable methods: semantic similarity and clustering. Summarizing large amounts of text is time-consuming and difficult, mainly when calculating semantic similarity during summarization. Summarizing the collected text involves in-depth processing and computation to produce the summary. Machine learning algorithms and AI techniques are widely used across various research domains to improve the accuracy and knowledge of the system[29].

3. Proposed methodology

The proposed methodology begins with text preprocessing techniques such as stop word removal, stemming, and tokenization to normalize the document data. Inverse Term Quantum Vector Frequency Evaluation (ITQCF) is then employed to extract important terms, followed by a best-fit term frequency evaluation using Particle Swarm Optimization (PSO) to select key features. Subsequently, an optimized auto-encoder based on LSTM Gated RoBERTa Algorithm is utilized for content summarization. The Sequence Ranking Sentence Fragmentation (SRSF) method is applied to formalize sentence dependencies based on topic relevance, thereby improving precision in summarization.

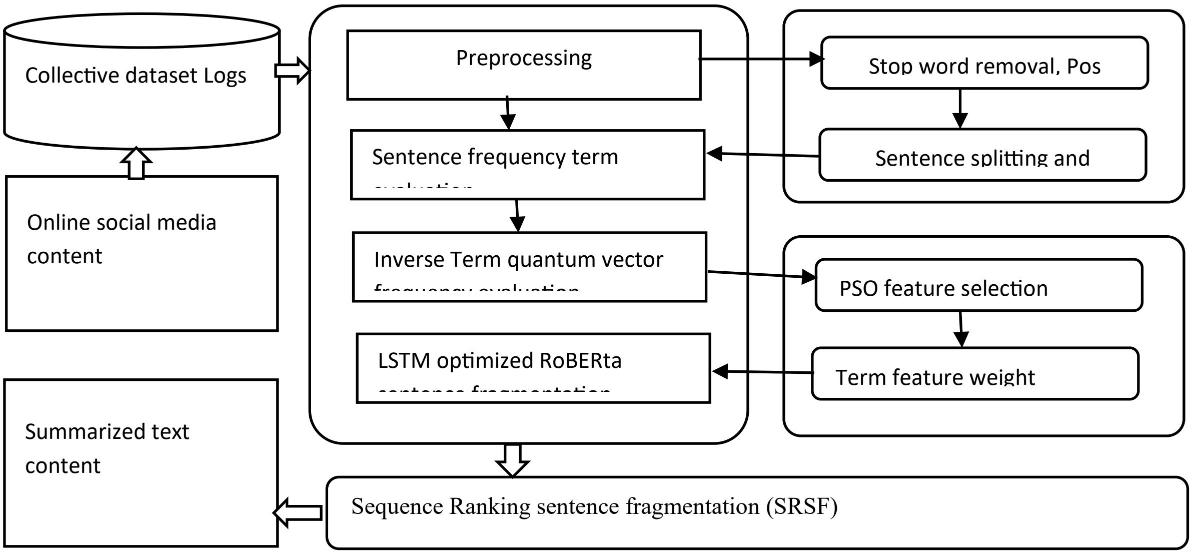


Figure 1 Proposed workflow architecture PSO-LSTMG-RoBERTa

The methodology commences with text preprocessing techniques such as stop word removal, stemming, and tokenization. These methods help in standardizing the document data and removing unnecessary elements that do not add value to the analysis. Next, Inverse Term Quantum Vector Frequency Evaluation (ITQCF) is utilized to identify important terms within the text. This step enables the system to focus on key aspects of the content, leading to more precise summarization. Following the extraction of important terms, a best-fit term frequency evaluation is conducted using Particle Swarm Optimization (PSO). This optimization technique aids in selecting key features that are most relevant to the overall context of the document. By employing PSO, the system is able to prioritize the most important information, thus improving the quality of the summary generated.

The next step in the proposed methodology involves the use of an optimized auto-encoder based on the LSTM Gated RoBERTa Algorithm for content summarization. Auto-encoders are neural network models that are adopt at capturing the latent representation of data, making them ideal for summarization tasks. By incorporating the LSTM Gated RoBERTa Algorithm, which is renowned for its ability to handle long sequences of text, the system is able to produce concise and informative summaries. To further enhance the precision of the summarization process, the Sequence Ranking Sentence Fragmentation (SRSF) method is employed. This method formalizes sentence dependencies based on topic relevance, thereby improving the coherence and fluency of the generated summary. By organizing sentences in a logical manner, SRSF ensures that the summarized content accurately reflects the key points of the original document.

3.1 Data Preprocessing and Indexing

First, in the pre-processing structure, all documents can be viewed as tokenized, and weighted terms are called documents. Each document summary is a minimum weight for the corresponding document containing all keywords based on the indexing terms.

Indexing the document is a pre-processed stage. At this stage, a file is created for each document that contains words without stop words (at, the, the, is, an, etc.). Also, stemming is done to get concepts in their original form (e.g., used, used, available stemming). Each word's frequency is calculated in the next step, and a threshold (based on the formula) is used as code words. At all times in the collection form, create a table for that document. In IR systems, indexing is a technique that makes information retrieval more accurate, faster, and more relevant. Indices can be generated from keywords in stored documents. The first step is to elaborate the key based on the stored database.

All indexing terms are measurements. This indexing term is usually a word of each document. All documents are in a vector of term or weight, and similarity approaches are evaluated as cosine measures by equation (1) – (6),

$$Dx = \left(t_{x1}, t_{x2}, t_{x3}, t_{x4} \ldots\ldots t_{xn}\right) \quad (1)$$

$$Dy = \left(t_{y1}, t_{y2}, t_{y3}, t_{y4} \ldots\ldots t_{yn}\right) \quad (2)$$

Document similarity is described as cosine similarity,

$$S(Dx, Dy) = \frac{\sum_{i=1}^{n} t_{xi} * t_{yi}}{\sqrt{\sum_{i=1}^{n} t_{xi}^{2}} X \sqrt{\sum_{i=1}^{n} t_{yj}^{2}}} \quad (3)$$

The similarity-based Tf-df is used for weighting approaches,

$$wtf - df_{t,d} = wtf_{t,d} \bullet df_t \qquad (4)$$

$$wtf_{t,d} = \begin{cases} 1 + log_{10} tf_{t,d} \\ 0 \end{cases}$$

$$if\ f_{TFT,d}, otherwise$$

$$df = \log_{10}(K/dft) \qquad (5)$$

Finally evaluated the weight,

$$w_{t,d} = (1 + \log_{10} tf_{t,d}) * \log_{10}\left({}^{K}\!/_{dft} \right) \qquad (6)$$

Where, $T_{TFT,d}$- Term frequency of term (t), d-document, dft-document frequency, calculating the terms and document frequency is analysis for similarity approached. The cosine measurements are used for document similarity frequency for terms weights and document indexing values.

