IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 36, No. 5, October, 2022, pp. 777-784
Journal homepage: http://iieta.org/journals/ria

Improving Extractive Text Summarization Performance Using Enhanced Feature Based RBM Method

Grishma Sharma*, Deepak Sharma

Department of Computer Engineering, Somaiya Vidyavihar University, Vidyavihar, Mumbai 400077, India

Corresponding Author Email: grishijsharma@gmail.com

https://doi.org/10.18280/ria.360516

Received: 30 July 2022
Accepted: 19 October 2022

Keywords:
extractive text summarization, neural networks, restricted boltzmann machine, unsupervised learning, deep learning, feature extraction & ROUGE score

ABSTRACT

Text summarization is the process of creating a short, accurate and fluent summary of a longer text document. As plenty of digital data is available online, automatic text summarization methods greatly needed to help and understand the lengthy & complex documents quickly by discovering the relevant information. This paper proposes the text summarization method for short news articles and long scientific papers using unsupervised neural network model. The proposed method works in four main steps: input document pre-processing, feature extraction, feature enhancement and final summary generation. We have extracted combination of various statistical and linguistic features from input document, which helps in improving the quality of sentence selection. Further Restricted Boltzmann Machine (RBM) model is used to capture & enhance the discriminative, abstract features in an unsupervised way to improve the overall performance without losing any significant information. Sentences are scored based on enhanced feature set and top sentences are selected for final extractive summary. Performance of the proposed method is evaluated using Rouge score and compared with TextRank, LexRank, LSA & Luhn baseline methods and the results demonstrates that proposed methodology performs better compared to other methods.

1. INTRODUCTION

The availability of internet-enabled technologies is bringing information from all sources around the world and generating data at an enormous rate. This rapid volume of data available on the internet has piqued researchers' interest in developing techniques for condensing it into a useful summary [1]. Because of the widespread availability of internet-based information including the digitalization of books, online news articles, scientific papers, and blogs, extracting important information from massive datasets has become infeasible. Manual data analysis takes so much time and effort, there is a growing demand for automatic summary generation from text documents to classify, understand, and present data in a concise manner [2]. The text summarization techniques can be classified in numerous ways like based on input document, single or multi, based on output summary, i.e., abstractive or extractive and query based or generic summarization.

Based on the summary generated, text summarization can be divided into two categories: extractive and abstractive summarization. In extractive summarization, important sentences are selected from the original document only, while the abstractive summary contains new words and phrases apart from the original content, these summaries are more like human summary. These techniques are more challenging compared to extractive summarization [3]. In this research work, we follow the extractive methodology to implement & develop a text summarization technique for short as well as long document like news articles, descriptions, factual reports, and long scientific articles. We have proposed and developed an unsupervised approach for single document summarization using RBM neural network architecture.

Restricted Boltzmann Machines (RBMs), proposed by Geoffrey Hinton in 2007 [4], are generative neural network models for unsupervised learning, which can learn a probability distribution over its set of inputs. In recent years, RBM are used in various applications like dimensionality reduction, feature learning, classification, topic modeling and collaborative filtering, depending on task they can be trained as supervised or unsupervised ways. RBM model consists of two layers of binary units: the first layer is the visible layer, to represent the input data and the second is hidden layer to increase the learning capacity. Most of the times, input data has hidden information which is not captured by feature extraction. The RBM model maps these simple or low-level features into a complex feature representation, i.e., RBM model the variation among the correlated variables of an input document.

This paper aims to propose an approach by referencing the architecture of RBM neural network model. The proposed approach has four main important phases: Preprocessing of input text, Feature extraction, Feature enhancements using RBM model and Final Summary Generation. We have implemented the RBM model from scratch on BBC News dataset and Scientific articles. Eleven important features are extracted from all input documents, features are enhanced using RBM model, based on the enhanced feature score, the important sentences are selected, and final summary is generated. The performance of proposed algorithm is compared with baseline unsupervised models TexRank [5], LexRank [6], LSA [7] & Luhn [8] based on Precision, Recall

and F1 score. The entire paper is covered in five sections: Section 1 Introduction, Section 2 presents the detailed literature and the use of various unsupervised techniques in summarizing the documents, Section 3 explains the proposed methodology step-by-step, section 4 is used to report the study's findings and results, and section.5 concludes the study with a discussion and future research works.

2. LITERATURE REVIEW

The Extractive text summarization system generate a summary that has a few important sentences selected from the original input documents. The research in this field has begun from the early sixties and until now it is going on. There is a great improvement in the research of text summarization in terms of representation of input document, different feature extraction techniques, sentence selection strategies, and various datasets. Extractive summarization is divided into five main subcategories by Gambhir and Gupta [3]. They are statistical, graph-based, topic-based, machine-learning-based, and discourse-based approaches. Recent research in text summarization in extractive as well as abstractive has inclined towards various Neural Network models and Deep Learning models [9].

