VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2025, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 13, Number 1, January-March 2025 pp:49-59

Next Word Prediction for Urdu using Deep Learning Techniques

M Haroon Ahmad [1], **Ali Saeed** [2], **M Usman Bhatti** [2*], **Naveed Hussain** [2], **Muhammad Farhat Ullah** [3,4], **Mehmood Anwar** [4]

[1]Department of Software Engineering, Ripha Internation University Lahore, Punjab, Pakistan;
[2]Department of Software Engineering, FOIT, University of Central Punjab, Lahore, Pakistan;
[3]School of Software, Dalian University of Technology, Dalian, Ganjingzi District, Liaoning Province, China; [4]Department of Software Engineering, FOIT, University of Lahore, Punjab, Pakistan;
[4]Faculty of Computer Science Mathematics, Universiti Malaysia Terengganu

Keywords: *Language Modeling, Next-Word Prediction, Deep Learning, Urdu Text Processing*

Journal Info:
Submitted: January 14, 2025
Accepted: February 23, 2025
Published: February 27, 2025

Abstract A language model for next-word prediction is a probabilistic representation of a natural language that utilizes text corpora to generate word probabilities. These models play a crucial role in text generation, machine translation, and question-answering applications. The focus of this study is to develop an improved algorithm for next-word prediction in Urdu. The study implements deep learning models, including RNN, LSTM, and Bi-LSTM, on a subset of the Ur-Mono Urdu corpus containing 3,000 and 5,000 sentences. To prepare the data for experimentation, tokenization and stemming data cleaning techniques are applied. The study achieved an accuracy of 87% using the RNN model on the first 3,000 sentences of the Ur-Mono dataset and 84% accuracy using the RNN model on the first 5,000 sentences of the Ur-Mono dataset. In conclusion, it can be stated that when the corpus size is small, the RNN outperforms both the LSTM and Bi-LSTM. However, as the corpus size increases, the Bi-LSTM exhibits superior performance compared to both RNN and LSTM.

***Correspondence author email address:** muhammad.usman1@ucp.edu.pk
DOI: 10.21015/vtse.v13i1.2044

1 Introduction

Next-word prediction research aims to develop and improve algorithms, models, or systems that can accurately predict the most likely word or words that will follow a given input or context in a sequence of text [1]. Next-word prediction has been previously implemented in other languages, but it has received less attention in the context of Urdu [2]. With over 300 million people [3], using Urdu in their everyday lives [4, 5]. Most languages are written from left to right, while Urdu is written from right to left, just like Arabic. The next-word prediction is a challenging task due to its syntax and ligature structures [6]. The Urdu language follows the subject, object, and verb order. To achieve good accuracy, the language models need to learn a state-of-the-art dataset. The research on next-word prediction has various practical applications, including enhancing efficiency in data entry [7, 8], improving accessibility [9], improving spelling and grammar mistakes [10], facilitating language learning [11], enabling content generation [12], streamlining text summarization [13], enhancing machine translation [14], and expediting the handling of legal documents in courts [15, 16]. This research addresses unexplored aspects of next-word prediction for the Urdu language.

In the literature, a variety of approaches have been employed for the next-word prediction task, such as statistical modeling [17], n-gram modeling [18], hidden Markov models [19], knowledge-based approaches [20], neural networks and deep learning techniques [21], not only for English but also for various international languages. However, the most promising results have been consistently attained through the utilization of deep learning methods like Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), and Bidirectional Long Short Term Memory (Bi-LSTM) [21, 22]. Therefore, this study also has chosen to implement deep learning approaches for the task of Urdu next-word prediction.

More than 300 million around the globe use Urdu for both speaking and understanding, as well as for their daily communications with each other [23]. The objective is to identify the best Urdu dataset for training our model, a dataset that consists of the vocabulary of daily life, including different domains like sports, politics, religion, etc. Therefore, we selected the UrMono [24] dataset of Urdu for experimentation in this study. Initially, we preprocess the dataset with data cleaning, aiming to determine the most suitable methods for our specific scenario. This will make the data more manageable and reliable for experimentation. After cleaning the data, the primary challenge is to identify the model that is most suitable for our problem[25]. The next step is to find the best models for the Urdu next-word task. We found from the literature that deep learning has the potential to perform the best for this task. Therefore, we selected these models for our study.

Very few attempts have been made in the domain of Urdu next-word prediction, with only a limited number of studies addressing this particular task [26]. In these prior studies, researchers primarily employed classical machine learning classifiers such as Naive Bayes, KNN, stochastic models, Hidden Markov Model, and SVM to tackle this challenge [27, 46, 47]. However, a noteworthy observation is that none of these researchers explored the potential of deep learning approaches for next-word prediction in the context of Urdu.

To the best of our knowledge, this study represents the inaugural effort to explore and identify the most suitable method for Urdu next-word prediction. This article aims to bridge a gap in the existing research landscape by investigating the effectiveness of deep learning techniques in the domain of Urdu language processing, potentially paving the way for more advanced and accurate predictive models in this language.

The structure of the remainder of this paper is as follows: Section 2, provides an overview of the existing evaluation resources available for the next-word prediction task. Section 3 explains the evaluation methodology of the model on Ur-Mono corpus and the methods used for their evaluation. After this, Section 4 presents the results obtained from these techniques and provides a comprehensive analysis of the outcomes. Ultimately, in the end, conclude and wrap up the paper in Section 5.