3.2 Inverse Term quantum vector frequency evaluation (ITQCF)

After preprocessing, the text frequency term is evaluated based on vector term related to topic modeling to find which term has relevant weight. The ITQCF finds the predominant support vector term words based on frequency contains word by adjusting the sentence inversely proportional to create high lexicon terms. This choice of high lexical terms is relatively based on sentence importance, with meaningful term suggestions. Quantum separates the important key terms in the sequence vector space.

The frequency vector space Tf is estimated by (7)

$$tf(t,d) = \frac{f_t,d}{\sum_{t' \in d} f'_{t,d}} \ \ldots\ldots\ldots.. \qquad (7)$$

Where tf is the rare total of text key term of word t in text d, and t′ are all additional terms inside the text. The inverse document frequency (IDF) indicates how educational the word is and exactly how shared or rare it is amongst the additional text. The inverse frequency is logarithmically evaluated in redundant term with maximum support in frequency limits are evaluated by (8).

$$\text{ITQCF}(t,D) = \log \frac{Max(N)}{|\{d \in D : t \in d\}| + 1} \qquad (8)$$

Where N is the entire number of text in the quantity N =| D |, and | {d ∈ D: t ∈ d} | is the amount of text containing term t, and 1 is supplementary to the denominator to evade division by zero if the term is absent in the corpus. The ITQCF is calculated using the multiplication of equation (7) and (8); and result get it (9)

$$\text{ITQCF(t, d, D)} = \text{tf(t, d)} \times \text{IDF(t, D)} \qquad (9)$$

The frequency correlation terms are extracted by matching the predominant work which is relatively to topic content. The retained words are important to formalize, as key terms to estimate the absolute mean count weight in each sentence. To perform the text summarization task with the proposed method, first extract keyword-based key terms from the given information and rank the documents.

Document Ranking

➢ Selecting the most relevant document D_1, which queries frequently evaluated document weights.

➢ Ranking the other document frequencies of low-weight terms in the document set

Calculating the summary length in equation (10)

$$Le_{Dx} = \left(Le_T - Le_{sq}\right) * \left(0.2 + 0.6^{x-3} / N - 1\right) \qquad (10)$$

To generate a summary, to be given the summary length of each term (Le_T), Length (Le_{Dx}) of sequence $D_x$is calculated based on the terms (sentence) weights, and N is the number of retrieved documents using keywords. Collect keywords used in the searching field and save them in a separate document file to analyze the terms and sentence length.

3.3 Sentence feature analysis- PSO

In the PSO methodology, birds in feature terms are characterized by particles. These particles can be well-thought-out modest agents that "fly" via the space in question. Represents a solution to a particle position problem in a multidimensional issue space. As particles move toward new positions, different solutions to the problem are generated. The fitness r_{best} function of the respective particle in the swarm is evaluated and associated with the fitness of the particle's preceding q_{best} state. After finding two optimal values, the x^{th} particle evolves by optimizing its speed and position according to the following equation:

$$k_{xc} = o * k_{xc} + G_{1*} rand_1 * (q_{best} - a_{xc}) + G_2 * rand_2 * (r_{best} - a_{xc}) \qquad (11)$$

$$a_{xc} = a_{xc} + k_{xc} \qquad (12)$$

Among them, c represents the dimension of the problematic space. Undefined estimates in the range $(0,1)$ of $rand_1$ and $rand_2$ undefined values $rand_1$ and $rand_2$ are depleted for comprehensiveness, i.e. to ensure that the particles explore a large search space already congregating to the ideal resolution. G_1 and G_2 are constants recognized as acceleration coefficients. The estimations of G_1 and G_2 rheostat the weight balance of q_{best} and r_{best} in determining the subsequent move of the particles. o is inertia weight coefficient. An enhancement over the original BSO is that o is not fixed during execution. Instead, it starts from a maximum value and decreases linearly with the amount of iterations, with a minimum assessment initially set to 0.9 and then to 0.4.

$$o = (o - 0.4)\frac{(MAXITER - ITERATION)}{MAXITER} + 0.4 \qquad (13)$$

Max-iteration represents the maximum amount of iterations and signifies the existing amount of iterations. The inertia weight factor w prevents particles from stagnating at local optima by changing their velocities, thereby providing the necessary diversity for the population. The relative sentence by object terms are that progressively diminishing the assessment of inertia weight factor as of a elevated assessment through the search process improves the search efficiency.

Equation 14 states that every particle archives its existing coordinates a_{xc} and velocity f_{xc}, which describes how fast the particle is moving in one dimension in the problematic space. At each generation, the particle's new position is calculated by the accumulation of the particle's existing velocity vector, k, to its position vector, a.

The finest fitness estimations are kept informed at every generation, constructed on

$$Q_x(h+1) = \left\{ \begin{array}{ll} Q_x(h) & f(A_x(h+1) \le f(A_x(h)) \\ A_x(h+1) & f(A_x(h+1) > f(A_x(h)) \end{array} \right\} \tag{14}$$

By means of a population-on the basis search methodology, the PSO considers discrete in the population to be a particle in the search space. Suppose that the position of the xth particle is

$Q_x(h) = (q_x1, q_x2, \dots, q_x, C)$ its velocity is $K_x(h) = (k_x1, k_x2, \dots ,\ k_x, C)$, the ideal location initiates by this particle up to now

$Sm_x(h) = (sm_x1, sm_x2, \dots, sm_x, C)$ the ideal location initiates by the swarm up to now is $Rm_x(h) = (rm_x1, rm_x2, \dots , rm_x, C\ )$ then, this particle is reorganized as respects,

$$\begin{cases} k_{x.y}(h+1) = 0\ x\ k_{x.y}(h) + d_1\ x\ g_1\ x\ \left(sm_{x.y}(h) - q_{x.y}(h)\right) \\ \qquad +d_2\ x\ g_2\ x\ \left(rm_{x.y}(h) - q_{x.y}(h)\right) \\ \qquad q_{x.y}(h+1) = q_{x.y}(h) + k_{x.y}(h+1) \end{cases} \tag{15}$$

where, h is the iteration times, g_1 and g_2 are two acceleration coefficients, d_1 and r_2 are undefined numbers among [0, 1], and inertia weight o of particle proceeding fly velocity.

