Nigerian Journal of Physics (NJP)
ISSN online: 3027-0936
ISSN print: 1595-0611
DOI: https://doi.org/10.62292/njp.v35i2.2026.592
Volume 35(2), June 2026

# An Optimized Stochastic Gradient Descent Approach to a Bidirectional Long Short-Term Memory with Bidirectional Contextual Embeddings for Extractive Text Summarisation

***[1]Abdullah, Khadijha-Kuburat A., [1]Odule, Tola J., [1]Ajayi, Ayobami J., [2]Oladiran, Omobola E., [1]Lawal, Olufunmilayo A., [3]Ologunleko Emmanuel F. and [1]Tijani Olatunde D.**

[1]Department of Computer Sciences, Faculty of Sciences, Olabisi Onabanjo University, Ago Iwoye, Ogun State, Nigeria.
[2]Department of Statistics, Faculty of Sciences, Olabisi Onabanjo University, Ago Iwoye, Ogun State, Nigeria.
[3]COPINE, Advance Space Technology Application, OAU Ile-Ife.

*Corresponding Author's Email: abdullah.adebisi@oouagoiwoye.edu.ng Phone: +2348060046592

**ABSTRACT**

With the increase in the amount of textual data on the web, this study explores the performance of extractive text summarisation model that integrates pretrained contextual word embeddings with Bidirectional Long Short-Term Memory (BiLSTM) encoder–decoder architecture. The embeddings capture context and semantic relationships, while the BiLSTM mechanism addresses the vanishing gradient problem and enables learning of long-term dependencies in both directions. Experiments were conducted on subsets of the Amazon Fine Food Reviews dataset of 5000 samples. The model was trained using Stochastic Gradient Descent to optimise with a learning rate of 0.05 across 10, 20, and 30 epochs. From the results, it shows that at 10 epochs, training and validation metrics are consistent and matched, indicating good generalisation with minimal overfitting. As the epoch increases, training loss decreases significantly; however, validation loss increases as dataset sizes increase with overfitting. Though, training performance improves dramatically, but validation performance deteriorates. The findings demonstrate that the training enhances memorisation of summarised text but required early stopping and careful epoch selection to handle generalisation in extractive text summarisation tasks.

**Keywords**:
BiLSTM,
Contextual embedding,
Stochastic gradient descent,
Encoder-decoder.

## INTRODUCTION

The rapid growth of digital information has made access to textual content, large volumes of text are generated daily from sources such as news platforms, academic publications, technical reports, emails, and social media. This abundance of information is valuable, and has created a serious challenge such as information overload. These make users to find it difficult to read, analyse, and extract useful knowledge from lengthy documents within a limited time. Automatic summarisation systems help users understand documents quickly, reduce reading time, and support efficient decision-making in various domains such as information retrieval, education, healthcare, and business intelligence (Sharma & Sharma, 2022). Abstractive methods generate more natural summaries, often complex, computationally expensive, and prone to generate inaccurate or misleading information. As a result, extractive text summarisation remains widely adopted in real-world applications, selects salient sentences from a document while preserving meaning and coherence, make summaries more reliability and interpretability (Zhang et al, 2023). Most early studies on extractive summarisation relied on statistical features (TF-IDF) and graph-based models (TextRank) such methods are computationally simple, suffer from curse of dimensionality, increases computational complexity but results in out-of-vocabulary (OOV) and poor generalisation problems. These approaches often failed to capture deep semantic relationships and leads to generation of summaries inconsistent with the original text (Ju et al, 2021).

