Long Short Term Memory model-based automatic next word generation for text-based applications In contrast to the N-gram model

P. Niharika [1], S. John Justin Thangaraj[2*]

[1]Research Scholar, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India. Pincode: 602105.

[2*]Project Guide, Corresponding Author, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India. Pincode: 602105.

ABSTRACT

Aim: The proposed study aim is to implement automatic next word generation for text based apps using novel long short term memory model and improving the accuracy using recurrent neural network in comparison with n-gram approach. **Materials and Methods:** Novel long short term memory is applied on data i.e. text file that consists of a sequence of words. Novel long short term memory model for suggestion accuracy of the next word which compares a n-gram model. LSTM has been proposed and developed in this study. The sample size was measured as 5046 per group with the G power value 0.8. **Results:** The accuracy was maximum in predicting the next word for text based using long short term memory model 48% with minimum mean error when compared with n-gram model 39% for the same dataset with the p value of 0.02 ($p<0.05$). **Conclusion**: The study proves that novel long short term memory exhibits better accuracy than n-gram in suggesting the next word for text based apps.

Keywords: Novel LSTM, Memory model, N-gram, Next word generation, Recurrent neural network, Language model.

INTRODUCTION

Natural language processing has been an area of research and used widely in different applications. People regularly text each other and find that the Long term short memory model is very good for analyzing the sequences of values and predicting the next one. The novel LSTM could be a good choice for predicting the very next point of the time series (sequence). LSTM recurrent neural networks will be used to guess the next word of a given sequence of words. The N-gram model can be trained by counting and normalizing. N Gram model trained by counting how often word sequences occur in corpus text and estimating the probabilities (van Gerven and Bohte 2018). The N-gram model has both limitations and improvements are often made using smoothing, interpolation and backoff. A model that simply depends on a word without looking at the previous words is called unigram. If a model considers only the previous word to predict the current word then it's called bigram (van Gerven and Bohte 2018). If two previous words are considered,then it's a trigram model for the next word generation. This model helps to predict the next word for real time applications like whatsapp, text based applications. The main application areas where natural language processing is employed are

speech recognition, language translation, question answering, language generation and summarization (Murray and Chiang 2015). All the applications are used in the health sector, hiring and recruitment process, advertising ,customer services etc

Reading the history from left to right, LSTM cells generated the most likely word, which was then fed back in as new input. It would be simple to derive the weighted dimensions in novel LSTM. Because the output of one layer should have the same dimension as the layer's needed input dimension. As a result, the output and input are always the same dimensions. This comes in useful when working with multiple layers (Pietro Giovanni Antiga, Stevens, and Viehmann 2020). The ngram model works well with a small or medium-sized training set, but not so much with a bigger one. Generally, the training and testing data sets are split 80-20, i.e. , 80 percent for training and 20% for testing. The same error on the test set arose after a stationary point, despite the fact that training errors on a longer context were smaller. Because of its exact string matching characteristics, N-gram performs exceptionally well on the training set (Brownlee 2017; van Gerven and Bohte 2018). It was appropriate for generating outputs based on observed data, such as similar check, text categorization, and classification. To be clear, the n-gram rarely works with unknown word combinations. The anticipated word is only reassuring due to the impracticality of long-term dependency in n-gram. The neural language model (NLM) is another name for the language model (Chen, Zhong, and Zhu 2022). There are two types of NLM : feed-forward recurrent neural network-based language model,which was designed to cope with data sparsity and recurrent neural system-based language model, which was designed to deal with limited settings. Recurrent neural system primarily based on recently achieved progressive performance (van Gerven and Bohte 2018). The first neural language model is intended to address the two major difficulties with n-gram models as previously mentioned (Irene et al. 2021). Ulterior works have gone on to have some expertise in sub-word modelling and corpus-level modeling,which is supported by a recurrent neural system and its alternative,the novel long short term memory recurrent neural network (LSTM). According to the literature,a feed forward neural system -based language model makes use of mounted length context (Meghanathan, Nagamalai, and Chaki 2012). The recurrent neural system based language model on other hand,does not rely on a finite quantity of context. Information can circulate inside these recurrent neural networks for an extended period of time by utilizing recurrent connections . (Bhavikatti et al. 2021; Karobari et al. 2021; Shanmugam et al. 2021; Sawant et al. 2021; Muthukrishnan 2021; Preethi et al. 2021; Karthigadevi et al. 2021; Bhanu Teja et al. 2021; Veerasimman et al. 2021; Baskar et al. 2021)(Bhavikatti et al. 2021; Karobari et al. 2021; Shanmugam et al. 2021; Sawant et al. 2021; Muthukrishnan 2021; Preethi et al. 2021; Karthigadevi et al. 2021; Bhanu Teja et al. 2021; Veerasimman et al. 2021; Baskar et al. 2021)