With the purpose of converting the distinct multi-label feature selection challenge interested in an unremitting issue appropriate for particle swarm optimization, this novel method adopts a practical encoding protocol termed probability-based encoding protocol. This methodology retrieves the odds estimates of the feature selected as a particle encrypting component. Therefore, the particles with the largest number of probability values are the most likely solutions to the problem. If we take particles $Q_x(h) = (q_x1, q_x2, \dots, q_x, C)$ with probability $q_x, c > 0.5, c = 1,2, \dots, C$ then the c -th properties are as follows. Select the consistent feature subcategory. Or else it will not exist.

We adopt two purposes: the number of features as a performance of multi-label cataloging fault and the fitness of the algorithm. Various metrics such as Hamming loss, precision, 1-error, coverage, and rank loss are intended to estimate the classification execution of multi-label classifiers. As with around multi-label classification methodologies, we deploy the Hamming loss ($Hloss$) to estimate the particle classification fault proportion. This is |L| the amount of illustrations L in the test data set, the truth class label set and the label set are the classifier p, b'_x,and $b_x, x = 1,2, \dots, |L|$ predicted by the $Hloss$ is distinct as:

$$Hloss(p, L) = \frac{1}{|L|} \sum_{x=1}^{|L|} \frac{1}{|G|} |b_x \Delta b'_x \tag{16}$$

here, Δ epitomizes the symmetric variance among the two arrays, and |G| represents the number of labels. Therefore, the fitness performance of the particle is written as

$min\, T(Q_x) = (Hloss(Q_x, L), |Q_x|$ (17)

where $|Q_x|$ is the number of features contain by the particle Q_x .

The PSO is noted for its fast convergence speed. However, due to the rapid convergence speed, PSO-based algorithms often converge to the wrong Pareto front. In this novel, the adaptive stochastic mutation is adopted to encompass the search capability of the deployed methodology. This operator uses Q_p iteratively to regulate the mutation odds and threshold of individual particle. By the side of every iteration, initial Q_p is reorganized conferring to the subsequent method.

$Q_p = 0.5 * e^{(-10*h/H)} + 0.01$ (18)

H is the highest number of iterations. We can see that the estimation of Q_p decreases exponentially as the number of iterations upsurges. Next, every particle in the population is investigated in chance. Suppose Q_p is greater than or equal to an undefined number among [0, 1]. In this case, perform a mutation on the existing particle like this: Primary, we arbitrarily select U attributes as of this particle and re-initialize the estimations of these attributes in the search space. At this time, the estimation of U is an integer deployed to regulate the range of variation.

$U = max\{1, \lceil C * Q_p \rceil\}$ (19) The local exploration tactic on the basis of differential learning is deployed to discover regions through scattered resolutions in the search space to progress the execution of the methodology, specifically the self-learning ability of the leading particles in the population. In this approach, foremost, the outcome through a large cluster distance in the archive is designated as the basis vector for differential learning and denoted as A_{best}.

$A'_x = A_{best} + T.(A_{n1} - A_{n2})$ (20)

Then, two undefined resolutions as of the archive (notations A_{n1} and A_{n2}) are set as difference vectors. A state of art outcome is formed by the accumulation of the variance among A_{n1} and A_{n2} to the basis vector A_{best}.

3.4 LSTM gated RoBERTa sentence fragmentation

The key term features can be predicted by processing ordinal data at irregular time intervals using LSTM techniques in the input layer. By iterating the hidden layer outputs, training samples on different time series are efficiently managed within LSTM methods. Moreover, a sigmoid function can be integrated into the model's output layer as an activation function for multi-label output. Furthermore, the LSTM method consists of a single unit with input, output, and forget gate. The cell analyses the estimates at various time points and estimates the stream of info in and out through the three gates. The forgetting threshold can be calculated by the onward circulation of the LSTM network. The input parameters of the oblivion gate are calculated using the time vector of the three-dimensional vector in smooth time intervals. The old cell positions can be updated using the LSTM technique to create temporary positions. The RNN method is an artificial neural model that can identify important terms through the connections between units that form directed loops. Arbitrary embedding arrays can be utilized as input to express dynamic timing behavior based on internal memory networks. Additionally, the hidden units of the RNN model can estimate the length of the input data comprising the output layer and the last time. For each time step's hidden layer output, the heart diagnosis probability is predicted using a sigmoid activation function. The predicted damage at every time phase is derived by incorporating the actual cataloging labels. Moreover, the total predicted loss across every period step and the loss from the final time step can serve as the LSTM-RNN model's loss function for parameter updates in the key term data.

Each gate has a point-wise multiplication function and a sigmoid activation function. Equations 1 through 3 show the results of the elementary unit of LSTM calculation. The forget gate takes the time vector as input, and a smooth computation of the time step yields a three-dimensional vector. Let's assume k-time, σ −logistic sigmoid function, $\mathrm{b_k}$ −output forget gate, $\mathrm{c_k}$ and $\mathrm{H_k}$ −input and output gate, $\mathrm{h_{k-1}}, \mathrm{p_k}$ −input and previous hidden state, z-weight matrices

$$\mathrm{b_k} = \sigma(\mathrm{z_b}[\mathrm{o_{k-1}}, \mathrm{p_k}] + \mathrm{f_b})$$

$$\mathrm{c_k} = \sigma(\mathrm{z_c}[\mathrm{o_{k-1}}, \mathrm{p_k}] + \mathrm{f_c})$$

$$\mathrm{h_k} = \sigma(\mathrm{z_h}[\mathrm{o_{k-1}}, \mathrm{p_k}] + \mathrm{f_h}) \qquad (21)$$

A smoothing vector calculates the time interval between consecutive time slices, as outlined in equations 4 and 5. Let's assume k-time, $\Delta_{\mathrm{k-1}}$ −time interval, $\mathrm{Xb_x}\Delta_{\mathrm{k-1}}$ −time interval smoothening vector, $\mathrm{x_b}$ −weight parameter,

$$\mathrm{b_k} = \sigma(\mathrm{z}[\mathrm{o_k} - 1, \mathrm{p_k}] + \mathrm{Xb_x}\Delta_{\mathrm{k-1}}\mathrm{:k} + \mathrm{f_b}) \quad \text{and}$$

$$\mathrm{X}\,\Delta\mathrm{K} - 1\mathrm{:k} = (\Delta\mathrm{K} - 1\mathrm{:k60}, (\Delta\mathrm{K} - 1\mathrm{:k180})2, (\Delta\mathrm{K} - 1\mathrm{:k365})3) \qquad (22)$$

Equations 6 and 7 indicates that the impotent key term on sentence information is stored in the unit state, and the old unit state is updated to create a temporary state. Let's assume $\mathrm{f_i}$ and $\mathrm{z_i}$ −connection weight of temporary state, i˜k −new candidate value,

$$\mathrm{i\tilde{}k} = \tanh(\mathrm{z_i}[\mathrm{o_{k-1}}, \mathrm{p_k}] + \mathrm{f_i}) \quad \text{and } \mathrm{i_k} = \mathrm{b_k} * \mathrm{i_{k-1}} + \mathrm{c_k} * \mathrm{i\tilde{}k} \qquad (23)$$