The Extractive summarization method extract the sentences from the source document based on different keywords and features. Earlier techniques of automatic text summarization were based on statistical approaches. Based on surface-level features such as words and sentences, which determines which parts of a text are important and relevant. Luhn [8] explains the phenomenon of frequent words playing a role in the categorization of relevant and nonrelevant words which ultimately decides the distance between relevant words, in one of the most often used extractive automatic text summarization algorithms. Term Frequency-Inverse Document Frequency (TF-IDF) is another type of statistical-based approach, and it works on the frequency of words in the given documents. The term frequency score is calculated by adding the frequency of terms appearing in a sentence [10].

In semantic-based approach Latent Semantic Analysis (LSA) it works on semantic principles to analyze the relationships between different words and documents, LSA learns latent features and topics by executing singular value decomposition (SVD) on a term-document matrix [7]. Another technique in text summarization is the graph-based technique. These techniques are being extensively used in text summarization since document structures are efficiently represented by graphs. The TextRank [5] and LexRank [6] are the modified version of the PageRank algorithm to score sentences. Both are unsupervised, iterative ranking algorithms calculate the sentence correlation and determines the total relevance of sentences.

Researchers consider the text summarization task not only from statistical graph or NLP point of view but also as a machine learning task. Most machine learning methods consider the summarization problem as a binary classification task [11]. The important sentences chosen for summary generation are labelled as positive in this process, while the remaining sentences which are not important are labelled as negative (or neutral). Then, on the tagged data, ML algorithms such as logistic regression, neural networks and support vector machines, etc. are used to predict the likelihood of selecting the remaining sentences for a summary generation. Hidden Markov model (HMM) [12], conditional random fields (CRF) [13] are other important and significant approaches in machine learning to improving the performance of extractive text summarization. In contrast, the most recent techniques in text summarization are neural network and deep learning models.

Traditional machine learning approaches like TF-IDF, text rank, and LSA are used in conjunction with manual feature engineering methods to produce better results in text summarization tasks. Since 2016, several authors have attempted to generate summaries using neural network-based algorithms. Performance of the complex task like machine translation, summarization, and question answering is also improved to a great extent. In this technique, words and sentences are often represented in vector form, the neural network learns to capture the semantic meaning between the words using the vector calculus operations. Various neural network models are used for text summarizatioin task like Multi-layer perceptron (MLP), Restricted Boltzmann machines [14], Convolution Neural network (CNN) [15] and Recurrent Neural network (RNN) [9]. In the study [16], author have extracted the statistical features and feature enhancement is done by RBM method to improve a summarization model's efficiency. The authors [17] have combined RBM with fuzzy logic for improving the accuracy of text summarization. Sentence-Centroid similarity and thematic words were used to improve the connectivity of the sentences, contributing to the model's high accuracy.

RNN, Long short term memory (LSTM) Sequence to sequence encoder decoder architectures are also used in extractive as well abstractive text summarization tasks [9]. These architectures process the input text sequentially, so cannot handle the long sequences very well and takes large time for training. To overcome this transformer neural network architecture [18] was introduced, they have an encoder-decoder architecture similar to RNN, but the difference is that the input sequence can be passed in parallel instead of sequentially. Extension to this language model, google has introduced state of art model BERT- Bidirectional Encoder Representation for Transformers [19]. BERT combines both word and sentence representations in a single very large transformer. It is pretrained on vast amounts of text, with an unsupervised objective of masked language modeling and next-sentence prediction and can be fine-tuned with various task-specific objectives. These language models have shown the great improvement in the high-end complex NLP tasks like Abstractive Text summarization [20], text classification [21], and many more. No doubt performance is up to the mark but to fine tune for any specific task, it needs lot of resources like high end GPU's.

To summarize the above literature survey about text summarization, survey of literature revealed that the accuracy of extractive text summarization can be improved using enhanced statistical and linguistic feature selection and pre-processing of the data in a unsupervised way. In this study, we aim to deploy RBM for enhancing the feature selection for text summarization. The purpose of using RBM is due to the factors of computational efficiency and expressive enough to encode any distribution.