The research gap identified from the literature is that existing approaches for the larger datasets have poor accuracy. So to increase the accuracy for the next word prediction using various approaches

for the next word generation. The study's goal is to increase word prediction accuracy by combining a long term short memory model which produces the best prediction of the next word using recurrent neural networks and comparing it to n-gram models.

MATERIALS AND METHODS

The proposed work is done in the computer networks lab, Department of Computer Science and Engineering, Saveetha School of Engineering, SIMATS. This study was developed using jupyter notebook software,and hardware configurations are intel i5 core processor, 50GB HDD, 4GB RAM, and the software configurations are windows OS, python jupyter notebook. The work was carried out on 5046 records from the text file from an online website gutenberg. The independent variable is the text file given as an input to the prediction model and the dependent variable is the prediction output. The accuracy in predicting the next word was performed by evaluating two groups.The sample size was measured as 5046 per group with the G power value 0.8 (Bengfort, Bilbro, and Ojeda 2018). A total 70 epochs were performed on each group to achieve better accuracy. The study uses a dataset downloaded from gutenberg website.

Novel long term short memory

Long term short memory preserves information from the past because the only inputs it has seen are from the past. The pseudocode for long term short memory are

Step 1: Import libraries and packages

```
import tensorflow as tf
from tensorflow. keras. preprocessing. text import Tokenizer
from tensorflow. keras. layers import Embedding, LSTM, Dense,
from tensorflow. keras. models import Sequential
from tensorflow. keras. utils import to_categorical
from tensorflow. keras. optimizers import Adam
import pickle
import numpy as np
import os
```

Step 2: Removing all the unnecessary data and label it as metamorphosis_clean.

```
file = open("metamorphosis_clean. txt", "r", encoding = "utf8")
```

Step 3: Cleaning the data

```
data = data. replace('\n', '). replace('\r', '). replace('\ufeff', '')
```

Step 4: Tokenization

Step 5: Creating the model

```
model. add(LSTM(1000, return_sequences=True))
model. add(LSTM(1000))
```

Step 6: Compile and fit the model

```
Model.compile() //complies the model with learning rate 0.001
model.fit() // fits the model x,y as targets and epochs=150
```

Step 7: Predict the model

Step 8: Results/evaluation

N-gram model

An N-gram model is built by counting how often word sequences occur in corpus text and then estimating the probabilities, since a simple N-gram model

has limitations, improvements are often made via smoothing, interpolation and backoff.

Step 1:Import libraries and package

```
from nltk. util import ngrams
from collections import defaultdict
from collections import OrderedDict
import string
import time
import gc
from math import log10
start_time = time. time()
```

Step 2: Remove unnecessary data

```
#remove punctuations and make the string lowercase
#changes the word to lowercase and remove punctuations from it
```

Step 3: Do processing

Step 4:Load the corpus for the dataset

Step 5: create the different Nc dictionaries for ngrams

```
create quadgram probability dictionary
create trigram probability dictionary
create bigram probability dictionary
sort the probability dictionaries of quad,tri and bi grams
```

Step 6:Take user input and print output

Statistical analysis

The SPSS statistical software was used in the research for statistical analysis. Using variables like training examples, corpus values, vocabulary size, input sequence length and testing samples, models get accuracy and loss values (Bengfort, Bilbro, and Ojeda 2018). Group statistics and independent sample tests were performed on the experimental results and the graph was built for two graphs with two parameters under the study.