As indicated in equation 8, the final network output is evaluated utilizing the subsequent method as the input to the current hidden state. Equation 9 illustrates the compute element-wise logistic sigmoid function or a nonlinear activation function to get the hidden level-based weight matrix. Where $\mathrm{o_k}$ −current hidden state, $\mathrm{i_k}\&\mathrm{o_k}$ −input time step, $\mathrm{h_k}$ −output state, o-hidden unit, R and z-weight matrix, m −non-linear activation or sigmoid function.

$$\mathrm{o_k} = \mathrm{h_k} * \tanh(\mathrm{i_k}) \text{ and } \mathrm{o_k} = \mathrm{m}(\mathrm{Rp_k} + \mathrm{zo_{k-1}}) \qquad (24)$$

Equation 10 indicates the normality factor for the output value by analyzing the conditional probability of the input value and calculating the value proportionate to the function's product. Calculate the score function for the output sequence prediction as shown in Equations 11 and 12. Let's assume $\mathrm{w_p}$ −normalization factor, $\mathrm{V(q, p)}$ −cliques set, $\mathrm{V(Q_v, P_v)}$ −clique potential, y-sequence, $\sigma(\mathrm{P, Q})$ −score function, A-transition score matrix, l-length, X-matrix score, $\mathrm{x_c}, \mathrm{q_c}$ −score of tag data, $\mathrm{E_{q_c q_{c+1}}}$ −score of transition tag, arguments

$$\mathrm{X(Q|P)} = \frac{1}{\mathrm{w_p}} \prod_{\mathrm{v} \in \mathrm{V(q,p)}} \Phi_{\mathrm{v}}\,(\mathrm{Q_v}, \mathrm{P_v})$$

$$\mathrm{Q^*} = \mathrm{ar}_{\mathrm{q} \in \mathrm{Q}}\sigma(\mathrm{P, Q})$$

$$\sigma(\mathrm{P, Q}) = \sum_{\mathrm{c=0}}^{\mathrm{l}} \mathrm{E_{q_c q_{c+1}}} + \sum_{\mathrm{c=1}}^{\mathrm{l}} \mathrm{x_c}, \mathrm{q_c} \qquad (25)$$

In equation 13, the classification probability vector for important prediction is calculated using a sigmoid function in a single time slice, showing the output probability of diagnosis. The calculation of the label prediction probability vector involves summing the prediction losses across every period slice and applying a weight to the cumulative sampling loss, factoring in the estimate loss from the final period slice. Let's assume $\hat{q}$ −probability vector, $\hat{q}_c$ −text term frequency of probability vector, N-loss function, i-dimension class label vector, $q^{(k)}$ −label of time slice, α −hyperparameter model.

$$\sigma(P,Q)\begin{cases} N(\hat{q},q)\frac{1}{|i|}\sum_{c=1}^{c=[i]}-(q_c.\log(\hat{q}_c)+(1-q_c).\log(1-\hat{q}_c)) \\ N=\alpha\frac{1}{K}\sum_{k=1}^{K}N\left(\hat{q}^{(k)}-q^{(k)}\right)+(1-\alpha).N\left(\hat{q}^{(K)}-q^{(K)}\right) \end{cases} \tag{26}$$

The classification label of a time slice represents a predicted probability vector. Compute the total expected loss across all time slices and weight the final time slice against the loss of the complete sample. Diagnostic classification with the LSTM-RNN model can offer the probability output for diagnosis.

RoBERTa Algorithm

The RoBERTa algorithm enables the analysis of causal relationships between important key term based on sequences of word-to-word proceedings. It can be analyzed that the input sequence of the BERT algorithm is more constant and accurate than the original sequence. The parameters can be updated utilizing the same transformer structure assessed by the original BERT method. A stochastic objective function consisting of a vector of exponential decay rate parameters for moment estimation can be used to predict sentence sequence. In addition, the gradient can improve the bias's first and second raw moment estimates by estimating random target time steps. Furthermore, the BERT algorithm can optimize the bias parameters of the corrected first bias-corrected second raw moment evaluation.

Algorithm: RoBERTa

```
Input: Feature key term learning rate
Output: Update Parameter $\theta_k$
Start
Compute the initial moment vector $G_0 \leftarrow 0$
Calculate the vector's second moment $S_0 \leftarrow 0$
Begin time step $k \leftarrow 0$
For each $\theta_k$ do
                    $k \leftarrow k+1$
    Calculate the stochastic gradients over time
                    $m_k \leftarrow \nabla_0 b_k(\theta_{k-1})$
    Compute the updated bias of the first and second original moments.
```

$$G_k \leftarrow \beta_1 . G_{k-1} + (1-\beta_1). m_k$$
$$S_k \leftarrow \beta_2 . S_{k-1} + (1-\beta_2). m_k^2$$

Estimate the bias-corrected first and second raw moments.

$$\widehat{G_k} \leftarrow G_k/\left(1-\beta_k^1\right)$$
$$\widehat{S_k} \leftarrow S_k/\left(1-\beta_k^2\right)$$

Parameter update $\theta_k \leftarrow \theta_{k-1} - \alpha . \widehat{G_k}/\left(\sqrt{\widehat{S_k} + \epsilon}\right)$

End for each

Return θ_k

Stop

Let's assume β_1, β_2 −exponential decay rate, α −leaning rate, t-time step, G-moment, S-vector, b-function, M-gradient vector, $b_k(\theta_{k-1})$ −objective function, θ_0 −initialize parameter, θ −parameter.

3.5 Sequence Ranking sentence fragmentation (SRSF)

Utilizing key-based ranking to retrieve and summarize relevant information from unstructured documents employing a simple similarity measure to calculate the similarity between queries, a separate ranking model is created for each training key and its corresponding document. The sentences are ranked based on their scores, determined by tokenizing the sentence and identifying the token with the highest frequency. To eliminate the redundancy, a cosine similarity matrix is employed. The rank function prioritizes the removal of more sentences than non-extracted sentences in each training document rather than training a classifier for sentence extraction.

Algorithm: SRSF

Step 1: Collection of Indexing Documents

For each document:

Perform relevance key term on topic relevance sequence using a hashtag key.

Remove irrelevant words and tokenize the remaining words.

Step 2: Extracting Sentences from the hashtag Key terms

Extract features from the key, such as

term length, position, similarity, nouns, weight, date, and numerical data.

Step 3: Sentence Score Calculation

Calculate a score for each sentence based on the extracted features.