3. PROPOSED METHODOLOGY

Extractive summarization usually suffers from three challenges, first ranking the words or sentences, the second

challenge is selecting the subset of sentences from ranked sentences and third is ensuring the coherence between the generated summaries to avoid disconnections between the generated text or summary. Our approach overcomes the ranking of sentences issue by selecting the eleven features which are used to score and re-order sentences from the paragraphs. The second issue of selecting the sentences can be over by considering the top n sentences with high sentence scores, and sentence scores with respect to paragraph. The third challenge of achieving coherence and the meaning of the generated text can be measured using the ROUGE score and manually verification of some of the generated summaries with source documents. Figure 1 shows the various steps of the methodology used for extractive text summarization using the RBM method. Initially, the text is preprocessed; important features are extracted from preprocessed text. Features are then enhanced using the RBM method. Finally, the sentences are scored and important sentences are extracted.

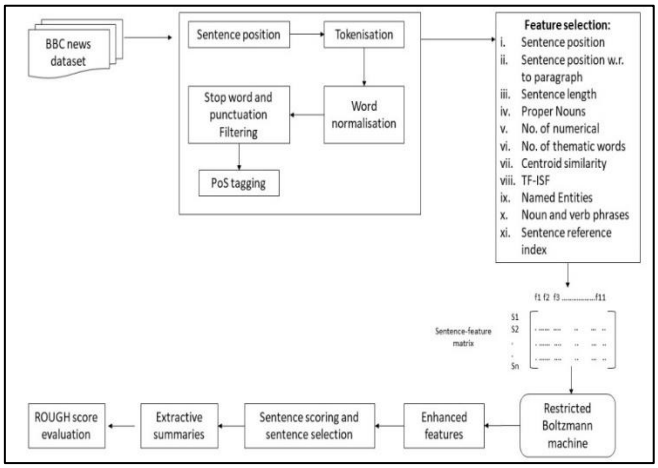

Figure 1. Proposed methodology

3.1 About the dataset

We have used the BBC news summary dataset [22] consisting of articles from five different domains, there are 2225 documents in total across the business, entertainment, politics, sport and technical domains published between 2004 to 2005.

3.2 Data pre-processing

This includes a set of operations to remove noise, imbalance, special characters and stop words from the text. This improves the model's accuracy by eliminating unnecessary information from the data. Data preprocessing is very important, properly labelled and tuned data has been shown to improve ML models' accuracy. The real world includes special characters, stop words, a lot of inconsistent and irrelevant information, missing values and class imbalance issues. Pre-processing eliminates irrelevant and unwanted words, phrases, and characters, it helps to improve the quality of data before applying ML or data mining algorithms. The following set of pre-processing operations is performed on the data.

3.2.1 Sentence splitting and tokenization

This process splits long text into a short text. The text which in paragraphs is split into no. of sentences and again sentences are divided into words until no further division is possible. The words are usually represented as tokens, and each token is assigned a separate array or vector value. Usually, a sentence starts with a word with a capital letter and ends with a full stop (.). The NLTK library available in Python programming language is used to split the text into multiple sentences.

3.2.2 Part of speech tagging

It is a process of assigning each word a Parts of Speech (PoS) tag-based on the existing dictionary and rules. Sometimes assigning the PoS tag to a word becomes difficult due to the ambiguity of English words. There are many algorithms to assign a PoS tag to the tokens, we have used the Stanford PoS tagger for assigning the PoS tagging work.

3.2.3 Stop word and punctuation Filtering

Stop words are meaningless and frequent words in the document that do not contain any important information related to the given topic. The presence of stop words might increase processing time and they are often causing a reduction in the model's accuracy. The NLTK module has more than 40 common stop words used in English, and we can add or remove specific stop words using append, extend, and remove functions. After removing the stop words all punctuation marks such as [,._/:;<>+*] are removed to improve the useful content in the text.

3.2.4 Word normalization

In the normalisation process, the inflectional form of a word is removed during the normalisation process so that the base form can be obtained.

3.3 Feature extraction

After preprocessing, the next step is to extract features from the newly generated text. This technique transforms textual data into a numerical format so that ML algorithms can process the data easily. Feature engineering computes the weight of each feature and selects the top features which are having close association with the target variable. These features are stored as vectors and a sentence feature matrix is generated. Various combinations of features are tried, and the 11 most suitable features are selected for the input documents. We discuss various feature extraction techniques in the following paragraphs.

