RESEARCH OF NEXT WORD PREDICTION MODEL

Dr.Vaka Murraali Mohan[1]., Abhinay Reddy M[2].,Rahul Angu[3].,Ravishankar Barahate[4].,Laxman Kumar Poloju[5].

[1] Principal & Professor of CSE, [2,3,4,5]Students B.Tech-CSE(N/W),
Malla Reddy Institute of Technology and Science.,Maisammaguda.,Medchal.,Ts,India

[1]vakamuralimohan@gmail.com, [2]abhinayreddykonkala@gmail.com,
[3]angurahul7@gmail.com, [4]ravishankarbarahate@gmail.com,
[5]polojulaxman93@gmail.com

ABSTRACT:

Word prediction apps that make typing simpler are made using a variety of methods. By suggesting phrases the user may want to type in a text field, these technologies also make typing on a mobile device easier. Additionally, it makes writing more fluent, which enables kids to produce greater writing abilities. Additionally, it facilitates free text typing on the machine. It also aids in the formation of well-structured language's sequence. It is used in the creation of highly regarded programs like Grammarly and others. In addition to being employed when the user inputs the needed word's letter, the system also shows a list of the most likely words to fit the position. Additionally, it can anticipate words in several languages, like Hindi, Spanish, etc. Predicting a sentence's next word is the primary goal. In order to anticipate the following word more accurately, this research uses recurrent neural networks, convolution neural networks, N-gram modeling, and a few other deep learning approaches. Results, analysis, and approaches are also included in this paper. We can simply guess the following word by going over all of them.

Keywords: Natural language processing, N-gram model, recurrent neural network, next word prediction, and long short-term memory (LSTM).

Introduction

In the current world, a variety of technologies have been developed in the field of machine learning and deep learning. Every day, we input information into a variety of devices and send it to the other end; nevertheless, repeatedly entering the same material is tiresome. Therefore, by guiding users through the process, future word prediction helps users compose phrases more rapidly and effectively. Textual information is typically conveyed more rapidly in a given period of time by anticipating the following word. It may be used to other languages as well. This makes it easier for the user to save keystrokes. The ability to predict the next few words in a given word sequence was added to this. Word prediction is a fundamental part of natural language generation. Correcting grammatical faults and word placement within a certain process might be beneficial. For beginning learners like students or unskilled researchers, word prediction may increase typing speed and decrease spelling errors. Probabilities are provided by language models for a phrase, a string of words, or the probability of a word given its preceding sequence of words. Correcting grammatical faults and word placement within a certain process might be beneficial. Spell checking, speech recognition, machine learning, and other fields may all benefit from these models. This study involves the use of convolutional neural networks, deep learning, machine learning, N-gram modeling, recurrent neural networks, and natural language processing.