2 Related Work

Previous literature discusses various efforts to develop standard methods for the next-word prediction task. The following sections contain a series of methods developed for the next-word prediction task.

2.1 Statistical Modeling

It is an NLP technique applied to a dataset in which statistical analysis is performed. This technique is also known as probabilistic modeling. It involves using probability to assess the presence of words in sentences and phrases, which are then transformed into a list of predictions. These predictions are made using Markov assumptions, constructed based on the n-gram model [26, 28].

2.2 N- Gram Language Model

It represents a more advanced form of statistical modeling that has overcome previous limitations. This probabilistic language model was initially proposed by Markov in 1913 [29, 48]. It considers the context of the preceding text, which greatly enhances word prediction efficiency and user convenience. Typically, the model considers the previous word as (n-1) when predicting the next-word, resulting in what is known as a bi-gram. If it utilizes two previous words to predict the next-word, it is referred to as a tri-gram, and so forth.

2.3 Skip-Gram Language models

The effectiveness of smoothing techniques in addressing the data sparsity problem has certain limitations. Alternative techniques have been introduced to overcome this issue, and SkipGram is one of them. Despite the shortcomings of statistical techniques, particularly the N-gram model, SkipGram incorporates long-distance relationships between words beyond n consecutive words without a significant increase in its parameter space. This feature enables it to skip tokens while predicting the context of a word [30, 49].

2.4 Hidden Markov Model

It is a statistical modeling technique introduced in 1966 and is primarily utilized for time series processing and sequences [31]. In 2020, the task of next-word prediction using the Hidden Markov Model (HMM) was employed in conjunction with a virtual keyboard. The results revealed a reduction in the total number of clicks by 26.3% to 61%, as users typed fewer words and their desired words were displayed to them with an accuracy of 90.7 [32].

2.5 Knowledge-based Modeling

One of the drawbacks of frameworks based on statistical modeling is that they often predict words that may be grammatically, semantically, or contextually inappropriate, which can make it challenging for the user to select the intended word. In response, data-driven modeling excludes certain inappropriate words, resulting in more unambiguous and relevant outcomes for the user. The linguistic knowledge used in the prediction system includes syntax, semantics, and logic [33].

2.6 Latent Semantic Analysis Indexing (LSAI)

It is a statistical method used to extract semantic knowledge from words in word prediction tasks. This method serves as a means of representing and extracting the meanings of words. It relies on singular value decomposition, which is a mathematical matrix decomposition technique. The meanings of words extracted by LSA are capable of simulating a variety of human cognitive phenomena. In the study of Spiccia et al [34]. 1040 question were used with one missing word with 5 possible words as a correct answer.

2.7 Neural Network

In the preceding paragraphs, we discussed how N-gram language models encounter a significant drawback in assigning probabilities of 0 to n-grams that are absent from the training corpus. This issue has been mitigated through the use of smoothing methods. However, there are other challenges, with the primary one being the curse of dimensionality. Neural networks were introduced to address this problem and provide a solution for modeling language in a continuous space [35].

2.8 Deep Learning Prediction Models

Deep learning is a vast field that addresses various problem types, and it plays a significant role in natural language processing. LSTM and Bi-LSTM are instrumental in enhancing system efficiency [36, 50]. In the following section, this article discusses a few deep learning technologies that have proven to be effective in next-word prediction and data processing.

In 2018, Barman [37] conducted research on a next-word prediction task in a low-resourced Assamese language. The primary focus of their work was predicting the next word to enhance text messaging, and they achieved an accuracy of 88.2% using an LSTM model. In another study, Demeke [38] addressed the next-word prediction task in the Amharic language in 2022.the dataset consist of 63300 Amharic sentences. They employed LSTM, GRU, Bi-LSTM, and BLSTM-GRU network models for their analysis. Applying these models to 63,300 sentences, they achieved accuracies of 75.02%, 73.5%, 76.1%, and 78.6%, respectively. The parameters used for these models were a batch size of 64, neurons set to 256, a learning rate of 0.001, and the Adam optimizer. At the end of this section Table 1. shows the summary of related work.

From all the discussions above, we have analyzed that for low-resourced languages like Urdu, it is imperative to implement deep learning models such as LSTM, GRU, and RNN to address the issue of next-word prediction.

Table 1. Table showing related work summary

Cite	Dataset	Model	Accuracy
[32]	40329 words	Hidden Markov	90.7
[34]	1040 questions	LSA	52.3
[37]	Dataset-1 1037 words Dataset-2 1076 words	LSTM	88.2
[38]	63300 Sentences	LSTM	78.6
[49]	170,220,000 Words	Skip-Gram	57.1
[50]	720-1260 Words	LSTM	75.0

3 Experimental Setup

This section delineates the dataset, the process of cleaning the dataset, the model description, and concludes with an overview of the evaluation measures.

3.1 Dataset

The experimental dataset underwent several iterations. Initially, 3,000 lines were extracted for the first iteration, followed by another extraction of 5,000 lines for subsequent model implementation. Upon finalizing the suitable dataset for experimentation, we partitioned each dataset into training and validation sets, employing an 80-20 split. This implies that 80% of the dataset was allocated for training purposes, while the remaining 20% served as the validation dataset. This meticulous process ensured a balanced division of data, allowing for comprehensive model training and effective evaluation through validation. The complete code and datasets are given on this link [1].

