Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 2430–2437

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Performance of Optimizers in Text Summarization for News Articles

Namrata Kumari[a], Nikhil Sharma[b], Pradeep Singh[c]*

[a]*CSE Department, NIT Hamirpur, 177001, India*
[b]*CSE Department, NIT Hamirpur, 177001, India*
[c]*CSE Department, NIT Hamirpur, 177001, India*

Abstract

Text summarization involves selecting the most important phrases from a given paragraph to produce a concise and pertinent summary. There are two primary approaches to text summarization. While the second is abstractive, the first is extractive. Extractive-based text summarization has reached its peak and as a result, its scope has shrunk, instead of that, let's look at the contrast Text summarization based on abstract concepts, which is inherently dynamic, has attracted a great deal of attention from academics in recent years. The production of a human-like summary has historically required the use of multiple optimizers. These optimizers have expanded text summarization's capabilities. As a result, this paper compares and assesses the effectiveness of two optimizers on a variety of datasets. The two utilized optimizers are adam and rmsprop. This paper compares their performance on various datasets as they are widely employed in text summarization.

Keywords: Type your keywords here, separated by semicolons ;

Everything is changing so quickly in the modern world that we could not have even imagined what we are going through ten years ago. Finding relevant documents among millions of them also gets incredibly challenging as both the size of documents and the number of Internet users grow exponentially [1]. Any document or research paper we look up online will turn up millions of results. Finding the one that satisfies our needs, though, will take some time and be difficult. Even if we found a pertinent document, reading it would be difficult. Text summarization, a subfield

1877-0509
10.1016/j.procs.2023.01.218

of information_retrieval, can be used in this situation. Today, information_retrieval is used broadly, from online stores to Google, which we use every day. It is divided into many sections, one of which is a text summary. The process of text summarization involves reproducing the clearest, most accurate text possible from a longer or larger text document. Essentially, we are trying to reduce the no_of_ lines in text itself while maintaining the quality of text and making sure that it still contains all of the important information from original text document. Because of this, reading the content becomes much easier for humans when the original document is reduced to a smaller no_of_ line. This also makes it easier for humans to find the information they need.

1.1. Related Work

In this field, researchers have exerted significant effort to develop a method for getting text summaries. In the past, numerous techniques have been utilized to determine the text data summary. Beginning with most fundamental algorithms used by researchers in the past, summarizing a text is increasingly performed automatically by machines and software today.

Researchers in this area have put in a lot of time and effort to come up with a system for extracting summaries from texts. Many methods have been used in the past to compile summaries of textual material. Beginning with the earliest algorithms developed by scholars, text summarization is now often carried out automatically by computers and software.

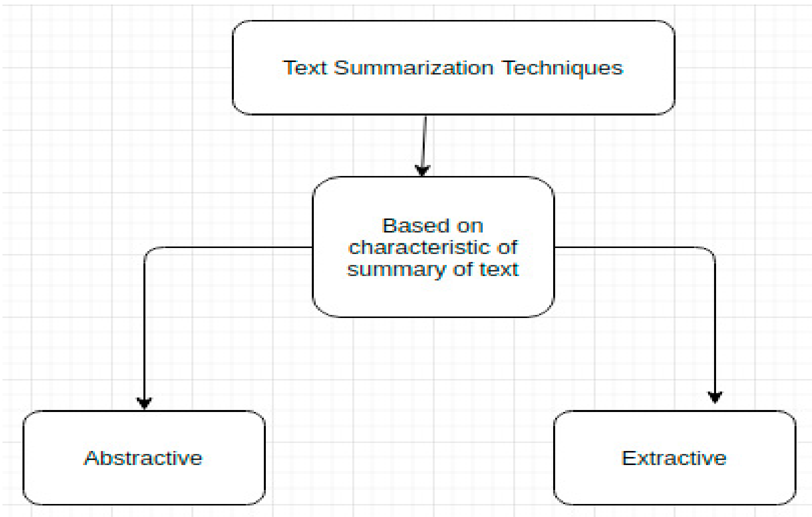

Fig. 1. Types of Text Summarization

Text summarization can be one in two ways. One is known as Extractive text summarisation. In this method, sentences are extracted from document itself. No any additional words are appended by the user; all words and sentences are taken from the document. In contrast, there is a method known as Abstractive text summarization. In these methods, we attempt to generate some of the sentences on our own as well as using the document's sentences.