*Corresponding Author

LITERATURE SURVEY

[1] Saeed, S. A., Hamarashid, H. K., and Rashid, T. A. (2021). For Kurdish Sorani and Kurmanji, the N-gram model predicts the next word. 33(9): 4547–4566 in Neural Computing and Applications.
The primary algorithm and its accuracy results for Next Word Prediction are provided by the author.The author of this work suggested the N-gram language model.Language models offer probabilities to a string of words, a phrase, or the likelihood of the next word given the set of words that came before it.An effort is made to construct language models by distributing actual probabilities in the Stupid Back Off method.Numerous domains, like spell checking, voice recognition, machine learning, etc., may benefit from these approaches.N-gram modeling is used to provide precise text suggestions. To cut down on time, the N-gram model has been used to next word prediction. Accuracy of the model is 96.3%.
[2] Guo, K.; Yang, J.; Wang, H. (2020). Word Prediction Model for Natural Language Using Residual Network and Multi-Window Convolution. Eighth IEEE Access, 188036–188043. For word prediction in natural language, multi-window convolution and residual-connected minimum gated unit (MGU) network are used. In order to extract the local feature information of varying graininess between word sequences, convolution kernels of various sizes are used.CNN uses various convolution kernel window widths to extract sequence feature information at varying granularities. The suggested MCNN-ReMGU considerably outperforms the conventional approaches in the word prediction task, according to the overall experimental findings. The text input rate at work has been increased via the usage of the N-gram language model. It turns out that there is a 73.53% decrease in keystrokes and a 33.36% reduction in typing time. The system's architecture cut down on the amount of time needed to type free text, which might be a way to enhance EHR documentation.
[3] Singh, A., and Stremmel, J. (2021, April). Federated text models are pre-trained for next word prediction. In Conference on the Future of Information and Communication (pp. 477–488). Springer, Cham. The LSTM-RNN language model for the Next Word Prediction was suggested by the author in this study. To average the model parameters, the Federated Averaging Algorithm is used. after the gradient's application to every database model. Without any retraining, federated training on Stack Overflow produces the two learning curves with the lowest levels of validation and train accuracy, respectively. When using federated training and facing model size constraints, the dimensionality reduction strategy might be helpful. This work has attained an accuracy level of around 22.1% and 22.2%.Federated learning, which sends aggregate parameters from local models to the cloud instead of the actual data, is a decentralized method of training models on dispersed devices by summarizing local changes.
[4] Saeed, S. A., Hamarashid, H. K., and Rashid, T. A. (2022). An extensive analysis and assessment of entertainment and text prediction technologies. Soft Computing, 1–22. In order to anticipate the following word, this study combined memory-based learning with long short-term memory and recurrent neural networks. Understanding and analyzing the challenges of automatically producing and understanding human languages is the aim of natural language processing (NLP).When there is little distance between the required location and the relevant information, RNN may learn to use the prior knowledge. Although the LSTM has a lot of parameters, it overcomes the memory issue. The accuracy of this model is as high as 44.2%.using hybrid methods, such as the Latent Semantic Analysis (LSA) model and Naive Bayes.In NLP, the Naive Bayes probabilistic approach is used, similar to N-gram. The accuracy of this model's output is 88.2%.
[5] In 2020, Barman, P. P., and Boruah, A. An RNN-based method for Assamese phonetic transcription next word prediction. Computer Science Procedia, 143, 117–123. One of the most talked-about topics in the field of natural language processing research nowadays is next word prediction. A language model based on recurrent neural networks is used to predict words in sequential data more accurately. To create intricate, long-range organized sequences, use LSTM.The Next Word employed LSTM with a phonetically transcripted accuracy of 72.10% and an Assamese text accuracy of 88.02%. Language of Assam

METHODOLOGY/RECENT TECHNOLOGY

Incompetent Backoff Algorithm: N-gram modeling approaches are used. Additionally, it is broken down into smaller approaches such as the Bigram, Trigram, and Unigram language models.

The likelihood of guessing the current word from a collection of hole words in the Unigram language model (1.1).

$$P(W_i)=C(W_i)/C(W)(1)$$

1.2 Bigram language model: The likelihood of guessing a word from the set of hole words when combined with a prior word.

$$P(W_i/W_{i-1}) =P(W_iW_{i-1})/P(W_{i-1})(2)$$

1.3 Trigram language model: The likelihood of correctly guessing the following two words from the set of hole words when combined with the prior word.

$$P(W_i/ W_{i-2} , W_{i-1}) = P(W_{i-2} , W_{i-1} , W_i)/P(W_{i-2} , W_{i-1})(3)$$

The effort to construct language models by allocating genuine probabilities is halted by the Stupid Back Off method. We will revert to lower-order N-gram if the higher-order N-gram is counted as zero. Consequently, the Stupid Back Off method does not provide a probability distribution.

$$S(W_i/W_{i-k+1}^{i-1})=\{count(W_{i-k+1}^{i})/count(W_{i-k+1}^{i})if$$

$$count(W_{i-k+1}^{i})=\{\lambda S(W_i/W_{i-k+1}^{i-1}) \quad (4)$$

where the Back Off variable is designed to depend on k, and λ is the weight of Back Off. Multiple situations are utilized simultaneously by the Stupid Back Off algorithm. This entails using a lot of words to find likelihood. For example, the system falls back to the bigram or unigram if there is insufficient evidence to support the trigram. Accuracy of the model is 96.3%.