3.3.1 Sentence position

The importance of a sentence for the summary can be determined by its placement. Some studies have mentioned that the first and last sentences of a document are usually the most important [23]. These sentences usually carry important information about what, why and when something has happened. So, based on the mathematical equation (1), the sentence score is calculated.

$$\text{Sentence\_Position} = \begin{cases} 1, if\ it\ is\ first\ or\ last\ sentence\ of\ a\ text. \\ else\ \cos\left(\left(sen\_pos - min\right)\left(\left(\frac{1}{max}\right) - min\right)\right) \end{cases} \quad (1)$$

3.3.2 Sentence position w.r.to paragraph

Baxendal [24] study has looked at 200 paragraphs and discovered that the theme sentence was the first sentence in 85 percent of the cases and the last sentence in 7% of the cases. The first and end sentences in the paragraph convey important

information, so this feature is described as:

$$\text{Sen\_Pos\_in\_para} = \begin{cases} 1, if\ it\ is\ a first\ and\ last\ paragraph\ in\ sentence \\ 0, otherwise \end{cases} \quad (2)$$

3.3.3 Sentence_length

The sentence length score helps to sort out sentences which are too small that don't provide much information. We have considered the following equation to rule out short sentences

$$\text{Sentence\_length} = \begin{cases} 0, if\ number\ of\ words\ is\ less\ than\ 3 \\ else\ number\ of\ words\ in\ the\ sentence \end{cases} \quad (3)$$

3.3.4 Proper noun score

A proper noun is a name or a location that has a unique identity. Nouns carry a lot of information about the event who, what and when something has been done. Initially text is preprocessed and NLTK library POS tagger is used for tagging the text and from tagged text, score of proper noun is calculated.

$$\text{Proper noun score} = \frac{Number\ of\ proper\ noun\ in\ tagged\ Si}{Total\ number\ of\ words\ in\ tagged\ Si} \quad (4)$$

where, S_i refers to ith sentence in the given text.

3.3.5 Number of numerals

Since the numbers in a text represent facts, having phrases with exact numeric values is crucial. The following formula can be used to calculate the number of numerical.

$$\text{Number of numerals} = \frac{\textbf{no.of numerical in sentence}_i}{\textbf{total no.of words in sentence}_i} \quad (5)$$

3.3.6 Number of thematic words

The top ten most frequently used words in a sentence are called thematic terms. Thematic words reveal the context of the given text such as playing cricket, accident on the highway, stock price increase etc. We have followed Kupiec et al. [25] proposed method of identifying thematic words which explains that top n frequent in a document can be considered as thematic words.

$$\text{Thematic-words} = \frac{\textbf{Number of thematic words in } \boldsymbol{S_i}}{\textbf{Total number of words}} \quad (6)$$

3.3.7 Centroid - Similarity

The sentence with a maximum term – frequency and inverse-sentence-frequency is known as a centroid sentence. One sentence from the entire document is used as the centroid in the sentence to centroid feature and the cosine similarity of each sentence is computed with that sentence.

$$\text{Centroid - Similarity} = \text{cosine\_similarity}(\text{centroid}_i, \text{sentence}_i) \quad (7)$$

where, $1<=i<=N$, N is the total no. of sentences in the document, centroid_feature$_i$ is the centroid feature of ith sentence and sentence$_i$ is ith sentence of the document.

3.3.8 TF-ISF

Term frequency - Inverse Sentence frequency, or TF-ISF, is a term frequency-inverse sentence frequency that works similarly to TF-IDF term frequency-inverse document frequency. Each word's frequency in a sentence is multiplied by the total number of times that word appears in all other sentences. The product is computed and totalled across all words.

$$\text{TF\_ISF} = \frac{\log\left(\sum_{all\ words} TF*ISF\right)}{Total\ Words} \quad (8)$$

3.3.9 Named entities

For named entities, in each sentence, we count the number of the mentioned and identified entities. Sentences containing references to individuals, such as a company or a group of individuals, are generally important to comprehend an actual report in a certain way. The SpaCy library available in Python programming language is used automatically identify the entities from the given text. This approach was inspired by the study conducted by Nobata et al. [26].

3.3.10 Noun and verb phrases

In text summarization, sentences containing a greater number of verb and noun phrases are considered as important sentences and should be included in the summary [27]. Noun and verb chunking can be used to extract the important noun and verb phrases. Here we have used the Stanford POS tagger for identifying the important verb and noun phrases.

3.3.11 Sentence reference index

A sentence that comes before a sentence with a pronominal reference is given more weight in this index [28]. If a sentence contains a pronoun, the preceding sentence's weight is increased using a list of pronouns. All of the above features are extracted for all of the sentences in the document, and a sentence to feature matrix is created.