[1] https://drive.google.com/drive/folders/1Ok5ILjCHlluFT3KmIzBn1W1T71JSuElh?usp=sharing

3.2 Data Cleaning

The initial phase involves data cleansing, where a series of steps are undertaken to eliminate unwanted interference from the dataset: (1) Removing punctuation marks (2) Eliminating numerical values (for example 1,2, 3, ... or i, ii, iii, ... etc.). (3) Excluding stop words[2]. (4) Tokenization. The complete data preprocessing is explained in pictorial form in Figure 1. Subsequently, we utilize specific evaluation metrics to assess the effectiveness of the models in predicting the next word in Urdu. These metrics serve as quantitative measures to gauge the performance and accuracy of our predictive model. All of these metrics are derived from [39, 40].

3.3 Model Description

This study proposed a three-layered neural network model, described in Figure 2. The initial layer, referred to as the embedding layer, takes input comprising Urdu preprocessed text that may contain any number of words in a sentence. This layer transforms the textual information into Keras embeddings, generating semantically rich vectors characterized by reduced dimensions. This layer also determines the vocabulary size and the input length [3]. The embedding layer has two roles. The first role is to serve as the input layer, transforming words into vectors that represent their semantic meaning. This allows the model to process the input according to its intended behavior. The second role is responsible for encoding the contextual information of the words. The output of the embedding layer is then fed into an RNN layer, followed by two dense layers. The dense layers are fully connected to the neurons in the previous layer, aiding in decoding the model output.

The second layer applies the RNN, LSTM, and Bi-LSTM one by one with the same parameters set (See Sections 3.4 and 3.5 for parameter settings) on the semantically rich vector from the initial layer. This layer determines the output length that produces the best results [41, 51, 52]. A Recurrent Neural Network (RNN) is designed for sequential data processing and utilizes recurrent connections to allow information persistence over time, enabling the network to re-

[2] https://www.kaggle.com/datasets/rtatman/urdu-stopwords-list

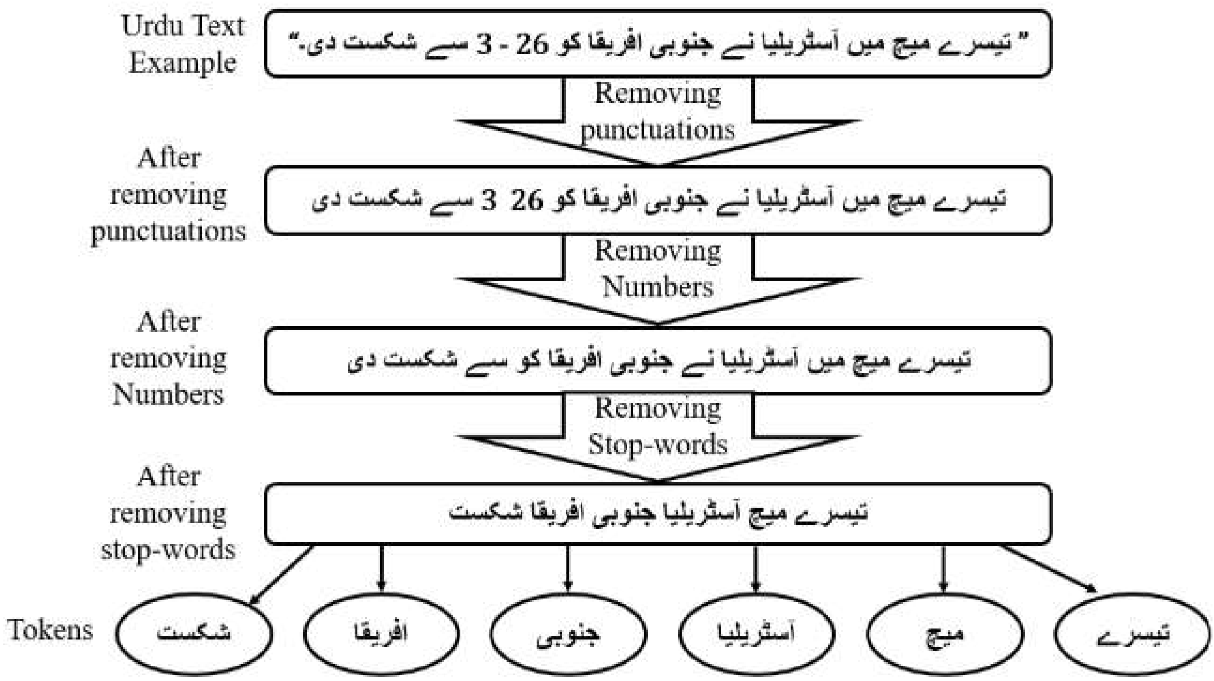

Figure 1. The data preprocessing phases for both experiments

member past inputs [42]. RNNs are characterized by loops within their structure, facilitating the incorporation of temporal dependencies. On the other hand, the Long Short-Term Memory (LSTM) neural network is a specialized recurrent neural network variant designed to address the vanishing gradient problem in sequential data processing [43, 53, 54]. LSTMs feature memory cells with gating mechanisms, allowing the network to selectively retain or forget information over long sequences. LSTMs mitigate the challenges faced by traditional recurrent networks, enabling improved modeling of sequential patterns and fostering long-range dependencies for more effective learning and information retention.