Recent studies focus on improving semantic representation with contextual embeddings to enhanced extractive text summarisation. Contextual embeddings is a major advancement in Natural Language Processing (NLP) that shows a significant impact on language understanding systems to capture semantic similarity. This enables models to generate dynamic word

representations based on surrounding context rather than static embeddings such as Word2Vec and GloVe. Arora et al. (2020) presented contextual embeddings as it outperforms traditional methods with significant improvement in tasks involving complex linguistic structures, ambiguity and rare or unseen words. Zhong et al. (2020) described a semantic matching that match candidate summaries to documents in embedding space, thus, improves performance over sentence-level extraction. Similarly, Turton et al. (2021) presented contextual embeddings that can encode deep semantic features across model layers, improving interpretability and representation quality. Traditional Recurrent Neural Networks (RNNs) suffer from the vanishing and exploding gradient problems, limiting its ability to learn long-term dependencies. The introduction of Long Short-Term Memory (LSTM) networks by Hochreiter & Schmidhuber (1997) marked a significant advantage in addressing these limitations. Bidirectional Long Short-Term Memory (BiLSTM) networks model sequential dependencies to captures dependencies of long-range sequence-level dependencies in forward and backward directions and contextual relationships between sentences (Vo et al, 2024). Bano et al, (2023) demonstrated the combination of contextual embeddings with a BiGRU network for extractive summarization model to improve document-level representation and sentence selection accuracy. Long Short-Term Memory (LSTM) (Ghojogh & Ghodsi (2023), and Gated Recurrent Units (GRU) have been identified as promising approaches for text summarisation, achieved improved performance with minimal human intervention when trained on large datasets (Yadav et al, 2022; Yaday et al, 2023). In imbalanced datasets, Aduragba et al. (2020) enhanced contextual embeddings with BiLSTM and fine-tuning significantly improved classification accuracy. Alghamdi and Alzahrani (2024) demonstrated a topic-based BERT model to enhance extractive summarisation by integrating sentence and topic representations in improving semantic consistency and document-level coherence.

This study demonstrates the enhancement of bidirectional language contextual embeddings with BiLSTM in sentence-level semantic understanding of extractive text summarisation. The major goal of this approach is to enable the model to generate context-sensitive embeddings, where the same words have different vector representations depending on its usage. This improves sentence-level importance estimation while preserving contextual and structural coherence by capturing sequential dependencies for extractive summarization tasks and improves accuracy. To ensure consistency and model readiness,. the encoder-decoder networks are jointly optimised using Stochastic Gradient Descent (SGD), with the objective of minimising the negative log-likelihood loss function computed using a softmax output layer.

## MATERIALS AND METHOD

In this section, bidirectional contextual embeddings Language Model (biLM) with sequence BiLSTM is considered as shown in figure 1. The datasets are converted to tokens using the keras.preprocessing.text.Tokenizer class. The datasets were downloaded online from amazon reviews (https://www.kaggle.com/snap/amazon-fine-food-reviews/) which are in excel format. It contains 10 features such (Id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary, Text). Two (2) features (summary and text) were selected from the 10 features and other features were dropped, as these are not needed for the implementation.

### Data Pre-processing and Embedding

The document is pre-processed (tokenization, removal of stopword, normalisation.) by dividing a continuous text document into sentence segmentation, each segmented sentence transforms raw text into structured sentence units for encoding. The variable sentence lengths is handled by paddle the sentences to zero values so that multiple filters can extract features and map them into fixed-length representations.