3.4 Feature enhancement using Restricted Boltzmann Machine

As mentioned in the above paragraph, 11 features were selected to prioritize the sentences. A '11'x 'n' feature-sentence which is then fed to RBM for generating complex selection criteria based on the simple feature matrix. Our approach to extractive summarization is by enhancing the feature set using RBM, which is capable of capturing, and learning internal representations from a set of inputs using a probability distribution. The Figure 2 shows the generic RBM neural network model. RBM is a two-layered artificial neural network; with the visible layer (input nodes) being the first layer and the hidden layer is the second (hidden nodes). The bias node is connected to every hidden node. In the visible layer, the input nodes are unrelated to one another. Furthermore, hidden nodes in the hidden layer are unrelated to one another. Because of the limited connections, the network is known as the Restricted-Boltzmann-Machine. Training of RBM is performed using stochastic gradient descent method. During the learning process, we sample using the Persistent Contrastive Divergence (PCD) technique [29]. We utilized the PCD approach to train the RBM for 5 epochs with a batch size of 4 and 4 parallel Gibbs Chains. Each sentence's feature vector is sent via the hidden layer, where acquired weights are multiplied by feature vector values for each sentence, and a bias value is added to all feature vector values, which is also learned by the RBM. Finally, we have a modified and improved matrix. It is important to note that the RBM will need to be retrained for each new document to be summarized.

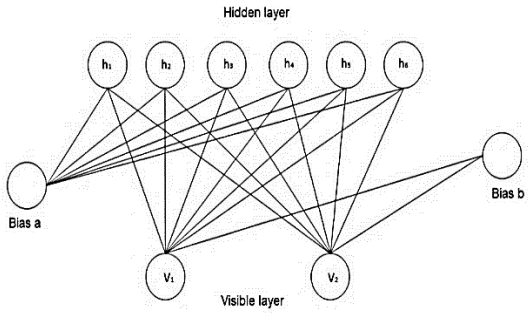

Figure 2. Graphical representations of RBM

To improve and enhance the extracted features of the input text document, this sentence feature matrix is passed through RBM, where these feature values are multiplied by randomly produced weights, and one bias value which is also randomly produced and added for all the sentences, then the sigmoid activation function is applied to produce the output.

$$P(s_i) = \sigma\left(\sum_{j=1}^{n} s_j \times W_{ij} + b_i\right)$$
$$\sigma(x) = \frac{1}{(1+e^{-x})} \quad (9)$$

where, $\sigma(x)$ the sigmoid activation function, w_{ij} is randomly generated weights and b_i is the bias. Following that, the RBM model continues to reconstruct data on its own in an unsupervised way. This is achieved by reversing the preceding procedure, in which the hidden layer becomes the input layer, with activations serving as the new input. The prior weights linked with the visible layer nodes are then multiplied by these activations, and the results are added to the visible layer bias at each visible node. As a result, the generated outputs are called reconstructions, and they are matched with the original input.

The likelihood probability of activation of a visible unit s_j is given as:

$$P(s_j) = \sigma\left(\sum_{j=1}^{n} s_j \times W_{ij} + b_i\right)$$
$$\sigma(x) = \frac{1}{(1+e^{-x})} \quad (10)$$

3.5 Summary generation

The text summary generation steps are given in Figure 3. All the sentence features are enhanced using the RBM model, and then the sentence score for all the sentences is calculated by adding all the feature scores. All sentences are arranged in decreasing order of their feature score. The first top sentence is always added in the final summary as it is highly informative. Then rest top 50% of the remaining sentences are included with the first sentence to form the final extractive summary.

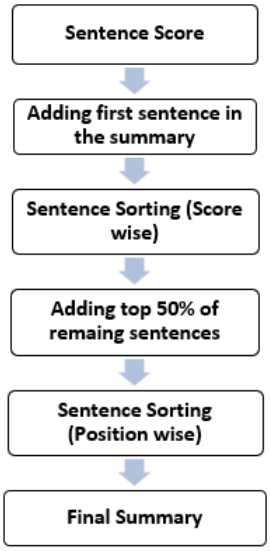

Figure 3. Summary generation steps

4. RESULTS AND DISCUSSION

The experimentation is done in Python. BBC news dataset & long scientific documents are used for experimentation, initially, each document is pre-processed and different linguistic and Statistical features are found. Features are enhanced using the RBM model and then each sentence is cored based on the features and the top sentences are selected as the final summary.