However, the Bidirectional Long Short-Term Memory (Bi-LSTM) Neural Network enhances sequential data processing by incorporating information from both past and future contexts [44]. It employs two LSTM layers: one processes data in a forward temporal direction, and the other in a backward direction. This bidirectional approach enables the network to capture dependencies from both preceding and succeeding elements in a sequence, enhancing its understanding of context and improving predictive accuracy. Their dual-directional architecture makes them effective in modeling and learning intricate sequential patterns.

The third layer is the dense layer, which is responsible for generating the output. In this layer, the Adam function is used to determine the next output based on the probabilities calculated by the previous layers. It identifies the output value with the highest probability and provides the optimal result. The output layer presents the generated output to the user. It determines whether the output meets the user's requirements. Ultimately, the decision of whether to accept the output rests with the user.

3.4 Experiment-1

In the first experiment we trained RNN, LSTM, and Bi-LSTM models on a dataset consisting of the first 3,000 lines of Ur-Mono [24] dataset, containing a total of 230,299 words. For this experiment, we set the hidden layer parameters to 256 neurons, used the Adam optimizer, employed a batch size of 64, and trained the models for 30 epochs. These parameters are consistent with those used in the next-word pre-

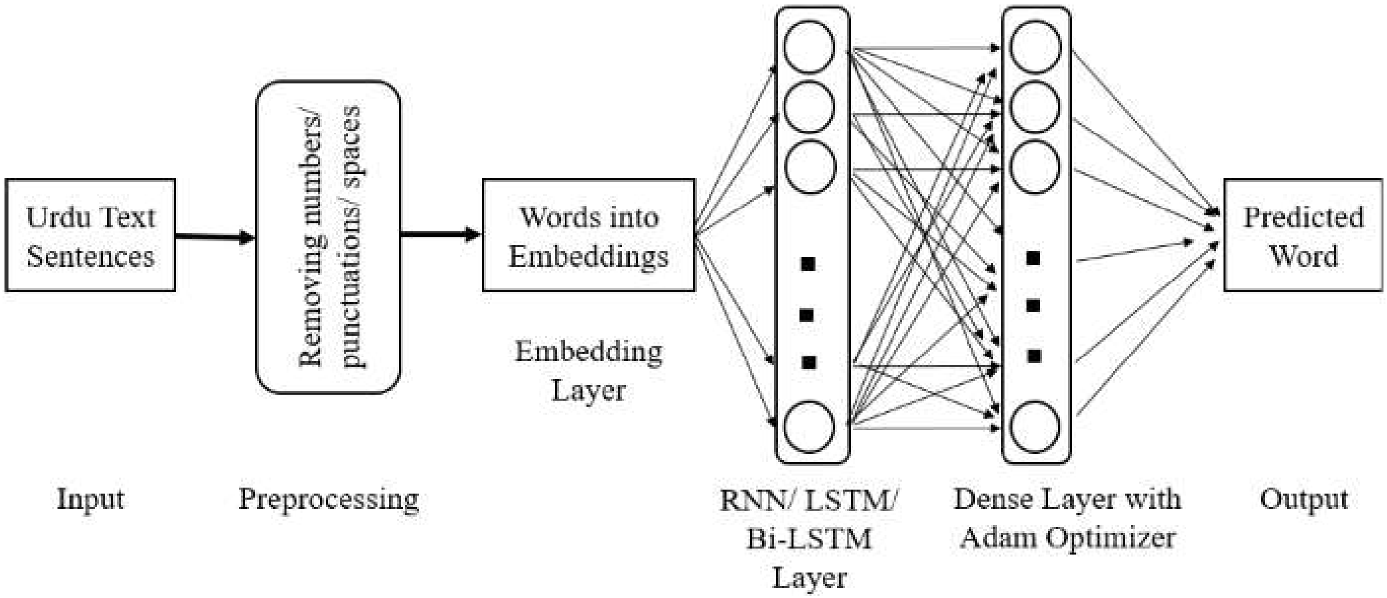

Figure 2. The proposed neural network model used for Urdu next-word prediction

Table 2. Parametric table for Experiment 1

Parameters	Values
Input Length	3
Number of Neurons	256
Optimizer	Adam
Batch Size	64
Learning Rate	0.001
Number of Epochs	30

Table 3. Parametric table for Experiment 2

Parameters	Values
Input Length	3
Number of Neurons	1000
Optimizer	Adam
Batch Size	64
Learning Rate	0.001
Number of Epochs	30
Activation	Relu, SoftMax

diction experiment described by [45]. Table 2 shows the parameter settings for experiment-1.

3.5 Experiment-2

In the second round of experimentation, we utilized 5,000 lines/sentences from the Ur-Mono [24] corpus, encompassing a total of 371,493 words. The total number of trainable parameters amounted to 1,425,802. We adjusted a few parameters from our previous iteration, where the total number of neurons was 1,000, the input length was three words, and the batch size was 64. All the parameters are presented in Table 3.