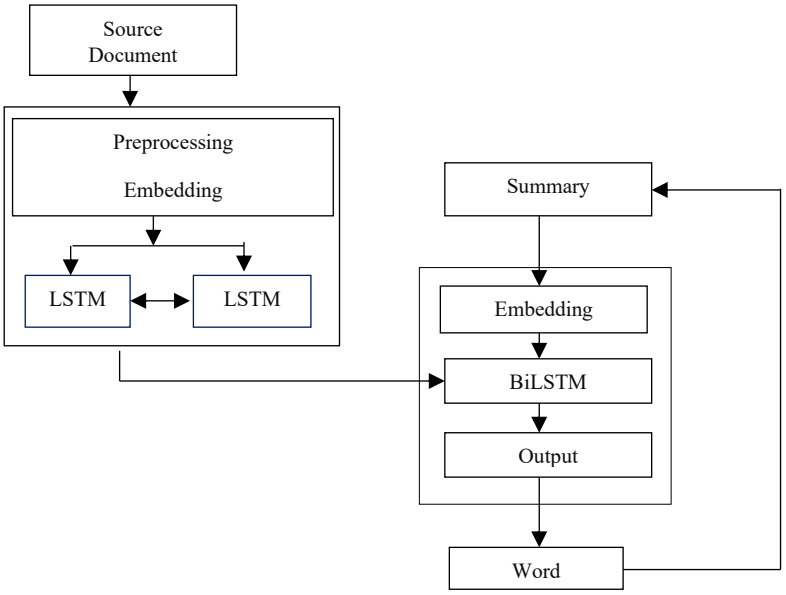


Figure 1: Text Summarisation with Contextual Word Embedding using Bidirectional Encoder-Decoder LSTM

**Word Representation with Bidirectional Contextual Embedding**

After pre-processing, the sentences are sent as input sequences to the embedding layer. This is passed into a Bidirectional contextual encoding model to generate sentence-level embedding, thus, operates on structured inputs. Hence, an input sequence of word embeddings $x = (x_1, \dots, x_n)$, where $x_i \in \mathbb{R}^d$, is transformed through nonlinear operations at each time steps $(t)$ to generate contextual vector representations in both forward and backward states with LSTM $\leftrightarrow$ LSTM language model in the learning process to capture context-dependent of word, performs word sense disambiguation and investigate the effectiveness of structure of recurrent. To account for a contextual dependency, the hidden sequence is computed for left and right $(\vec{h}, \overleftarrow{h})$ respectively as equation 1 and 2 with output in equation 3:

$$\vec{h}_t = \tau_H(w_{x\vec{h}}x_t + \vec{V}\vec{h}_{t-1} + b_{\vec{h}}) \quad (1)$$

$$\overleftarrow{h}_t = \tau_H(w_{x\overleftarrow{h}}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (2)$$

$$y_t = \vec{U}\vec{h}_t + \overleftarrow{U}\overleftarrow{h}_t + c \quad (3)$$

w, U and V are parameters with c as bias at different directions with a weighted average pooling operation.

**BiLSTM Network Model**

The training word vectors are initialised by a pre-trained model to captures syntactic and semantic word relationships. Long Short-Term Memory (LSTM) consists of memory cell state ($c_t$) to remember values over time interval and connected to three (3) gates; forget gate $(\tau_i)$, input gate $(i_i)$, and output gate $(o_i)$, with activation function $(\delta_t)$ at interval time $(t)$ to overcome the vanishing gradient problem and captures long term dependencies in text summarisation. According to Graves et al., (2013) described the LSTM network in equations 4:

$$\left.\begin{aligned} i_t &= \sigma(w_i x_t + V_i h_{t-1} + +b_i \\ \tau_t &= \sigma(w_\tau x_t + V_\tau h_{t-1} + +b_\tau \\ o_t &= \sigma(xo_t + V_o h_{t-1} + b_o) \\ \hat{\mu}_t &= tanh(w_{\hat{\tau}} x_t + V h_{t-1} + b_{\hat{\mu}}) \\ \mu_t &= \tau_t^o \mu_{t-1} + i_t^o \hat{\mu}_t \end{aligned}\right\} \quad (4)$$

The bidirectional LSTM is used at different epochs to investigate the effectiveness of the languages model. This involves using encoder and decoder bidirectional LSTM, an encoder reads the input sequence $x$ and generates a hidden state $\vec{h}_t$, and decoder generate $\overleftarrow{h}_t$. Hence, the output units is sum up to one word probabilities using softmax activation function. Finally, a logistic layer makes a binary decision as to whether it should be included in the summary taking into consideration previous decisions. However, SGD performed parameter update for each training samples with the objective of minimising the negative log-likelihood (convergence of the training loss) as in equation 5 to improve accuracy, then, updates the parameter weights iteratively via back-propagation

$$L = -\sum_{t=1}^{T} log\ P\ (y_t | y_{<t}, x) \quad (5)$$

Where: $y_t$ = target token, $y_{<t}$ = previous token, $x$ = input sequence

**Experimental Setup**

The model was trained on the Amazon Fine Food Reviews dataset, from which two fields, Text and Summary, were retained. Five subsets of sizes 1000,