RESULTS

The proposed algorithm novel long short term memory and existing algorithm N-gram were run at a time in jupyter notebook. As the sample sets are executed for a number of iterations the accuracy and loss values of novel long short term memory and N-gram classifier vary for next word generation as shown in Table 1. Analysis of overall prediction of next word by novel LSTM and N-gram model is done. Novel LSTM shows better accuracy 48% than N-gram. The statistical analysis for the parameters of accuracy and loss based on the iterations and epochs were done. The standard error is also less in novel LSTM in comparison with N-gram as shown in Table 2. In Fig. 1, the comparison of the significance level for novel LSTM and n-gram models was analyzed with the value p=0.02, Both LSTM and n-gram models have a less significant level less than 0.05 .

DISCUSSION

Observations were conducted among the study groups LSTM and N-gram by varying sample size, from observations the proposed novel long short term memory performed better in terms of next word generation by achieving the accuracy and less error rate compared to the N-gram model.

The N-gram model is used for the prediction. The assigned probability to each word and select the next word with the high probability. The use of a good turing algorithm for smoothing is required for removing irregularities, like cleaning datasets, creating train and test data corpus ,generating prediction by applying n-gram

model for predicting the next word (Hamarashid, Saeed, and Rashid, n.d.; Staub et al. 2012; van Gerven and Bohte 2018). The Context Based Word Prediction system outperforms the standard frequency-based technique. The optimal Markov model for modeling the causal link between parameters was chosen after a thorough examination of various Markov models. Due to the enormous number of classes, the SVM model for sequence tagging was determined to be ineffective for the given scenario (Brownlee 2017). The bi-gram model can be expanded to a tri-gram or more, however because SMS text messages are often brief phrases, a higher N-gram model isn't necessary (Kehagias 1990; Staub et al. 2012). At the moment, the simplied model first-order Markov dependency between the POS of subsequent words (Abraham et al. 2020; Forsyth 2019). Deep Learning and recurrent Neural Networks (RNNs) have greatly aided language modeling analysis in recent years since they have allowed academics to investigate a variety of tasks that the robust conditional independence requirements fail to do (Tsamtsakiri and Karlis 2021). Despite the fact that simpler models, like N-grams, rely merely on a limited history of prior words to predict the future word, they remain an important component of high-quality, low-mental-conflict Language Models (Castro 1999; Murray and Chiang 2015). Indeed, recent work on large-scale Language Models has demonstrated that RNNs perform well with N-grams because they have distinct strengths that complement N-gram models .

The limitations of the proposed language model do not have required memory for storing the predicted words .It works well for input data files of size less than 40 Mb and n-gram size 4. For larger data files and n-gram size, more memory and CPU power will be needed. Although the proposed methodology obtained satisfactory results, the approach's shortcoming is the requirement for enhanced word prediction accuracy and memory for storing the predicted word. This could be paired with more data text files in the future, resulting in improved outcomes.

CONCLUSION

The results show that the proposed LSTM outperforms N-gram in terms of accuracy and loss for next word prediction. The proposed LSTM is to remember previous output or store the previous results in memory to predict future results proves with better accuracy 48% when compared with the N-gram model for next word generation.

DECLARATIONS

Conflict of Interests

No conflict of Interest in this manuscript.

Authors Contributions

Author PNR was involved in data collection, data analysis, implementation, algorithm framing and manuscript writing. Author JJT was involved in designing the workflow, guidance and review of manuscript.

Acknowledgements

We would like to acknowledge Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (formerly known as Saveetha University) for providing facilities and constant help to carry out this study.

Funding:

We thank the following organizations for providing financial support that enabled us

to complete the study.

1. Qbecinfosol, Chennai.
2. Saveetha University
3. Saveetha Institute of Medical and Technical Sciences.
4. Saveetha School of Engineering.