All other parameters remain the same, except for the number of neurons, as shown in the figure. This change is intended to improve the ability of the neurons to learn more effectively. We have used the embedding layer and dense layer as before.

3.6 Evaluation Measures

In the realm of machine learning, the accuracy metric serves as a prevalent indicator, providing a comprehensive assessment of the effectiveness of predictions in the word prediction model. In tandem with accuracy, we have also included the reporting of loss. Our comprehensive reporting encompasses both training and validation sets, presenting a holistic view of model performance. The mathematical formulations for both accuracy and loss are illustrated in Equations 1 and 2, offering a precise representation of these quantitative measures.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

$$Loss(MeanSquareError) = \frac{1}{N}\sum_{1}^{n}(Y - \hat{Y})^2 \quad (2)$$

A subset of the dataset reserved for validating the model's performance is known as the validation set. The validation loss, which is determined by adding the mistakes for every sample in the validation set, is comparable to the training loss. Since the model has never seen the validation data before, validation accuracy is often lower than training accuracy. But occasionally, you could come across a scenario where the validation accuracy exceeds the training accuracy.

4 Results and Discussion

Tables 4 and 5 indicate the results obtained using three algorithms, i.e., RNN, LSTM, and Bi-LSTM, on two datasets of 3,000 and 5,000 instances, respectively. Based on the findings of Experiment 1 (using 3K sentences), it can be inferred that RNN outperformed other models on the smaller dataset, while LSTM exhibited the lowest accuracy. Table 3 elucidates the results obtained in Experiment 1. For instance, the RNN model achieved a peak accuracy of 87%, a training loss of 0.443, a validation accuracy of 0.0595, and a validation loss of 20.96 on the distinct validation dataset. Similar metrics are presented for the Bi-LSTM model, which achieved an accuracy of 81%, a loss of 0.67, a validation accuracy of 0.637, and a validation loss of 27.25 on the separate validation dataset. In the third position, LSTM attained an accuracy of 79% with a loss of 0.76. Experiment 1 recorded a validation accuracy of 0.622 and a validation loss of 25.08 for the LSTM model.

In Experiment 2 (using 5K sentences), the Bi-LSTM model demonstrated superior performance compared to the other two models. This improvement can be attributed to the increased number of neurons in the second experiment, leading to a larger pool of trainable parameters. Consequently, the model was able to capture patterns more effectively. Table 5 explains the results of the 2nd experiment. The Bi-LSTM model outperforms both RNN and LSTM with

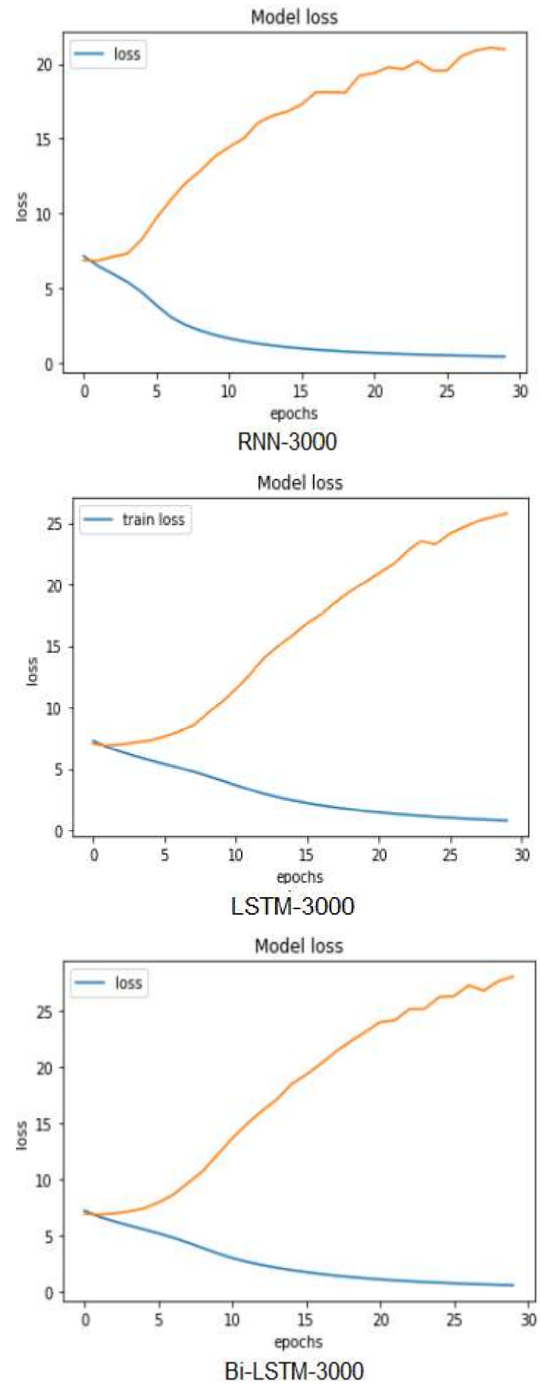

Figure 3. Experimental loss function graphs of RNN, LSTM, and Bi-LSTM for 3k datasets.