K. Vimal Kumar et al. presented research on graph-based, automatically generated Hindi text summarization. Based on a weighted graph, its performance measures 79 percent precision, 69 percent recall, and 70 percent F-measure. The classical method was the most well-known and straightforward of the early approaches that were utilized. Compared to the prevalent machine learning approach of today, researchers have made significant progress in this field. Several have been discussed in this article. Table 1 illustrates the distinction between the algorithms based on their advantages and benefits [2].

1.2. Optimization Algorithm

In order to cut down on losses, optimizers are utilized to make adjustments to or updates to learning rates. Let's look at an example to help illustrate our point: a person who is unable to see is climbing a hill in an effort to reach the summit. A person can get to the top of the hill using only their feet after some trial and error based on the hits they get. Optimization algorithms are utilized since doing so reduces the amount of data that is lost in the process. There are several methods for optimization, including gradient descent, stochastic gradient descent, adaptive gradient, adadelta, Adam, and RMSprop. Some of these algorithms are described in more detail below [3].

Table 1. Algorithms on finding text summary of a document

S.No.	Algorithm	Advantages	Disadvantages
1	TF-IDF	Relevant documents returned	1. Work poorly for synonyms and large document 2. redundancy
2	TextRank	1. Short summary generation 2. Based on graph theory	Less Semantic
3	Latent Semantic Method	Extracts related sentences even if they don't share similar words	Calculation is complex
4	Decision Tree	Better quality of summary (uses language generator)	Grammatically incorrect sentences
5	Semantic Based Method	1. Grammatically correct 2. Less redundant summary	Limited to a single document
6	Cluster Based Technique	Used to group similar sentences	Complex algorithm
7	Query Based	Only query based sentences picked	Cannot handle complex queries
8	Multi-Document Summarization	Summary from more than one document	Redundancy
9	Regression Model	Mathematical model	Complex method
10	Ontology Based	Handling unpredictable data	Only domain experts can define ontology
11	Template Based	Logical summary generated	Designing of template is difficult
12	Fuzzy Logic	Use of fuzzifier makes the summary logical	Narrow scope
13	Information Item Based Method	Small, logical and information rich summary	Grammatically incorrect sentences
14	Lead & Body Phrase Method	Semantically good	Grammatical error can occur
15	Machine Learning Based Summarization	Semantically better summary	Complex method

2. Approach

2.1. LSTM

LSTM can overcome the R.N.N. problem because they have memory cells that can retain past information, making them useful for remembering it (Recurrent Neural Network). "The Long Short Term Memory architecture was motivated by an analysis of error flow in existing RNNs which found that long time lags were inaccessible to existing architectures, because backpropagated error either blows up or decays exponentially.[5]" There are several functionally identical neural network cells in the case of R.N.N. The LSTM is more effective at learning previous data because it has more than 1 of the neural-networks (typically four) which interacts with one another in addition to sequentially arranged functional cells. Vanishing gradients cause a problem during R.N.N. backpropagation.

2.2. Vanishing Gradient Problem

The problem of vanishing gradient occurs when the value of gradient becomes very small & no longer contributes to relevant learning[6]. Due to this issue, R.N.N. is not useful for longer sequences. To resolve this issue, LSTM is employed.