2000, 3000, 4000 and 5000 reviews were constructed, each split 70 percent for training and 30 percent for validation. Three training regimes were investigated at 10, 20 and 30 epochs, with an initial learning rate of 0.05 using Stochastic Gradient Descent as the optimiser. The architecture comprises pretrained contextual embeddings feeding a Bidirectional LSTM encoder, an LSTM decoder with sigmoid activation for sentence scoring, and a softmax output layer that minimises the negative log-likelihood.

***Training and Validation Performance***

From Tables 1, 2 and 3, the training loss, training accuracy, validation loss, and validation accuracy for 10, 20, and 30 epochs are presented respectively. It is therefore a proxy for how well the model fits the sequence distribution of the reference summaries and should not be interpreted as summarisation quality directly.

**Table 1: Shows the Loss function and Accuracy for Epoch 10**

| Dataset | Epoch =10 for Training= 0.7 and Validation= 0.3 | | | |
|---|---|---|---|---|
| | Training loss | Accuracy | Validation loss | Accuracy |
| 1000 | 1.0211 | 0.8504 | 1.5228 | 0.8484 |
| 2000 | 0.9398 | 0.8540 | 1.2457 | 0.8596 |
| 3000 | 0.9533 | 0.8554 | 1.3652 | 0.8496 |
| 4000 | 0.9793 | 0.8545 | 1.2357 | 0.8540 |
| 5000 | 1.0070 | 0.8530 | 1.2602 | 0.8522 |

**Table 2: Shows the Loss function and Accuracy for Epoch 20**

| Dataset | Epoch =20 for Training= 0.7 and Validation= 0.3 | | | |
|---|---|---|---|---|
| | Training loss | Accuracy | Validation loss | Accuracy |
| 1000 | 0.7583 | 0.8567 | 1.6421 | 0.8386 |
| 2000 | 0.6754 | 0.8655 | 1.4444 | 0.8476 |
| 3000 | 0.6961 | 0.8673 | 1.5914 | 0.8343 |
| 4000 | 0.8434 | 0.8607 | 1.3418 | 0.8466 |
| 5000 | 0.5507 | 0.8943 | 1.4941 | 0.8379 |

**Table 3: Shows the Loss function and Accuracy for Epoch 30**

| Dataset | Epoch =30 for Training= 0.7 and Validation= 0.3 | | | |
|---|---|---|---|---|
| | Training loss | Accuracy | Validation loss | Accuracy |
| 1000 | 0.4596 | 0.8895 | 1.8353 | 0.8311 |
| 2000 | 0.3745 | 0.9121 | 1.5632 | 0.8445 |
| 3000 | 0.4676 | 0.9007 | 1.8046 | 0.8263 |
| 4000 | 0.5169 | 0.899 | 1.5559 | 0.8343 |
| 5000 | 0.0264 | 0.9958 | 1.7854 | 0.8334 |

At 10 epochs, training and validation accuracy are closely aligned across all dataset sizes. Differences between the two quantities are below one percentage point in every setting, which indicates that after 10 epochs the model has fitted the data without memorising idiosyncrasies of the training partition. Training loss remains in the range of 0.9398 to 1.0211.

At 20 epochs, training loss falls substantially (as low as 0.5507 on the 5000-sample subset), while validation loss increases compared with the 10-epoch setting for the 1000, 3000, and 5000 sample subsets.

At 30 epochs, divergence between training and validation behaviour becomes severe. On the 5000-sample subset the model reaches a training loss of 0.0264 and training accuracy of 0.9958, while validation loss rises to 1.7854 and validation accuracy

Figure 2 presents validation loss and validation accuracy as grouped bars across the five dataset sizes, with the three bars per group corresponding to epochs 10, 20 and 30. Figure 3 presents the same structure for training loss and training accuracy.