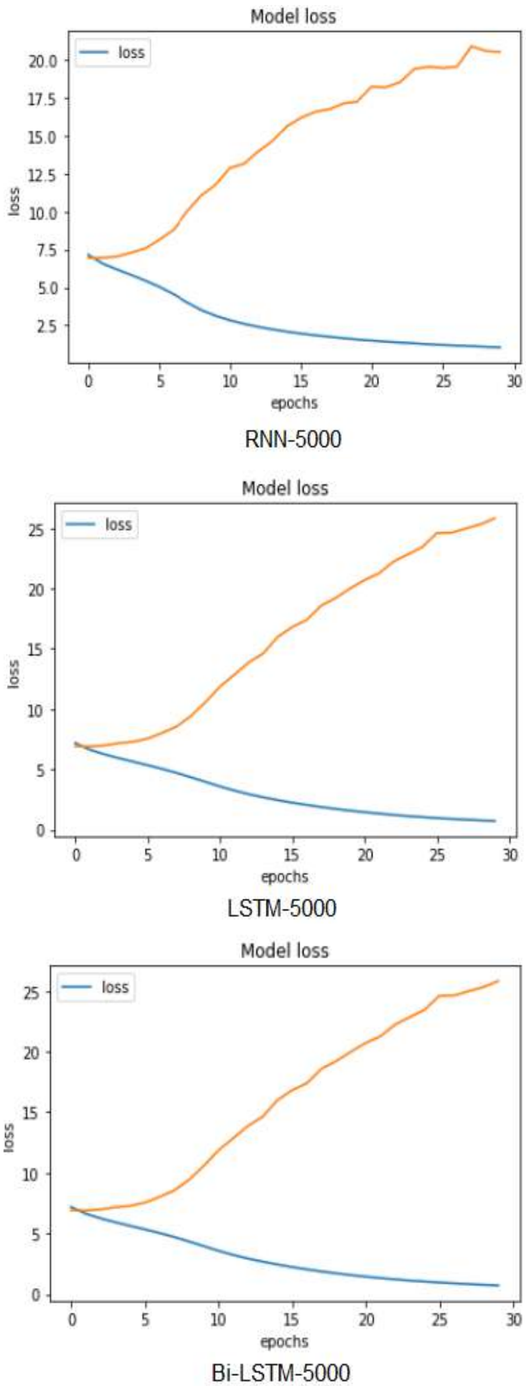

Figure 4. Experimental loss function graphs of RNN, LSTM, and Bi-LSTM for 5k datasets.

Table 4. Experimental results of RNN, LSTM, and Bi-LSTM models using 3k instances

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
RNN	87%	0.44	0.059	20.96
LSTM	79%	0.76	0.062	25.08
Bi-LSTM	81%	0.67	0.637	27.25

Table 5. Experimental results of RNN, LSTM, and Bi-LSTM models using 5k instances

Model	Accuracy	Loss	Validation Accuracy	Validation Loss
RNN	72%	1.05	0.0596	20.50
LSTM	80%	0.71	0.0675	25.82
Bi-LSTM	84%	0.58	0.0742	26.82

an accuracy of 84%, the lowest training loss of 0.588, a validation accuracy of 7.42%, and a validation loss of 26.82. In the second, the LSTM model performs with an accuracy of 80%, a lower training loss of 0.712, a validation accuracy of 6.75%, and a validation loss of 25.82. The RNN model has the lowest accuracy of 72% on the training data, a training loss of 1.05, a validation accuracy of 5.96%, and a validation loss of 20.50. Figure 3 and fig 4 shows the loss function of six experimental results. These experiments include three models: RNN, LSTM, and Bi-LSTM, using a dataset of 3,000 Urdu sentences shows in Figure3.

The second set of experiments includes three models: RNN, LSTM, and Bi-LSTM, using a dataset of 5,000 Urdu sentences shows in Figure 4. The loss function represents the difference between the actual and predicted outputs. A higher loss value indicates less validity of the model. From the given figure, it can be observed that all methods exhibit an increase in loss values as epochs increase. This is the reason we use 30 epochs for all experiments. In conclusion of these experiments, we can say that the RNN_3000 and RNN_5000 techniques are the most error-free models on our given datasets.

5 Conclusion

We used the models RNN, LSTM, and Bi-LSTM for our experiments with two subsets of the corpus of Urdu monolingual data, containing 3000 and 5000 Urdu sentences. In the first experiment involving 3,000 sentences, RNN outperformed LSTM and Bi-LSTM in terms of accuracy, achieving a maximum of 87%, while LSTM scored 79%, and Bi-LSTM reached 81%. Transitioning to the second experiment with 5,000 sentences, Bi-LSTM excelled due to its bidirectional nature, resulting in satisfactory results and a maximum accuracy of 84%. Our experiments demonstrated that adjusting model parameters can significantly impact improvements and deficiencies within the models, emphasizing the pivotal role of parameters in a model's success, along with the importance of high-quality data for successful experimentation. Further studies will focus on expanding the dataset and incorporating other regional or local languages.

Author Contributions

Haroon Ahmad: Model Development Data Collection and Cleaning. **Ali Saeed**: Conceptualization, Methodology, Supervision **M Usman Bhatti**: Writing-Original draft preparation, Experiments and Implementation **Naveed Hussain**: Reviewing and Suggestion. **Muhammad Farhat Ullah**: Model Development and Experiments. **Mehmood Anwar**: Reviewing and Suggestion.