REFERENCES

1. Abraham, Ajith, M. A. Jabbar, Sanju Tiwari, and Isabel M. S. Jesus. 2020. *Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019)*. Springer Nature.
2. Baskar, M., R. Renuka Devi, J. Ramkumar, P. Kalyanasundaram, M. Suchithra, and B. Amutha. 2021. "Region Centric Minutiae Propagation Measure Orient Forgery Detection with Finger Print Analysis in Health Care Systems." *Neural Processing Letters*, January. https://doi.org/10.1007/s11063-020-10407-4.
3. Bengfort, Benjamin, Rebecca Bilbro, and Tony Ojeda. 2018. *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning*. "O'Reilly Media, Inc."
4. Bhanu Teja, N., Yuvarajan Devarajan, Ruby Mishra, S. Sivasaravanan, and D. Thanikaivel Murugan. 2021. "Detailed Analysis on Sterculia Foetida Kernel Oil as Renewable Fuel in Compression Ignition Engine." *Biomass Conversion and Biorefinery*, February. https://doi.org/10.1007/s13399-021-01328-w.
5. Bhavikatti, Shaeesta Khaleelahmed, Mohmed Isaqali Karobari, Siti Lailatul Akmar Zainuddin, Anand Marya, Sameer J. Nadaf, Vijay J. Sawant, Sandeep B. Patil, Adith Venugopal, Pietro Messina, and Giuseppe Alessandro Scardina. 2021. "Investigating the Antioxidant and Cytocompatibility of Mimusops Elengi Linn Extract over Human Gingival Fibroblast Cells." *International Journal of Environmental Research and Public Health* 18 (13). https://doi.org/10.3390/ijerph18137162.
6. Brownlee, Jason. 2017. *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery.
7. Castro, M. J. 1999. "MLP Emulation of N-Gram Models as a First Step to Connectionist Language Modeling." *9th International Conference on Artificial Neural Networks: ICANN '99*. https://doi.org/10.1049/cp:19991228.
8. Chen, Ping, Jianyi Zhong, and Yuechao Zhu. 2022. "Intelligent Question Answering System by Deep Convolutional Neural Network in Finance and Economics Teaching." *Computational Intelligence and Neuroscience* 2022 (January): 5755327.
9. Forsyth, David. 2019. *Applied Machine Learning*. Springer.
10. Gerven, Marcel van, and Sander Bohte. 2018. *Artificial Neural Networks as Models of Neural Information Processing*. Frontiers Media SA.
11. Hamarashid, Hozan K., Soran A. Saeed, and Tarik A. Rashid. n.d. "Next Word Prediction Based on the N-Gram Model for Kurdish Sorani and Kurmanji." https://doi.org/10.36227/techrxiv.12725084.

12. Irene, D. Shiny, D. Shiny Irene, V. Surya, D. Kavitha, R. Shankar, and S. John Justin Thangaraj. 2021. "An Intellectual Methodology for Secure Health Record Mining and Risk Forecasting Using Clustering and Graph-Based Classification." *Journal of Circuits, Systems and Computers*. https://doi.org/10.1142/s0218126621501358.
13. Karobari, Mohmed Isaqali, Syed Nahid Basheer, Fazlur Rahman Sayed, Sufiyan Shaikh, Muhammad Atif Saleem Agwan, Anand Marya, Pietro Messina, and Giuseppe Alessandro Scardina. 2021. "An In Vitro Stereomicroscopic Evaluation of Bioactivity between Neo MTA Plus, Pro Root MTA, BIODENTINE & Glass Ionomer Cement Using Dye Penetration Method." *Materials* 14 (12). https://doi.org/10.3390/ma14123159.
14. Karthigadevi, Guruviah, Sivasubramanian Manikandan, Natchimuthu Karmegam, Ramasamy Subbaiya, SivasankaranChozhavendhan, Balasubramani Ravindran, Soon Woong Chang, and Mukesh Kumar Awasthi. 2021. "Chemico-Nanotreatment Methods for the Removal of Persistent Organic Pollutants and Xenobiotics in Water - A Review." *Bioresource Technology* 324 (March): 124678.
15. Kehagias, Athanasios. 1990. "Optimal Control for Training: The Missing Link between Hidden Markov Models and Connectionist Networks." *Mathematical and Computer Modelling*. https://doi.org/10.1016/0895-7177(90)90192-p.
16. Meghanathan, Natarajan, DhinaharanNagamalai, and Nabendu Chaki. 2012. *Advances in Computing and Information Technology: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India* -. Springer Science & Business Media.
17. Murray, Kenton, and David Chiang. 2015. "Auto-Sizing Neural Networks: With Applications to N-Gram Language Models." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/d15-1107.
18. Muthukrishnan, Lakshmipathy. 2021. "Nanotechnology for Cleaner Leather Production: A Review." *Environmental Chemistry Letters* 19 (3): 2527–49.
19. Pietro Giovanni Antiga, Luca, Eli Stevens, and Thomas Viehmann. 2020. *Deep Learning with PyTorch*. Simon and Schuster.
20. Preethi, K. Auxzilia, K. Auxzilia Preethi, Ganesh Lakshmanan, and Durairaj Sekar. 2021. "Antagomir Technology in the Treatment of Different Types of Cancer." *Epigenomics*. https://doi.org/10.2217/epi-2020-0439.
21. Sawant, Kashmira, Ajinkya M. Pawar, Kulvinder Singh Banga, Ricardo Machado, Mohmed IsaqaliKarobari, Anand Marya, Pietro Messina, and Giuseppe Alessandro Scardina. 2021. "Dentinal Microcracks after Root Canal Instrumentation Using Instruments Manufactured with Different NiTi Alloys and the SAF System: A Systematic Review." *NATO*