Step 4: Sentence Grouping Using Key Searching

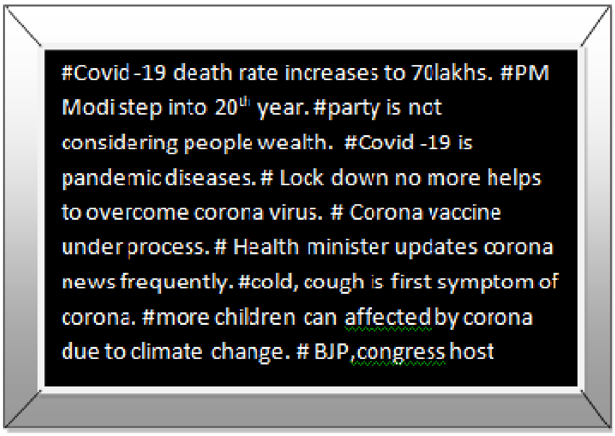
Group sentences based on their similarity to the key terms.

```
Remove similar terms to avoid redundancy.
Step 5:   Ranking
Rank the sentences based on their position within the document.
Step 6:   generate sequence relevance hashtag relevant sentence
Select sentences based on the calculation of term weights and generate scores
for each   sentence.
For each sentence feature term, u_s
Retrieve the user's queries (Qs=u_r.getKeySet()).
Segment the key set by function values.
For each key set, Q_(sr)
If the document contains Q_sr
If the score values for Q_sr in the document < dataset for Q_sr
Update the values for Q_sr in the dataset.
Else
Add Q_sr to the document.
Return the document.
Step 7:   Summarization
Generate a summary based on the selected sentences from the document.
Step 8:   Stop
```

The proposed work extracts information features and uses similarity measurement techniques to generate sentence scores. Once the sentence score is generated, the sentence is generated. Sort the set of queries and sentences in each group and include the sentences with relatively high scores in each set in the final summary. A summary of a text document is created by identifying the basic sentences in the document.

4. Results and discussion:

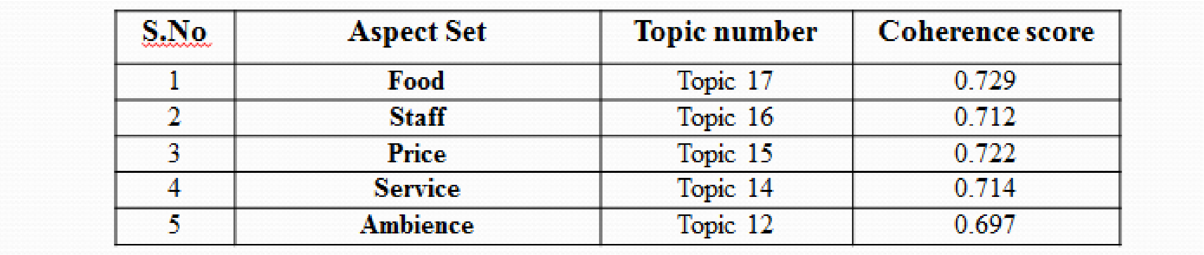
The experimental results demonstrate that the proposed system significantly enhances the precision rate and reduces sentence fragmentation in text summarization for social media content. By leveraging the PSO-optimized LSTM Gated RoBERTa Algorithm, the system achieves high performance in terms of F1 score and time complexity, showcasing its effectiveness in handling vast amounts of information efficiently. The integration of advanced AI techniques and optimization algorithms proves to be instrumental in improving the overall quality of content extraction and summarization. The dataset is collected from Twitter API using the Python interface. The collected data was stored for preprocessing and checking for the normalization process, having 472 documents from COVID-19 logs and 5678 sentences and observed key terms with 17 Topics. The accuracy and precision of the proposed method are calculated and tested with a confusion matrix in Python language. Sentences or words that are semantically matched with WordNet-based trained data are tested.



#Covid -19 death rate increases to 70lakhs. #PM Modi step into 20th year. #party is not considering people wealth. #Covid -19 is pandemic diseases. # Lock down no more helps to overcome corona virus. # Corona vaccine under process. # Health minister updates corona news frequently. #cold, cough is first symptom of corona. #more children can affected by corona due to climate change. # BJP,congress host

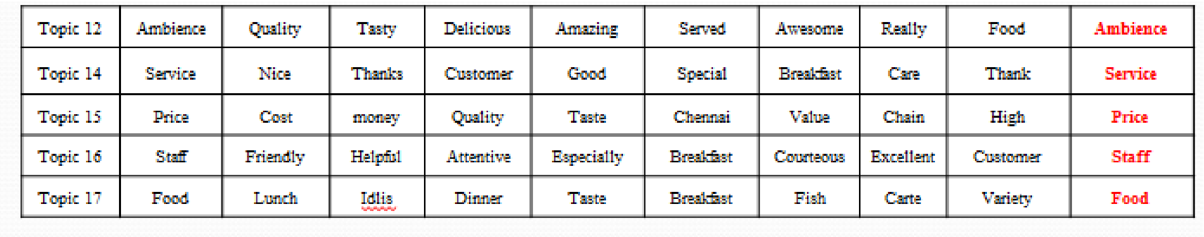
Figure 2. Twitter Covid-19 tweet data loading

Table:1 Feature representation with key terms



S.No	Aspect Set	Topic number	Coherence score
1	**Food**	Topic 17	0.729
2	**Staff**	Topic 16	0.712
3	**Price**	Topic 15	0.722
4	**Service**	Topic 14	0.714
5	**Ambience**	Topic 12	0.697

Table:2 Topic representation with key terms



Topic 12	Ambience	Quality	Tasty	Delicious	Amazing	Served	Awesome	Really	Food	**Ambience**
Topic 14	Service	Nice	Thanks	Customer	Good	Special	Breakfast	Care	Thank	**Service**
Topic 15	Price	Cost	money	Quality	Taste	Chennai	Value	Chain	High	**Price**
Topic 16	Staff	Friendly	Helpful	Attentive	Especially	Breakfast	Courteous	Excellent	Customer	**Staff**
Topic 17	Food	Lunch	Idlis	Dinner	Taste	Breakfast	Fish	Carte	Variety	**Food**

In Figure 2, The twitter data is loaded into the database. The data is preprocessed, and the similarity score is calculated using WordNet. Large numbers of data are collected on the database for similarity-based extraction.