References

[1] Rianti, A., Widodo, S., Ayuningtyas, A., Hermawan, B.: Next word prediction using lstm. Journal of Information Technology and Its Utilization 5 (2022)

[2] Saeed, M., Nawaz, A., Ahsan, K., Jabeen, S., Islam, K., Hassan, M., Siddiqui, F.A.: Effective word prediction in urdu language using stochastic model 2, 38–46 (2019)

[3] Farhat Ullah, M., Saeed, A., Hussain, N.: Comparison of pre-trained vs custom-trained word embedding models for word sense disambiguation. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 12(1), 31084 (2023)

[4] Kour, H., Gondhi, N.K.: Machine learning approaches for nastaliq style urdu handwritten recognition: A survey. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 50–54 (2020). IEEE

[5] Khan, L., Amjad, A., Ashraf, N., Chang, H.-T., Gelbukh, A.: Urdu sentiment analysis with deep learning methods. IEEE Access 9, 97803–97812 (2021)

[6] Daud, A., Khan, W., Che, D.: Urdu language processing: a survey. Artificial Intelligence Review 47, 279–311 (2017)

[7] Gong, Y., Hua, L., Wang, S.: Leveraging user's performance in reporting patient safety events by utilizing text prediction in narrative data entry. Computer methods and programs in biomedicine 131, 181–189 (2016)

[8] rnka, K., McCaw, J., Yarrington, D., McCoy, K.F., Pennington, C.: User inter- action with word prediction: The effects of prediction quality. ACM Transactions on Accessible Computing (TACCESS) 1(3), 1–34 (2009)

[9] larcon, R., Moreno, L., Mart´ınez, P.: Lexical simplification system to improve web accessibility. IEEE Access 9, 58755–58767 (2021)

[10] Alkhatib, M., Monem, A.A., Shaalan, K.: Deep learning for arabic error detec- tion and correction. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 19(5), 1–13 (2020)

[11] ssaneo, M.F., Ripoll´es, P., Orpella, J., Lin, W.M., Diego-Balaguer, R., Poeppel, D.: Spontaneous synchronization to speech reveals neural mechanisms facilitating language learning. Nature neuroscience 22(4), 627–632 (2019)

[12] an Laere, O., Strobbe, M., Leroux, P., Dhoedt, B., De Turck, F., Demeester, P.: Enabling platform for mobile content generation based on 2d barcodes. In: International Conference on Internet Computing, pp. 209–214 (2008)

[13] Babar, S., Patil, P.D.: Improving performance of text summarization. Procedia Computer Science 46, 354–363 (2015)

[14] Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. arXiv preprint arXiv:1606.02892 (2016)

[15] Ghosh, S., Dutta, M., Das, T.: Indian legal text summarization: A text normalization-based approach. In: 2022

IEEE 19th India Council International Conference (INDICON), pp. 1–4 (2022). IEEE

[16] Kalamkar, P., Tiwari, A., Agarwal, A., Karn, S., Gupta, S., Raghavan, V., Modi, A.: Corpus for automatic structuring of legal documents. arXiv preprint arXiv:2201.13125 (2022)

[17] Sharma, R., Goel, N., Aggarwal, N., Kaur, P., Prakash, C.: Next word prediction in hindi using deep learning techniques. In: 2019 International Conference on Data Science and Engineering (ICDSE), pp. 55–60 (2019). IEEE

[18] Chambers, J.M., Hastie, T.J.: Statistical models. In: Statistical Models in S, pp. 13–44. Routledge, ??? (2017)

[19] Forsyth, D., Forsyth, D.: Hidden markov models. Applied Machine Learning, 305–332 (2019)

[20] Morsy, S., Karypis, G.: Cumulative knowledge-based regression models for next- term grade prediction. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 552–560 (2017). SIAM

[21] oam, M., Thakur, S.: Next word prediction using deep learning: A comparative study. In: 2022 12th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 653–658 (2022). IEEE

[22] Atccili, A., Ozkaraca, O., Sariman, G., Patrut, B.: Next word prediction with deep learning models. In: The International Conference on Artificial Intelligence and Applied Mathematics in Engineering, pp. 523–531 (2021). Springer

[23] berhard, D.M., Simons, G.F., Fennig, C.D.: Ethnologue: Languages of the world. 23rd edn. dallas: Sil international. Online: https://www. ethnologue. com (2020)

[24] Jawaid, B., Kamran, A., Bojar, O.: A tagged corpus and a tagger for Urdu. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp. 2938–2943. European Language Resources Association (ELRA), Reykjavik, Iceland (2014).

[25] Saeed, A., Ullah, M. F., Sauood, M., Ali, S. N., Hussain, N. (2023). Prediction of Dengue Cases and Deaths using Machine Learning Algorithm. Pakistan Journal of Scientific Research, 3(2), 210-216.