4.1 Performance evaluation

To evaluate the performance of the model ROUGE score [30] parameters are used. The 'precision is the quantity of right information recovered by a system compared to what it has recovered', i.e., it is the ratio given in the equation below:

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (11)$$

Precision is computed as the number of overlapping n-grams in both the model output summary and reference summary by the total number of n-grams in the model summary. Recall is the quantity of right information recovered by a system compared to what it should recover, given in the equation below:

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (12)$$

Recall is computed as the number of overlapping n-grams in both the model output summary and reference summary by the total number of n-grams in the reference summary. f-score is a trade-off between precision & recall, given below:

$$F = \frac{2 * P * R}{P + R} \quad (13)$$

BBC news dataset and long scientific articles are used for experimentation. The BBC news dataset contains various documents from various diverse domains such as politics, technology, sports, business and entertainment along with

human-generated summaries for each document. The performance of the RBM proposed methodology is compared with the performance of other state of art extractive summarization algorithms such as TextRank [5], LexRank [6], LSA [7] and Luhn [8]. Each document is summarized with these four different summarization techniques. At ten percent, twenty percent, thirty percent, forty percent, and fifty percent of the document's length, we extracted the top-ranked sentence to build a summary. The goal of testing with different percentage levels is to see how different techniques perform. We compared and contrasted each system summary with its associated reference summary. The ROUGE score is used for the evaluation of generated summary. ROUGE score includes precision, recall and f-measure. The results show that the proposed method outperforms all of the previous approaches.