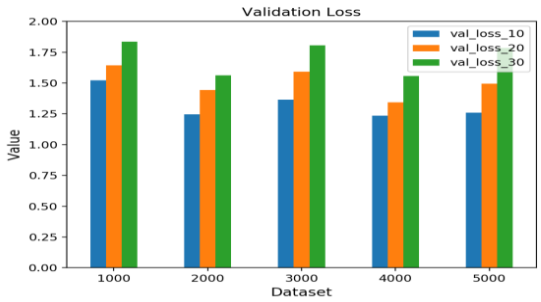


Figure 2a: Validation Loss at Epochs 10, 20 and 30 for each dataset size

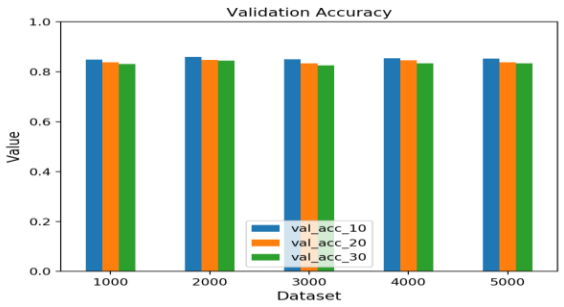


Figure 2b: Validation Accuracy at Epochs 10, 20 and 30 for each dataset size

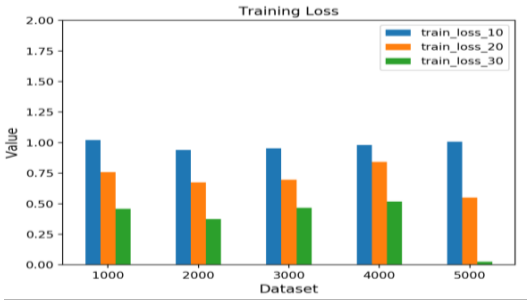


Figure 3a: Training Loss at Epochs 10, 20 and 30 for each dataset size

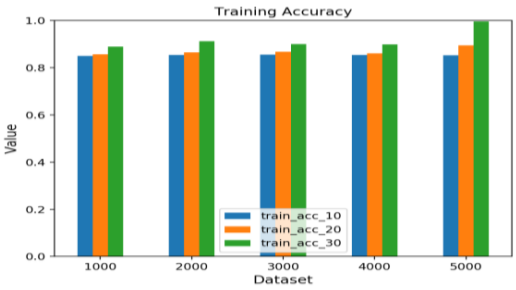


Figure 3b: Training Accuracy at Epochs 10, 20 and 30 for each dataset size

Figures 2a and 3a, the training loss bars shrink from epoch 10 to epoch 30 in every dataset, while the validation loss bars grow from epoch 10 to epoch 30 in every dataset, The divergence between these two directions is the defining signal of overfitting. Figures 2b and 3b tell the same story for accuracy: training accuracy bars grow with epochs while validation accuracy bars shrink or stagnate. Section 3.3 quantifies this effect.

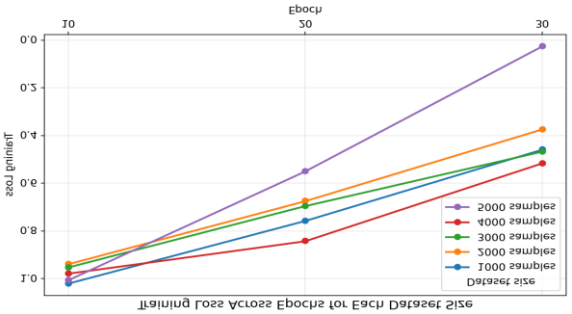

Figure 4: Training loss across epochs for each dataset size. Training loss decreases monotonically with epochs in every subset, most dramatically for 5000 samples, which drops from 1.0070 at epoch 10 to 0.0264 at epoch 30