Table 2. Optimization Algorithms

S.No.	Algorithm	Advantages	Disadvantages
1	Gradient Descent	Easy computation. Easy to implement. Easy to understand.	May trap at local minima. Weights are changed after calculating gradient on the whole dataset. So, if the dataset is too large than this may take years to converge to the minima. Requires large memory to calculate gradient on the whole dataset.
2	Stochastic Gradient Descent	Frequent updates of model parameters hence, converges in less time. Requires less memory as no need to store values of loss functions. May get new minima.	High variance in model parameters. May shoot even after achieving global minima. To get the same convergence as gradient descent needs to slowly reduce the value of learning rate.
3	Adaptive Gradient	Learning rate changes for each training parameter. Don't need to manually tune the learning rate. Able to train on sparse data.	Computationally expensive as a need to calculate the second order derivative. The learning rate is always decreasing results in slow training.
4	Momentum	Reduces the oscillations and high variance of the parameters. Converges faster than gradient descent.	One more hyper-parameter is added which needs to be selected manually and accurately.
5	Adadelta	Now the learning rate does not decay and the training does not stop.	Computationally expensive.
6	Adam	The method is too fast and converges rapidly. Rectifies vanishing learning rate, high variance.	The method is too fast and converges rapidly. Rectifies vanishing learning rate, high variance.
7	RMSprop	It is a very robust optimizer which has pseudo-curvature information. Additionally, it can deal with stochastic objectives very nicely, making it applicable to mini batch learning. It converges faster than momentum.	Learning rate is still manual, because the suggested value is not always appropriate for every task. Implementation of RMSProp Descent with Employee Attrition

2.3. Dataset

Two datasets were used in this paper. "News summary dataset" is the name of the first dataset. The headlines and news columns in this dataset each have 98280 and 98360 distinct values, respectively. A CSV file containing articles and highlights makes up the second dataset. This information is set out in a row-column format and includes news-news_summary.

3. Results with NEWS summary dataset

3.1. RMSprop Optimizer

Earlier gradients are utilised by this optimizer to normalise the gradient. For example, for high gradients, there is a decrease in step to ignore the exploding, while for small gradients, the step size is increased to ignore the vanishing[8]. The graph below illustrates the results achieved by various optimizers. Using the Bidirectional LSTM model and the rmsprop optimizer, system achieved an accuracy of 37.91%

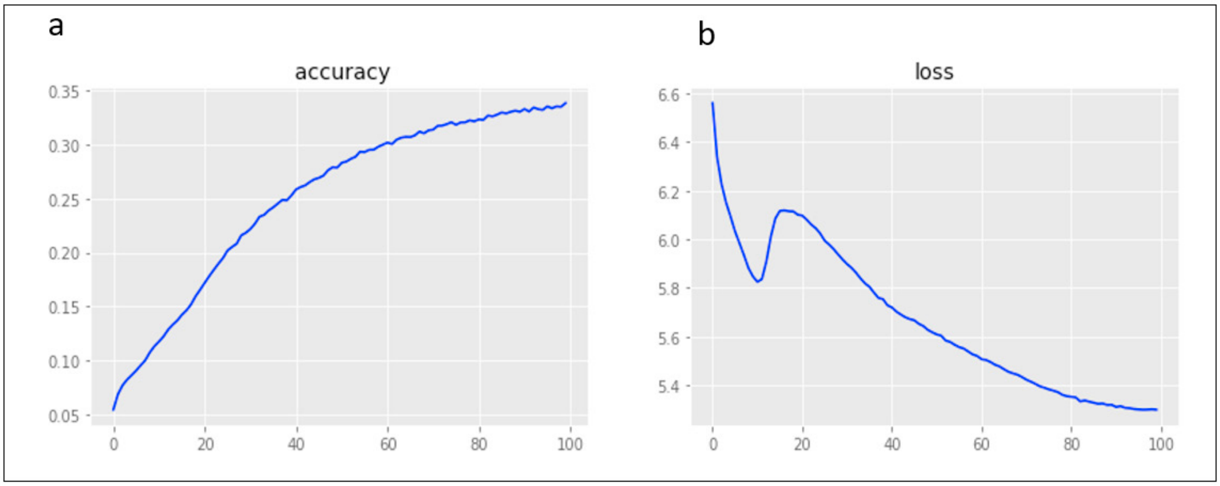

Fig. 2. (a) Accuracy graph for Rmsprop Optimizer; (b) Loss graph for Rmsprop Optimizer

3.2. Adam Optimizer

Adam(adaptive moment estimation), which dramatically lowers cost of neural networks. The main goal of this algorithm is to handle sparse gradients in noisy problems by combining the best aspects of AdaGrad and RMSProp. The graph below illustrates the results achieved by various optimizers[9]. number of epochs in the loss graph are shown on x axis, and y axis represents loss during each epoch. Using the Bidirectional LSTM model and Adam optimizer, system achieved an accuracy rate of 81.60%.