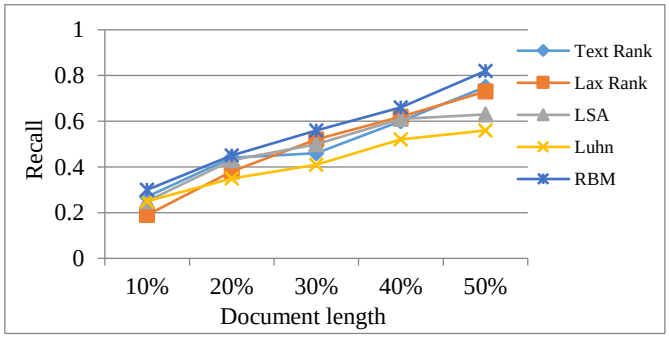

Figure 4. Recall values corresponding to document length of various documents

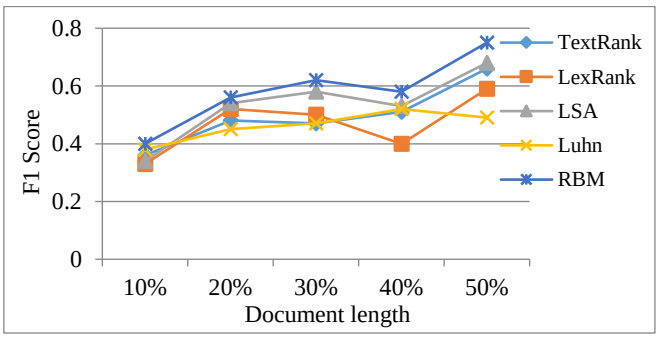

Figure 5. F1-score values corresponding to document length of various documents

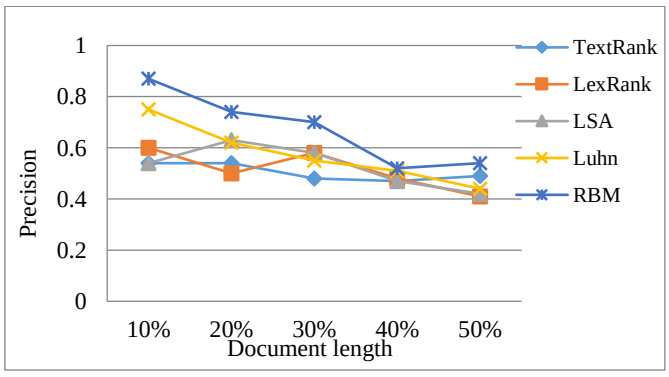

Figure 6. Precision values corresponding to document length of various documents

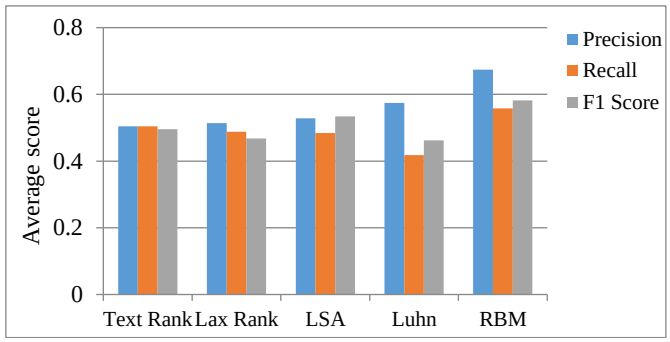

Figure 7. Comparison of performance of various methods on the given dataset

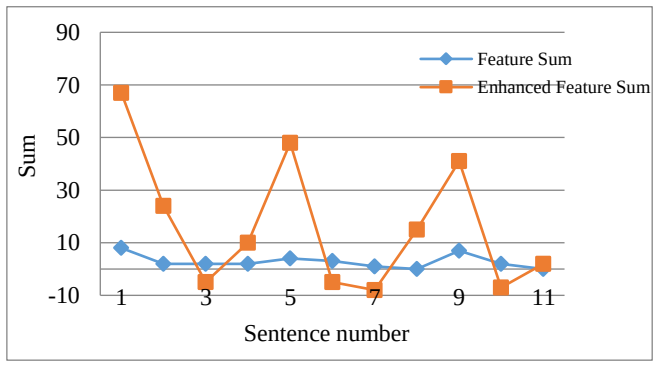

Figure 8. Feature sum versus enhanced sum for each sentence

The Recall score of several documents is shown in Figure 4. The recall is calculated by dividing the number of common words between the model generated summary by the size of the reference summary. It indicates how much of the reference summary has been covered by the model generated summary. Luhn's model has the lowest recall value, resulting in poor performance.

The F1 score of several documents is shown in Figure 5. At 10%, 20%, 30%, and 40% of document length, we receive low F-Measure scores for the Tex-rank, Lex Rank, LSA, and Luhn, indicating that the model-generated summaries are not very useful. When comparing the other stated method to the RBM model in terms of F-Measure, the other model falls short of the RBM model. As can be observed from the graph above, our current strategy produces a positive F1-Score when compared to the previous approaches.

The precision score of various documents is shown in Figure 6. Precision is calculated by dividing the number of common terms in the model generated summary by the size of the model generated summary. As a result, precision offers us a notion of the model-generated summary's relevancy and conciseness. We acquire great precision when we extract the top-ranked sentence from the document to produce a summary at 10% of the document length.

The Figure 7 shows the comparison of the performance of various approaches on the same BBC dataset. Our proposed approach which is on the right-hand side in Figure 8 as an average F1 score value of 0.58 which is higher than the Tax rank, Lax Rank, LSA, and Luhn's approaches. Hence, the proposed methodology gives a better F1 Score than all other techniques stated above. The proposed approach has an average precision value of 0.70 which are higher than the Tax rank, Lax Rank, LSA, and Luhn's approaches. Hence, the proposed methodology gives better precision than all other

techniques stated above. The proposed approach has also achieved an average recall value of 0.56 which are higher than the Tax rank, Lax Rank, LSA, and Luhn's approaches. Based on the above three metrics, we can conclude that the proposed methodology gives better recall than all other techniques stated above.

The Figure 8 shows the values of feature sum and enhanced feature sum for each sentence of one such document. The Restricted Boltzmann Machine has extracted a hierarchical representation out of data that initially did not have much variation, hence discovering the latent factors.

4.2 Scientific paper summary analysis

Instead of verifying the accuracy of our proposed approach on the BBC dataset, we have also measure the effectiveness of proposed method on long documents. To perform the analysis, we have taken a total of ten scientific research articles from the computer science domain of different length. Just like the abstract in a research paper, the purpose of summarizing the research paper is to give the audience a brief overview of what that study says. The summary is generated for each heading such as introduction, literature survey, method, etc.

The Table 1 shows the average values of Precision, recall and f1 score of all methods on the ten scientific papers. Ten different scientific papers are taken from different repositories of varying lengths. For gold summary, we have manually marked the important sentences to measure the rouge score be. As scientific documents have a specific structure, for the summary generation we have not considered the abstract and conclusion section. As we can see, performance of proposed method is best for long document also. The proposed approach has an F1 score value of 0.72 average precision value of 0.68, and a recall value of 0.78, which are higher than the Tax rank, Lax Rank, LSA, and Luhn's approach. Hence, the proposed methodology gives better recall than all other techniques stated above.

Table 1. Precision, recall & F1 score values

Methods	Average Score		
	Precision	Recall	F1-score
TextRank	0.57	0.62	0.59
LexRank	0.52	0.63	0.56
LSA	0.47	0.6	0.52
Luhn	0.51	0.63	0.56
RBM	0.68	0.78	0.72

5. CONCLUSIONS

The performance of an unsupervised extractive text summarization methodology based on RBM neural networks is evaluated using the BBC dataset and a few examples of scientific papers to ensure that it is practical. Because each document is unique in its own way, the algorithm works separately for each of the input documents. RBM has the advantage of being computationally efficient, so it can be used for single or multiple documents. Each document has different semantically driven features extracted, and a feature-sentence matrix was created as an input to RBM. RMB then subjected the input feature matrix to complex iterations to develop optimal ranking criteria. Both small and long document datasets were used to test the proposed methodology. In both cases, the proposed methodology produced an effective summary is produced. We compared our model to other existing techniques using the ROUGE1 score, and the results show that our model produces better results than the other techniques. The proposed methodology could be used to summarise multiple documents from different domains. Future studies can adopt to extract domain-specific features and adjust the hyperparameters of RBM for achieving high accuracy.