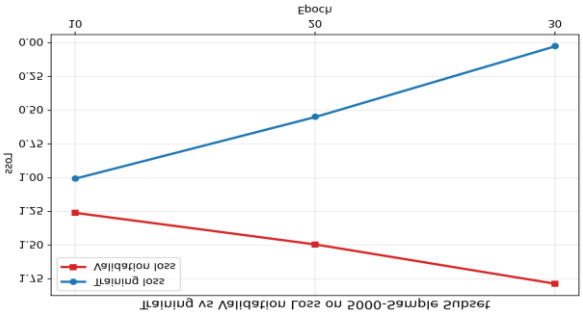

Figure 5: Training loss versus validation loss on the 5000-sample subset across epochs. Training loss drops by more than 97 percent while validation loss rises by more than 40 percent, a textbook overfitting signature on the most data-rich configuration tested

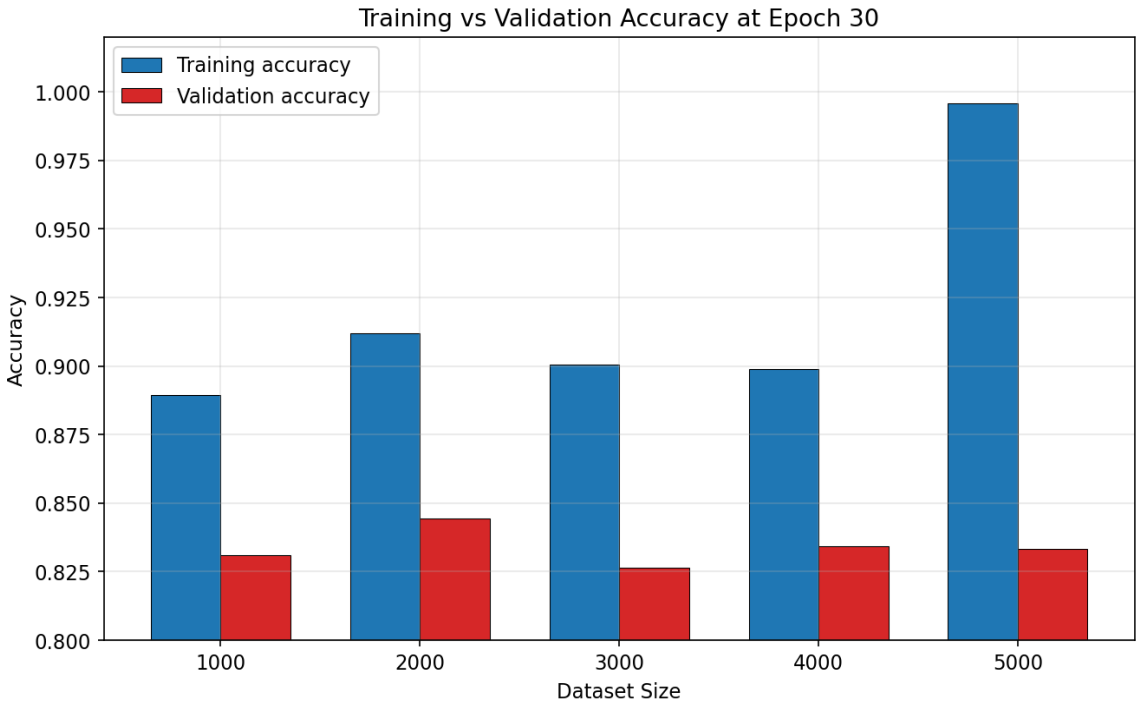


Figure 6: Training versus validation accuracy at epoch 30 across all dataset sizes. The bars diverge most strongly on the 5000-sample subset (99.58 percent versus 83.34 percent), showing that the overfitting problem does not self-correct when more training data is supplied under the current configuration.

**Discussion**

The contextual embedding initialised BiLSTM encoder-decoder fits the Amazon Fine Food Reviews training distribution to very low loss by epoch 30, but the validation-side evidence shows that generalisation does not track this improvement. Increasing the number of epochs without compensating for regularisation degrades held-out performance. Increasing the training corpus beyond 5000 samples is expected to shift the onset of overfitting to later epochs; combined with dropout, weight decay, and early stopping, this is the most direct path to improved generalisation.