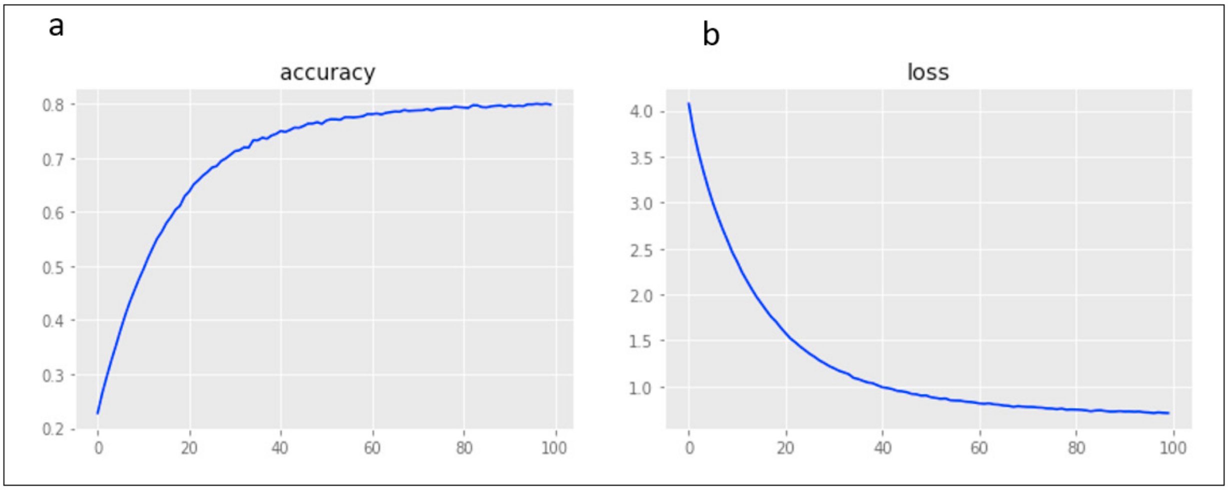

Fig. 3. (a) Accuracy graph for adam Optimizer; (b) Loss graph for adam Optimizer

Table 3. Techniques and their accuracy

Technique Used	Maximum Accuracy Achieved (in %)
Bi-directional LSTM Model with 'rmsprop' optimizer	33.98
Bi-directional LSTM Model with 'adam' optimizer	80.78

The table 3 above displays the maximum precision achieved by the two optimizers utilised. At a dropout value of 0.1, rmsprop optimizer's maximum accuracy of 33.98 percent has been reached. With the 'adam' optimizer, however, an improvement of 80.78 percent was achieved.

4. Results with CNN Daily Mail dataset

4.1. RMSprop Optimizer

The graph below illustrates the results achieved by various optimizers. Using the Bidirectional LSTM model and the rmsprop optimizer, system achieved an accuracy of 55.09%.