[26] Siddiqui, M.F., Hassan, M.: Effective word prediction in urdu language using stochastic model. Sukkur IBA Journal of Computing and Mathematical Sciences 2(2), 38–46 (2018)

[27] Saeed, A., Nawab, R. M. A., Stevenson, M. (2021). Investigating the feasibility of deep learning methods for urdu word sense disambiguation. Transactions on Asian and Low-Resource Language Information Processing, 21(2), 1-16.(2021)

[28] Avasthi, S., Chauhan, R., Acharjya, D.P.: Processing large text corpus using n-gram language modeling and smoothing. In: Proceedings of the Second International Conference on Information Management and Machine Intelligence: ICIMMI 2020, pp. 21–32 (2021). Springer

[29] Naik, D., Naik, I., Naik, N. (2024, July). Large data begets large data: studying large language models (LLMs) and its history, types, working, benefits and limitations. In The International Conference on Computing, Communication, Cybersecurity AI (pp. 293-314). Cham: Springer Nature Switzerland.

[30] ggarwal, C.C., et al.: Neural networks and deep learning. Springer 10(978), 3 (2018)

[31] Ruffolo, J. A., Madani, A. (2024). Designing proteins with language models. nature biotechnology, 42(2), 200-202.

[32] Jordan, M., Neto, G.N.N., Brito Jr, A., Nohama, P.: Virtual keyboard with the prediction of words for children with cerebral palsy. Computer Methods and Programs in Biomedicine 192, 105402 (2020)

[33] Yu, J., Weng, J., Wang, T., Lin, P., Sun, Y., Chai, J. (2025). How do differences in airline passengers' satisfaction with connectivity modes affect last-mile travel choices? A SALC modeling based on RRM. Transportation Research Part A: Policy and Practice, 192, 104374.

[34] Spiccia, C., Augello, A., Pilato, G., Vassallo, G.: A word prediction methodology for automatic sentence completion. In: Proceedings of the 2015 IEEE 9th Inter- national Conference on Semantic Computing (IEEE ICSC 2015), pp. 240–243 (2015).

[35] jing, K., Xu, J.: A Survey on Neural Network Language Models (2019)

[36] Song, Z., Liu, L., Song, W., Zhao, X., Du, C.: A neural network model for Chinese sentence generation with keyword. In: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 334–337 (2019).

[37] Barman, P.P., Boruah, A.: A rnn based approach for next word prediction in assamese phonetic transcription. Procedia Computer Science 143, 117–123 (2018).8th International Conference on Advances in Computing Communications (ICACC-2018)

[38] Endalie, D., Haile, G., Abebe, W.: Bi-directional long short term memory-gated recurrent unit model for amharic next word prediction. PLOS ONE 17, 0273156(2022)

[39] Yacouby, R., Axman, D.: Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In: Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, pp. 79–91 (2020)

[40] Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f- score, with implication for evaluation. In: European Conference on Information Retrieval, pp. 345–359 (2005). Springer

[41] Rakib, O.F., Akter, S., Khan, M.A., Das, A.K., Habibullah, K.M.: Bangla word prediction and sentence completion using gru: An extended version of rnn on n- gram language model. 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 1–6 (2019)

[42] Elman, J.L.: Finding structure in time. Cognitive Science 14(2), 179–211 (1990)

[43] ochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9(8), 1735–1780 (1997)

[44] Zhang, S., Zheng, D., Hu, X., Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Zhao, H. (ed.) Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, pp. 73–78 (2015).

[45] Endalie, D., Haile, G., Taye, W.: Bi-directional long short term memory-gated recurrent unit model for amharic next word prediction. PLoS ONE 17 (2022)

[46] Ullah, Muhammad Farhat, et al. BERT model for Roman Urdu fake review identification. (2023).

[47] Bhatti, Muhammad Usman, et al. "Optimizing Brain Tumor Prediction: A Comparative Study Of Machine Learning Algorithms." VFAST Transactions on Software Engineering 12.4 (2024): 209-219.(2024)

[48] Saeed, A., Nawab, R. M. A., Stevenson, M., and Rayson, P. (2019). A sense annotated corpus for all-words Urdu word sense disambiguation. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 18(4), 1-14.(2019)

[49] Li, Peijun. (2023). The study for word prediction based on Skip-gram and CBOW model. Theoretical and Natural Science. 18. 210-215. 10.54254/2753-8818/18/20230392. (2023)

[50] Rianti, Afika, Widodo, Suprih, Ayuningtyas, Atikah and Hermawan, Bima. (2022). NEXT WORD PREDICTION USING LSTM. Journal of Information Technology and Its Utilization. 5. 10.56873/jitu.5.1.4748. (2022)

[51] Saeed, A., Nawab, R. M. A., Stevenson, M., Rayson, P. (2019). A word sense disambiguation corpus for Urdu. Language Resources and Evaluation, 53, 397-418.(2019)

[52] Muneer, I., Fatima, G., Khan, M. S., Adeel Nawab, R. M., Saeed, A. (2023). Developing a large benchmark corpus for urdu semantic word similarity. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(3), 1-19.(2023)

[53] Fatima, G., Nawab, R. M. A., Khan, M. S., Saeed, A. (2021). Developing a Cross-lingual Semantic Word Similarity Corpus for English–Urdu Language Pair. Transactions on Asian and Low-Resource Language Information Processing, 21(2), 1-16.(2021)

[54] Vardag, M. H. K., Saeed, A., Hayat, U., Ullah, M. F., Hussain, N. (2022). Contextual Urdu text emotion detection corpus and experiments using deep learning approaches. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal,11(4),489-505.