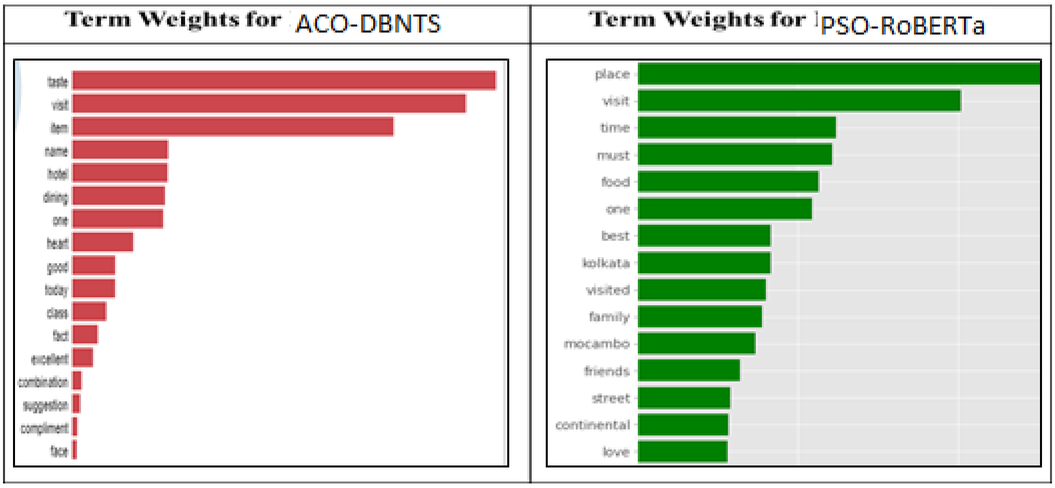


Figure 3 and 4: Covid WordNet term weight evaluation based on topic relevance

Figure 6 shows that data related to Covid-19 lexicons are stored in the trained data set. The lexicons and syntax that are relevant to COVID news are stored manually using WordNet and semantic matching. This trained WordNet is used to calculate the semantic similarity score of the system. The accuracy of data classification using similarity score is shown in the below graph. Figure 7 shows the accuracy of data classification in the proposed system improves better than existing k-means, and SVM classifier algorithms.

Table :3 Comparison of proposed Precision, Recall and F1 measure

Classifiers	Precision(P), Recall(R) and F1 Score (%)											
	Food			Service			Staff			Ambience		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Naive Bayes	0.86	0.85	0.85	0.94	0.94	0.94	0.95	0.95	0.95	0.91	0.91	0.91
Decision Tree	0.91	0.91	0.91	0.95	0.95	0.95	0.94	0.94	0.94	0.91	0.90	0.90
SVM	0.94	0.93	0.93	0.96	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95
ACO-DBNTS	0.95	0.94	0.94	0.98	0.97	0.97	0.97	0.96	0.96	0.97	0.96	0.96
LSTMg-RoBERTa	0.97	0.96	0.96	0.98	0.98	0.98	0.98	0.97	0.97	0.97	0.97	0.97

The precision is about similarity, which is how effectively matched and data is classified. The proposed system has classifications that are very accurate when every data is sentimentally classified based on lexicon and semantics.

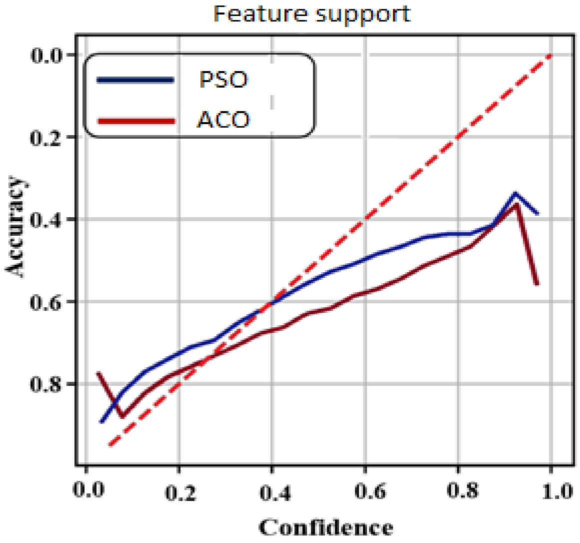


Figure 5 Confidence sentence score evaluation

The estimation of how close the features are extracted to a particular domain is measured in Figure 8. The proposed methodology computes similarity two times while optimizing relevant data to a particular domain. So that highly relevant and most accurate data can be extracted and stored using our proposed semantic similarity k-means-ACO method.

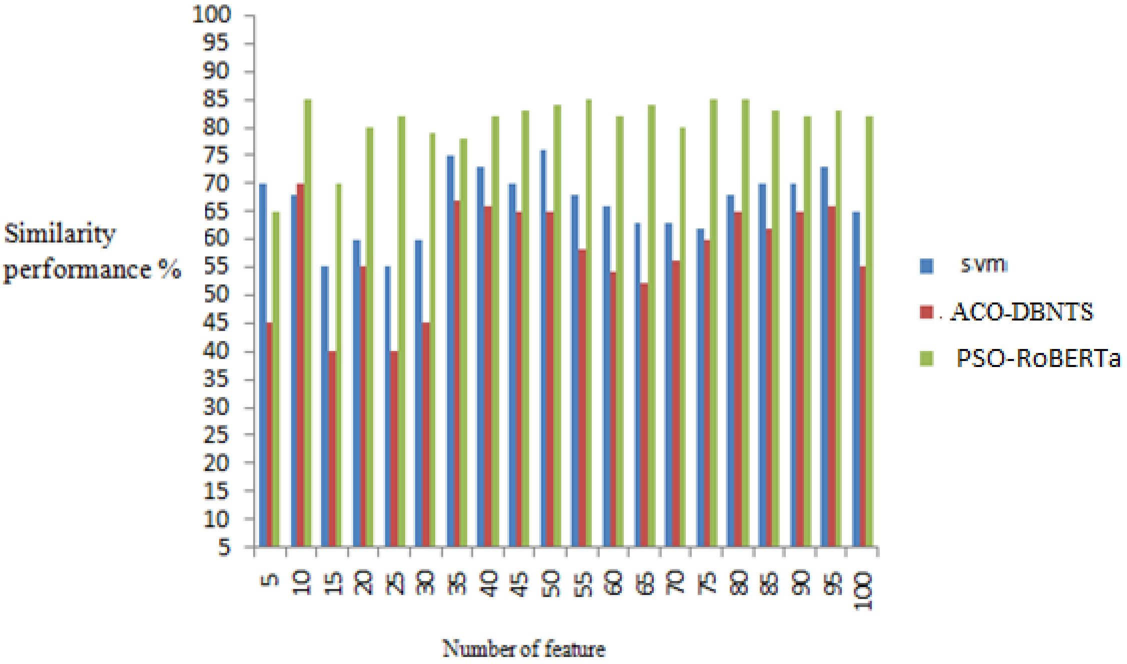


Figure 6: word similarity measure and feature extracted.