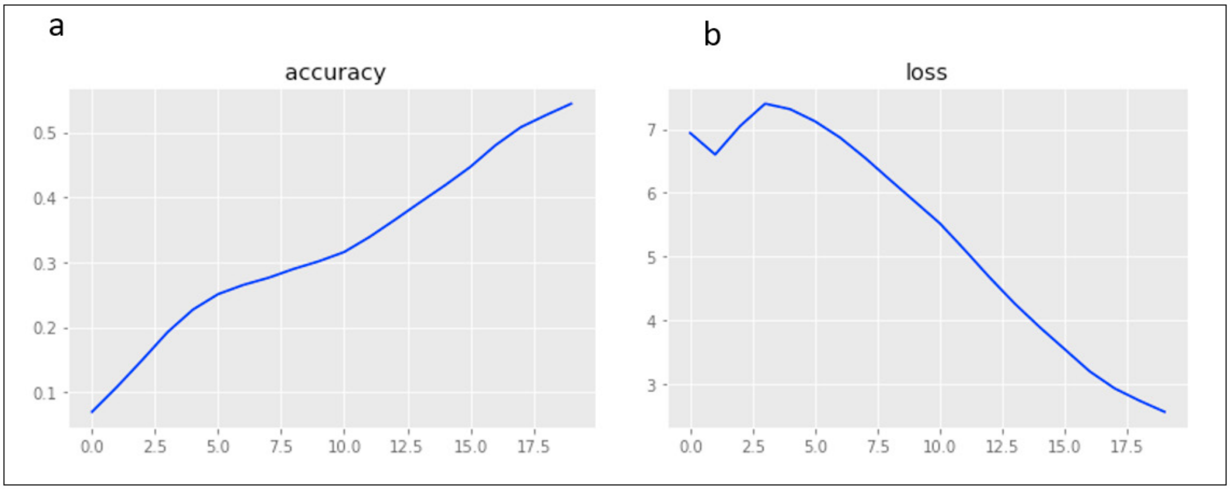

Fig. 4. (a) Accuracy graph for Rmsprop Optimizer; (b) Loss graph for Rmsprop Optimizer

4.2. Adam Optimizer

The graph below illustrates the results achieved by various optimizers. The no_of_epoch are represented on x axis, and loss is shown on y aixs during each epoch. Using the Bidirectional LSTM model and Adam optimizer, system attained a 96.08% accuracy rate.

Table 4. Techniques and their accuracy

Technique Used	Maximum Accuracy Achieved (in %)
Bi-directional LSTM Model with 'rmsprop' optimizer	55.09
Bi-directional LSTM Model with 'adam' optimizer	96.08

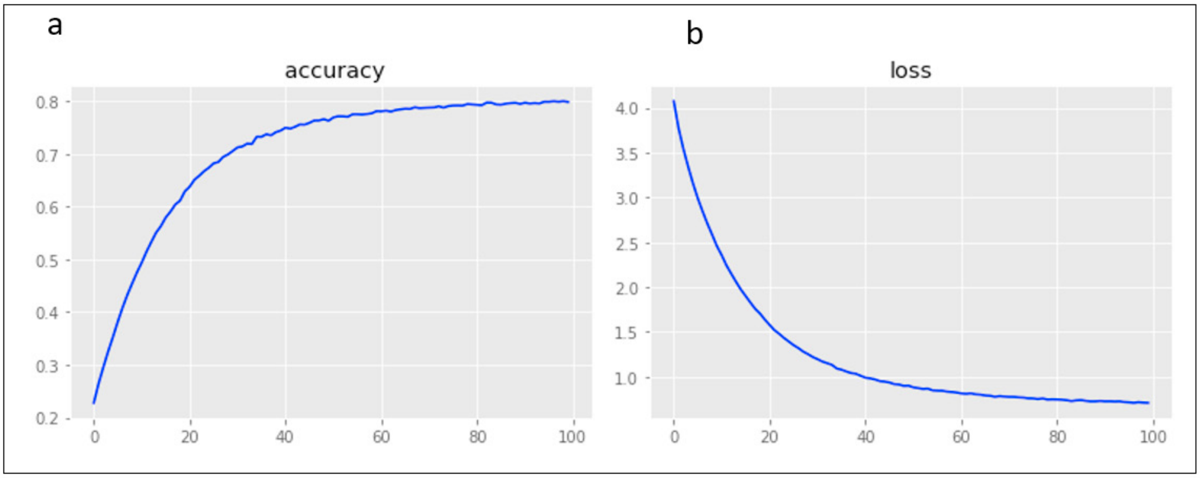

Fig. 5. (a) Accuracy graph for adam Optimizer; (b) Loss graph for adam Optimizer

The table 4 above displays the maximum precision achieved by the two optimizers utilised. At a dropout value of 0.1, rmsprop optimizer's maximum accuracy of 55.09 percent has been reached. With the 'adam' optimizer, however, an improvement of 96.08 percent was achieved.