**CONCLUSION**

The current experimental results establish that the model can be fitted to the training data, but the ceiling on validation accuracy (approximately 0.860 across all configurations and epochs) and the rising validation loss at later epochs both indicate that the proposed configuration is at or near the limit of what can be learned under the current optimisation regime and embedding architecture. The deficiencies embedded in the language model can be improved by better fine-tuning the model with 3 or more deep layers' exploration for training a large dataset

**REFERENCES**

Aduragba, O. T., Yu, J., Senthilnathan, G., & Cristea, A. I. (2020). Sentence contextual encoder with BERT and BiLSTM for automatic classification with imbalanced medication tweets. *In* Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task (SMM4H 2020) *(pp. 155–159). Association for Computational Linguistics.*

Alghamdi, N. S., & Alzahrani, S. M. (2024). Improving extractive summarization with semantic enhancement through topic-injection based BERT model. *Knowledge-Based Systems, 292, 111626.* https://doi.org/10.1016/j.knosys.2024.111626

Arora, S., May, A., Zhang, J., & Ré, C. (2020). Contextual embeddings: When are they worth it? *In* Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020) *(pp. 2650–2663). Association for Computational Linguistics.* https://doi.org/10.18653/v1/2020.acl-main.236

Bano, S., Khalid, S., Tairan, N. M., Shah, H., & Khattak, H. A. (2023). Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach. *Journal of King Saud University – Computer and Information Sciences, 35(9), 101739.* https://doi.org/10.1016/j.jksuci.2023.101739

Ghojogh, B., & Ghodsi, A. (2023). Recurrent neural networks and long short-term memory networks: Tutorial

and survey *(arXiv:2304.11461). arXiv.* https://arxiv.org/abs/2304.11461

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks*. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6645–6649). IEEE. https://doi.org/10.1109/ICASSP.2013.6638947

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8), 1735–1780.* https://doi.org/10.1162/neco.1997.9.8.1735

Ju, J., Liu, M., Koh, H. Y., Jin, Y., Du, L., & Pan, S. (2021). Leveraging information bottleneck for scientific document summarization. *In* Findings of the Association for Computational Linguistics: EMNLP 2021 *(pp. 4091–4098). Association for Computational Linguistics.* https://doi.org/10.18653/v1/2021.findings-emnlp.345

Sharma, G., & Sharma, D. (2022). Automatic text summarization methods: A comprehensive review. *SN Computer Science, 4(1), 33.*

Turton, J., Smith, R. E., & Vinson, D. (2021). Deriving semantic features from contextual embeddings. *In* Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021) *(pp. 203–214). Association for Computational Linguistics.* https://doi.org/10.18653/v1/2021.repl4nlp-1.26

*Vo, S.-N., Vo, T.-T., & Le, B. (2024).* Interpretable extractive text summarization with meta-learning and BiLSTM: A study of meta-learning and explainability techniques. *Expert Systems with Applications, 245, 123045.* https://doi.org/10.1016/j.eswa.2023.123045

Yadav, A. K., Singh, A., Dhiman, M., Vineet, Kaundal, R., Verma, A., & Yadav, D. (2022). *Extractive text summarization using deep learning approach*. International Journal of Information Technology, 14(5), 2407–2415.

Yadav, A. K., Ranvijay, Y., Yadav, R. S., & Maurya, A. K. (2023). *State-of-the-art approach to extractive text summarization: A comprehensive review*. Multimedia Tools and Applications, 82(19), 29135–29197.

Zhang, S., Wan, D., & Bansal, M. (2023). Extractive is not faithful: An investigation of broad unfaithfulness problems in extractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2153–2174). Association for Computational Linguistics.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., & Huang, X. (2020). *Extractive summarization as text matching*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)* (pp. 6197–6208). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.552