From figure 6, it's clear that semantic score helps to extract the related data more. Machine learning techniques like SVM and K-means show less similarity matching performance compared to our proposed algorithm. This classification is the first step in the sentimental analysis before the further polarity allocation process. Sometimes, the same data can be relevant to two or more domains. This problem is also handled in our problem. Semantic similarity matching is important in all fields of data classification. The limitation of the proposed method is that similarity calculation has to be improved by high-level methods. In our proposed method, a simple methodology is tested. Furthermore, semantic techniques can be studied to improve performance.

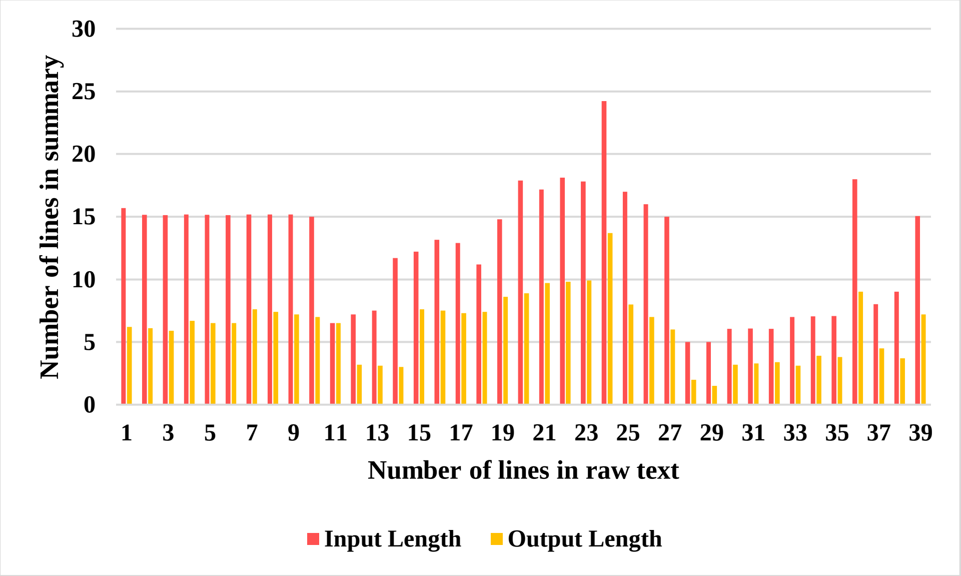


Figure 7. Total number of lines in summary Vs raw text

For example, the recommended size ratio for summaries is 33 to 40%, but some summaries may have a specified text size ratio of up to 80%. As seen in Figure 7, about 40 things are separated into subsections.

Table:4 Comparison of summarized content mean rate

Model	Sentence length mean count in %)	Summarized content length mean count in %
Decision tree	45.17	30.34
SVM	37.22	22.21
ACO-DBNTS	37.72	15.42
LSTMg-RoBERTa	43.12	13.72

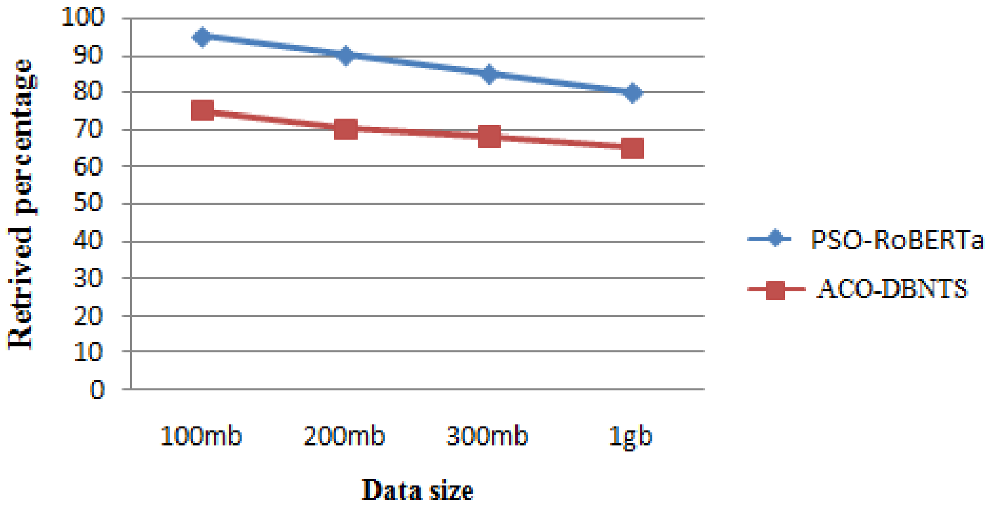


Figure 8: Recommended data using ACO-DBNTS technique vs. PSO-RoBERTa

Figure 8 represents semantic data retrieving for Twitter data recommendation. The ACO technique helps to retrieve more news than existing hashtag methods. The pheromone level is used to match the data and retrieve the more relevant data from the server.

5. Conclusion

In conclusion, the implementation of sentence-based frequency fragmentation text summarization using the PSO-optimized LSTM Gated RoBERTa algorithm presents a promising solution for handling the challenges posed by information overload on social media platforms. By prioritizing key features and optimizing the summarization process, the proposed system offers a more precise and efficient approach to extracting valuable insights from social media content. Future research could explore further enhancements and applications of this methodology in diverse domains to maximize its impact on information handling and knowledge extraction in the digital era. These results indicate that our proposed method, QS-RSDR, performs better than the existing approaches in terms of minimizing false classifications. With a lower false rate, QS-RSDR demonstrates its effectiveness in providing more accurate and reliable information retrieval compared to the other methods. By reducing the false rate, QS-RSDR enhances the precision and reliability of the document summarization process. These findings support the notion that our proposed method improves the quality of search results and contributes to a more effective and efficient information retrieval system. The number of features extracted from the dataset using optimal PSO-based classification is higher and more accurate resulting in a precision rate of 97.6 % and recall rate of 96.2%, better than other existing methods. The extracted feature is more relevant to the particular domains on topic modeling summarized with LSTMG-RoBERTa. It improves the accuracy of 97.8 % of summarized content as well compared to the other systems.

References

1. G. MalarSelvi and A. Pandian, "Analysis of Different Approaches for Automatic Text Summarization," IEEE (ICCMC), 2022, pp. 812-816, doi: 10.1109/ICCMC53470.2022.9753732.