5. Evaluation Metrics

Table 5. Evaluation Metrics

	Precision	Recall	F1
Rouge-1	0.757	0.935	0.854
Rouge-2	0.696	0.895	0.791
Rouge-3	0.698	0.897	0.769

From the above table 5, it is evident that different evaluation metrics have been used to check the accuracy of the text summarizing technique. Thereafter, various scores have been calculated.

$$Precision = \frac{RP}{RP + FP} \tag{1}$$

$$Recall = \frac{RP}{RP + FN} \tag{2}$$

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \tag{3}$$

Where,

Real Positive, False Positive, and False Negative are written as R.P., F.P., and F.N. respectively.

The preceding table 6 demonstrates that the Bi-directional LSTM model with optimizer 'adam' has proven to be the most effective, as it has produced the highest mean accuracy with the CNN DailyMail dataset. In both datasets, the 'adam' optimizer performed better than the 'rmsprop' optimizer. Because adam optimizer combines the most effective aspects of rmsprop and adagrad, it is more effective than rmsprop.

Table 6. Techniques and their mean accuracy

Dataset	Techniques Used	Mean Accuracy Achieved
News Summary Dataset	Bi-directional LSTM Model with 'rmsprop' optimizer	23.90
	Bi-directional LSTM Model with 'adam' optimizer	63.76
CNN Dailymail Dataset	Bi-directional LSTM Model with 'rmsprop' optimizer	29.77
	Bi-directional LSTM Model with 'adam' optimizer	71.09

6. Conclusion

There are numerous papers available, making it take more time to read about a single subject. Here, we discuss various methods that have been applied to this area of research thus far. This paper discusses various extractive-based and abstractive-based techniques, as well as the advantages and disadvantages of each. One of the abstractive-based techniques is employed in this paper in order to the desired results. In future, a researcher can make use of a variety of popular Abstractive-based techniques because they are dynamic by nature. The ideal point at which abstractive techniques produce the best results is still being sought despite the fact that they automatically summarise texts. In this study, two optimizers—adam and rmsprop—produced various results when used on various data sets. Because it combines two algorithms, Adam optimizer performs better than rmsprop optimizer. With a variety of other techniques and optimizers, these datasets can be used to test whether future results will be optimal.

References

[1]. Pardeep Singh, N. K. A. G. R. S. (2019). A Supervised Approach for Keyphrase Extraction using SVM. International Journal of Control and Automation, 12(5), 469 - 477. Retrieved from http://sersc.org/journals/index.php/IJCA/article/view/2653

[2]. Kumari, N., & Singh, P. (2021). Text Summarization and Its Types: A Literature Review. Handbook of Research on Natural Language Processing and Smart Service Systems, 368-378.

[3]. Mersmann, O., Trautmann, H., Naujoks, B., & Weihs, C. (2010, July). Benchmarking evolutionary multiobjective optimization algorithms. In IEEE Congress on Evolutionary Computation (pp. 1-8). IEEE.

[4]. Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

[5]. Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), 602-610.

[6]. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107-116.

[7]. Huang, F., Li, X., Yuan, C., Zhang, S., Zhang, J., & Qiao, S. (2021). Attention-emotion-enhanced convolutional LSTM for sentiment analysis. IEEE transactions on neural networks and learning systems.

[8]. Huk, M. (2020, March). Stochastic optimization of contextual neural networks with RMSprop. In Asian Conference on Intelligent Information and Database Systems (pp. 343-352). Springer, Cham.

[9]. Zhang, Z. (2018, June). Improved adam optimizer for deep neural networks. In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS) (pp. 1-2). Ieee.

[10]. Kumari, N., & Singh, P. (2020). Automated Hindi Text Summarization Using Tf-Idf And Textrank Algorithm. Journal of Critical Reviews, 7(17), 2547-2555.

[11]. R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts", 2010.

[12]. K.M.S. Kumar and R. Soumya, "Text summarization using clustering technique", 2015.

[13]. A. Ajay, S. Rajendran, V. Shravan, "Query-based Text Summarization", 2020.

[14]. K.S. Thakkar, R.V. Dharaskar, M. Chandak, "Graph-Based Algorithms for Text Summarization", 2010.

[15]. P.M. Sabuna, D.B. Setyohadi, "Summarizing Indonesian text automatically by using sentence scoring and decision tree", 2017.