REFERENCES

[1] Sinha, A., Yadav, A., Gahlot, A. (2018). Extractive text summarization using neural networks. arXiv preprint arXiv:1802.10137. https://arxiv.org/abs/1802.10137

[2] Gupta, V., Lehal, G.S. (2010). A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence, 2(3): 258-268.

[3] Gambhir, M., Gupta, V. (2017). Recent automatic text summarization techniques: A survey. Artificial Intelligence Review, 47(1): 1-66. https://doi.org/10.1007/s10462-016-9475-9

[4] Hinton, G.E., Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786): 504-507. https://doi.org/10.1126/science.1127647

[5] Mihalcea, R., Tarau, P. (2004). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404-411.

[6] Erkan, G., Radev, D.R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22: 457-479. https://doi.org/10.1613/jair.1523

[7] Steinberger, J., Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. Proc ISIM 4:93-100.

[8] Luhn, H.P. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2): 159-165. https://doi.org/10.1147/rd.22.0159

[9] Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K. (2021). Neural abstractive text summarization with sequence-to-sequence models. ACM Transactions on Data Science, 2(1): 1-37. https://doi.org/10.1145/3419106

[10] Moratanch, N., Chitrakala, S. (2017). A survey on extractive text summarization. In 2017 international conference on computer, communication and signal processing (ICCCSP), pp. 1-6. https://doi.org/10.1109/ICCCSP.2017.7944061

[11] Kupiec, J., Pedersen, J., Chen, F. (1995). A trainable document summarizer. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 68-73. https://doi.org/10.1145/215206.215333

[12] Conroy, J.M., O'leary, D.P. (2001). Text summarization via hidden markov models. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 406-407. https://doi.org/10.1145/383952.384042

[13] Shen, D., Sun, J.T., Li, H., Yang, Q., Chen, Z. (2007). Document summarization using conditional random

fields. Proc. of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 2862-2867.
[14] Suleiman, D., Awajan, A.A. (2019). Deep learning based extractive text summarization: approaches, datasets and evaluation measures. In 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 204-210. https://doi.org/10.1109/SNAMS.2019.8931813
[15] Ferreira, R., de Souza Cabral, L., Lins, R.D., e Silva, G.P., Freitas, F., Cavalcanti, G.D., ... & Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14): 5755-5764. https://doi.org/10.1016/j.eswa.2013.04.023
[16] Sharma, G., Gupta, S., Sharma, D. (2022). Extractive text summarization using feature-based unsupervised RBM method. In Cyber Security, Privacy and Networking, pp. 105-115. https://doi.org/10.1007/978-981-16-8664-1_10
[17] Shirwandkar, N.S., Kulkarni, S. (2018). Extractive text summarization using deep learning. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1-5. https://doi.org/10.1109/ICCUBEA.2018.8697465
[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.
[19] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[20] Liu, Y., Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.
[21] Sun, C., Qiu, X., Xu, Y., Huang, X. (2019). How to fine-tune bert for text classification?. Proc. of China national conference on Chinese computational linguistics, pp. 194-206. https://doi.org/10.1007/978-3-030-32381-3_16
[22] Narayan, S., Cohen, S.B., Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745.
[23] Lloret, E., Palomar, M. (2012). Text summarisation in progress: a literature review. Artificial Intelligence Review, 37(1): 1-41. https://doi.org/10.1007/s10462-011-9216-z
[24] Baxendale, P.B. (1958). Machine-made index for technical literature—an experiment. IBM Journal of Research and Development, 2(4): 354-361. https://doi.org/10.1147/rd.24.0354
[25] Kupiec, J., Pedersen, J., Chen, F. (1995). A trainable document summarizer. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 68-73. https://doi.org/10.1145/215206.215333
[26] Nobata, C., Sekine, S., Murata, M., Uchimoto, K., Utiyama, M., Isahara, H. (2001). Sentence extraction system assembling multiple evidence. Proc. of NTCIR, pp. 1-6.
[27] Jing, H. (2000). Sentence reduction for automatic text summarization. In Sixth applied natural language processing conference, pp. 310-315.
[28] Gupta, V.K., Siddiqui, T.J. (2012). Multi-document summarization using sentence clustering. In 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1-5. https://doi.org/10.1109/IHCI.2012.6481826
[29] Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In Proceedings of the 25th international conference on Machine learning, pp. 1064-1071. https://doi.org/10.1145/1390156.1390290
[30] Lin, C.Y. (2004). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pp. 74-81.