2. N. Yadav, R. Kumar, B. Gour and A. U. Khan, "Extraction-Based Text Summarization and Sentiment Analysis of Online Reviews Using Hybrid Classification Method," IEEE (WOCN), 2019, pp. 1-6, doi: 10.1109/WOCN45266.2019.8995164.
3. G. Shidaganti, H. Dagdi, I. Jagdish and Aman, "Summarization of Student Feedback using Sentiment Analysis: Case Study," IEEE (GCAT), 2021, pp. 1-4, doi: 10.1109/GCAT52182.2021.9587751.
4. L. Xu, V. Hristidis and N. X. T. Le, "Clustering-Based Summarization of Transactional Chatbot Logs," IEEE (HCC), 2019, pp. 60-67, doi: 10.1109/HCC46620.2019.00017.
5. D. Marques, A. V. de Carvalho, R. Rodrigues and E. Carneiro, "Spatiotemporal Phenomena Summarization through Static Visual Narratives," IEEE (IV), 2020, pp. 467-472, doi: 10.1109/IV51561.2020.00081
6. K. S. Umadevi, R. Chopra, N. Singh, L. Aruru and R. J. Kannan, "Text Summarization of Spanish Documents," IEEE (ICACCI), Bangalore, India, 2018, pp. 1793-1797, doi: 10.1109/ICACCI.2018.8554839.
7. B. N, D. Kumari, B. N, M. N, S. K. P and S. R. A, "Text Summarization using NLP Technique," IEEE , 2022, pp. 30-35, doi: 10.1109/DISCOVER55800.2022.9974823.
8. M. Steinert, R. Granada, J. P. Aires and F. Meneguzzi, "Automating News Summarization with Sentence Vectors Offset," IEEE (BRACIS), 2019, pp. 102-107, doi: 10.1109/BRACIS.2019.00027.
9. Narisha Zaho, Combination of Convolutional Neural Network and Gated Recurrent Unit for Aspect-Based Sentiment Analysis, IEEE (J&M), Volume 9, PP 15561-15569, 2021.
10. T. Wang, K. Lu, K. P. Chow and Q. Zhu, "COVID-19 Sensing: Negative Sentiment Analysis on Social Media in China via BERT Model," in IEEE Access, Volume. 8, pp. 138162-138169, 2020.
11. F. A. Ghanem, M. C. Padma and R. Alkhatib, "Elevating the Precision of Summarization for Short Text in Social Media using Preprocessing Techniques," 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, 2023, pp. 408-416, doi: 10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00063.
12. P. Gupta, S. Nigam and R. Singh, "A Ranking based Language Model for Automatic Extractive Text Summarization," 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), Hyderabad, India, 2022, pp. 1-5, doi: 10.1109/ICAITPR51569.2022.9844187.
13. Das, Ajit & Chitkara, Kushagra & Sarkar, Apurba. (2022). Time Series Analysis on Covid 19 Summarized Twitter Data Using Modified TextRank. 10.1007/978-981-19-3089-8_2.
14. C. S. Lakshmi, S. Saxena and B. S. Kumar, "Text Mining Data Based Summarization of Novel Approach for Covid-19 with the Use of Machine Learning Techniques," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1349-1354, doi: 10.1109/IC2PCT60090.2024.10486675.
15. S. Vatsa, S. Mathur, M. Garg and R. Jindal, "COVID-19 Tweet Analysis using Hybrid Keyword Extraction Approach," 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2021, pp. 136-140, doi: 10.1109/CSNT51715.2021.9509636.
16. Das, Ajit & Thumu, Bhaavanaa & Sarkar, Apurba & Shanmuganathan, Vimal & Das, Asit. (2022). Graph-Based Text Summarization and Its Application on COVID-19

Twitter Data. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 30. 513-540. 10.1142/S0218488522400190.

17. P. Omrani, Z. Ebrahimian, R. Toosi and M. A. Akhaee, "Bilingual COVID-19 Fake News Detection Based on LDA Topic Modeling and BERT Transformer," 2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA), Qom, Iran, Islamic Republic of, 2023, pp. 01-06, doi: 10.1109/IPRIA59240.2023.10147179.
18. A. Balaram and S.Vasundra,"Sampling-based Software Prone Technique for an Optimal Prediction of Software Faults",Indian Journal of Computer Science and Engineering (IJCSE), Vol. 13 No. 4 Jul-Aug 2022.
19. Gudakahriz, S.J., Moghadam, A.M.E. and Mahmoudi, F., 2023. Opinion texts summarization based on texts concepts with multi-objective pruning approach. The Journal of Supercomputing, 79(5), pp.5013-5036.
20. Verma, P., Verma, A. and Pal, S., 2022. An approach for extractive text summarization using fuzzy evolutionary and clustering algorithms. Applied Soft Computing, 120, p.108670.
21. Gambhir, M. and Gupta, V., 2022. Deep learning-based extractive text summarization with word-level attention mechanism. Multimedia Tools and Applications, 81(15), pp.20829-20852.
22. A.Balaram, and S.Vasundra,"A Hybrid Soft Computing Technique for Software Fault Prediction based on Optimal Feature Extraction and Classification", IJCSNS International Journal of Computer Science and Network Security, VOL.22 No.5, May 2022.
23. Joshi, A., Fidalgo, E., Alegre, E. and Fernández-Robles, L., 2019. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. Expert Systems with Applications, 129, pp.200-215.
24. J. Cheng, F. Zhang, and X. Guo., A syntax-augmented and headline-aware neural text summarization method. IEEE Access, 8, pp.218360-218371. 2020.
25. Yadav, A.K., Singh, A., Dhiman, M., Vineet, Kaundal, R., Verma, A. and Yadav, D., 2022. Extractive text summarization using deep learning approach. International Journal of Information Technology, 14(5), pp.2407-2415.
26. S.V. Moravvej, A. Mirzaei, and M. Safayani., Biomedical text summarization using conditional generative adversarial network (CGAN). arXiv preprint arXiv:2110.11870. 2021.
27. Rani, R. and Lobiyal, D.K., 2021. An extractive text summarization approach using tagged-LDA based topic modeling. Multimedia tools and applications, 80, pp.3275-3305.
28. Vinitha, M., Vasundra, S. (2023). Review on Recent Advances in Text Summarization Techniques. In: Kumar, A., Ghinea, G., Merugu, S. (eds) Proceedings of the 2nd International Conference on Cognitive and Intelligent Computing. ICCIC 2022. Cognitive Science and Technology. Springer, Singapore. https://doi.org/10.1007/978-981-99-2742-5_70.
29. Vinitha M,Vasundra S.(2024). An Efficient Ant Colony Optimization Optimized Deep Belief Network Based Text Summarization Using Diverse Beam Search Computation for Social Media Content Extraction International Journal of Intelligent Engineering and Systems, Vol.17, No.6, 2024 DOI: 10.22266/ijies2024.1231.89