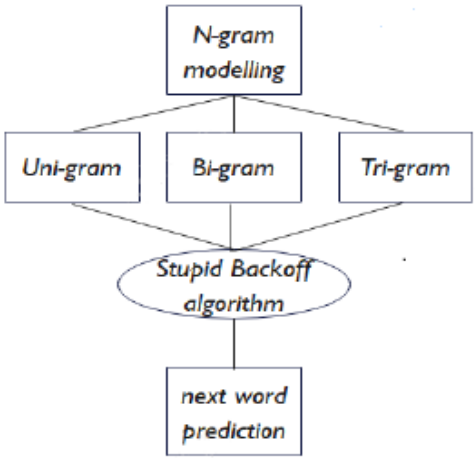

Fig-1: N-gram modeling

Encoding Words for Federated Training: Retraining Word Embeddings is implemented in this manner as well. The post-processing method is employed in this procedure to gather various details on word relationships. The

Dimensionality Reduction algorithm is used after this post-processing procedure, improving the experimental runs. (scaled to 300 and 100 degrees)This test run has produced Fast Text.

Post-Processing Algorithm PPA(X, D) is the first algorithm. Data: Threshold Parameter D, Word Embedding Matrix X

Outcome: Word Embedding Matrix X After Processing /* Subtract Mean Embedding

$$1.\ X = X - x$$

/* Calculate the Principal Component Analysis (PCA) components.

$$2.u_i = PCA(X) \text{ where } i = 1, 2,\ldots,D$$

/* Remove Top-D Components

3. for all v in X do

4. $v = v - D\ i=1(uT_i \cdot v)u_i$

5. end for

Algorithm 2: Reduction of Dimensionality PP PCA PP(X, N, D) algorithm

Data: Threshold parameter D, New Dimension N, Word Embedding Matrix X Word Embedding Matrix of Reduced Dimension N: X is the outcome.

/* Apply Algorithm 1 (PPA)*/

1. X = PPA(X, D)

/* Transform X Using PCA to N Dimensions*/

2. X = PCA Transform(X)

/* Apply Algorithm 1 (PPA) */

3. X = PPA(X, D)

When using federated training and facing model size constraints, the dimensionality reduction strategy might be helpful. the same degree of accuracy as the 22.1% and 22.2% achieved in the article.

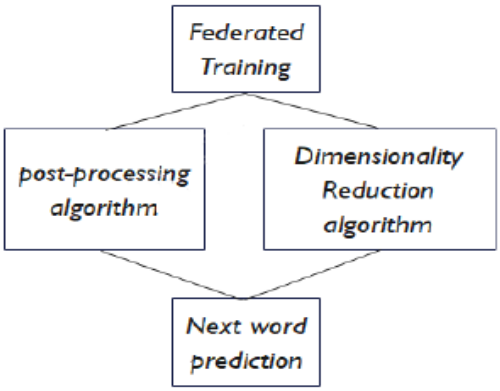

Fig-2: Federated training

RESULTS & DISCUSSION

The proposed next word prediction approach uses weighted values and the Stupid Back Off algorithm to generate a list of possible next words. Depending on the N-gram frequencies, the highest frequency for the top five words either increases or matches. It then combines word probability sequences to determine which one has the best chance of being guessed correctly. It is clear from the N-gram model that performance increases with N. Thus, mapping the accuracy of the used language model provides a simple way to assess it. The result represents the percentage of terms that were correctly predicted. Accuracy is equivalent to a correct prediction or entire forecast. The N-gram model, which is simple to use and just considers word frequency rather than complex probabilities that affect system performance, was shown to be an effective way to measure system effectiveness.

TABLE-1:N-gram modeling

N-grams	Accuracy (in %)
One-gram	25.4
Two-gram	58.6
Three-gram	72.38
Four-gram	88.24
Five-gram	96.3

Approaching the second technique, it is discovered that the use of retrained word embedding might potentially reduce the number of federated training cycles needed to get a satisfactory level of model accuracy. Despite using federated fine-tuning in conjunction with central pretraining, we are unable to outperform federated training. When compared to federated training, it cannot manage the enormous data collection. The accuracy level of the document is between 22.1% and 22.2%.On the Amharic phrase dataset, the third technique yields an overall estimate using the LSTM, GRU, and BLSTM models. As the RNN model's accuracy is 2.5% higher. To reduce execution time, the suggested predictive network model combines BLSTM and GRU.

TABLE-2:BILSTM-GRU MODELLING

Algorithm	LSTM	GRU	BLSTM	BLSTM-GRU
Accuracy	75.02%	73.5%	76.1%	78.6%

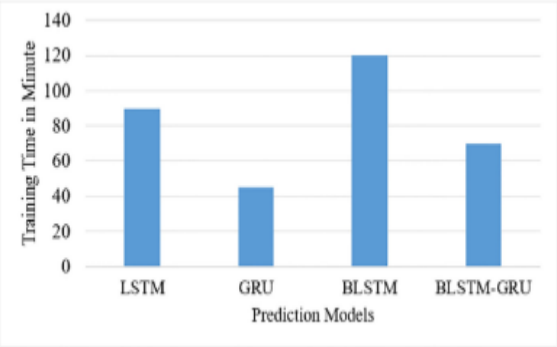

Fig-4:Time graph for BLSTM - GRU

CONCLUSION

According to the research, the first methodology—which relies on probability and weights of the supplied probability in line with n-gram modeling—achieved more accuracy. The N-gram is a prospective predictive model that calculates a word's likelihood of occurring. The text input rate at work has been increased via the usage of the N-gram language model. The second process is less accurate but works longer with huge data sets. It was shown that increasing the number of layers decreased the accuracy of both training and validation. Without any pretraining, federated training on Stack Overflow produces the two learning curves with the lowest levels of validation and train accuracy, respectively. Natural language generation (NLG) is a methodical and important way to generate meaningful text that can be comprehended by people. When using federated training and facing model size constraints, the dimensionality reduction strategy might be helpful. a Recurrent Neural Network that predicts the next word in a text sequence using a combination of LSTM and RNN techniques. The third way has the greatest accuracy while requiring least running time. The LSTM model only learns a sentence's word order in one direction—either forward or backward. Taking into account, the first is a good choice for word prediction after that.

REFERENCES

[*1] Hamarashid, H. K., Saeed, S. A., & Rashid, T. A. (2021). Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji. Neural Computing and Applications, 33(9), 4547-4566.*
[2] Stremmel, J., & Singh, A. (2021, April). Pretraining federated text models for next word prediction. In Future of Information and Communication Conference (pp. 477-488). Springer, Cham.

[3] Hamarashid, H. K., Saeed, S. A., & Rashid, T. A. (2022). A comprehensive review and evaluation on text predictive and entertainment systems. Soft Computing, 1-22.

[4] Barman, P. P., & Boruah, A. (2020). A RNN based Approach for next word prediction in Assamese Phonetic Transcription. Procedia computer science, 143, 117-123.

[5] Rakib, O. F., Akter, S., Khan, M. A., Das, A. K., & Habibullah, K. M. (2019, December). Bangla word prediction and sentence completion using GRU: an extended version of RNN on N-gram language model. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.

[6] Ambulgekar, S., Malewadikar, S., Garande, R., & Joshi, B. (2021). Next Words Prediction Using Recurrent NeuralNetworks. In ITM Web of Conferences (Vol. 40, p. 03034). EDP Sciences.

[7] Naulla, N. T. K., & Fernando, T. G. I. (2022, February). Predicting the Next Word of a Sinhala Word Series Using Recurrent Neural Networks. In 2022 2nd International Conference on Advanced Research in Computing (ICARC) (pp. 13-18). IEEE.

[8] Shakhovska, K., Dumyn, I., Kryvinska, N., & Kagita, M. K. (2021). An Approach for a Next-Word Prediction for Ukrainian Language. Wireless Communications and Mobile Computing, 2021.

[9] Stremmel, J., & Singh, A. (2021, April). Pretraining federated text models for next word prediction. In Future of Information and Communication Conference (pp. 477-488). Springer, Cham.

[10] Yazdani, A., Safdari, R., Golkar, A., & R Niakan Kalhori, S. (2019). Words prediction based on N-gram model for free-text entry in electronic health records. Health information science and systems, 7(1), 1-7.