Advanced Science Institutes Series E: Applied Sciences 11 (11): 4984.

22. Shanmugam, Vigneshwaran, Rhoda Afriyie Mensah, Michael Försth, Gabriel Sas, Ágoston Restás, Cyrus Addy, Qiang Xu, et al. 2021. "Circular Economy in Biocomposite Development: State-of-the-Art, Challenges and Emerging Trends." *Composites Part C: Open Access* 5 (July): 100138.
23. Staub, Adrian, Margaret Grant, Lori Astheimer, and Andrew Cohen. 2012. "Predicting the Next Word: Data and Model From a Speeded Cloze Task." *PsycEXTRA Dataset*. https://doi.org/10.1037/e502412013-261.
24. Tsamtsakiri, Panagiota, and Dimitris Karlis. 2021. "On Bayesian Model Selection for INGARCH Models Viatrans-Dimensional Markov Chain Monte Carlo Methods." *Statistical Modelling*. https://doi.org/10.1177/1471082x211034705.
25. Veerasimman, Arumugaprabu, Vigneshwaran Shanmugam, Sundarakannan Rajendran, Deepak Joel Johnson, Ajith Subbiah, John Koilpichai, and Uthayakumar Marimuthu. 2021. "Thermal Properties of Natural Fiber Sisal Based Hybrid Composites – A Brief Review." *Journal of Natural Fibers*, January, 1–11.

TABLES AND FIGURES

Table 1. Comparison of accuracy obtained between LSTM and N-gram model during evaluation of the models with same datasets.

	Corpus	Training examples	Test dataset	Epochs	Accuracy	Loss/Uncertainty
LSTM	79841	12453	5046	30	15	23
				50	38	19
				70	48	29
N-gram	79841	12453	5046	2 gram	18	23
				3 gram	31	35
				4 gram	39	46

Table 2. Statistical analysis of mean,standard deviation and standard error mean for both long short term memory and n-gram algorithms.

	Group	**N**	**Mean**	**Std.deviation**	**Std.error mean**
ACCURACY	1	3	33.667	16.92	9.786

	2	3	29.333	10.56	6.119
LOSS	1	3	23.6667	5.032	2.934
	2	3	34.6667	11.50	6.641

Table 3. Independent sample test for significance and standard error determination. 95% confidence interval was considered.

		Levene's Test for Equality of Variance		**T-test for Equality of Means**						
		F	**Sig**	**t**	**df**	**Sig. (2-tailed)**	**Mean Difference**	**Std.Error Difference**	**95% Confidence of the Differences**	
									Lower	**Upper**
ACCURACY	**Equal variances assumed**	.888	.399	.376	4	.726	4.3333	11.52774	-27.67	36.339
	Equal variances not assumed			.376	3.360	.726	4.3333	11.52775	-30.226	38.8935

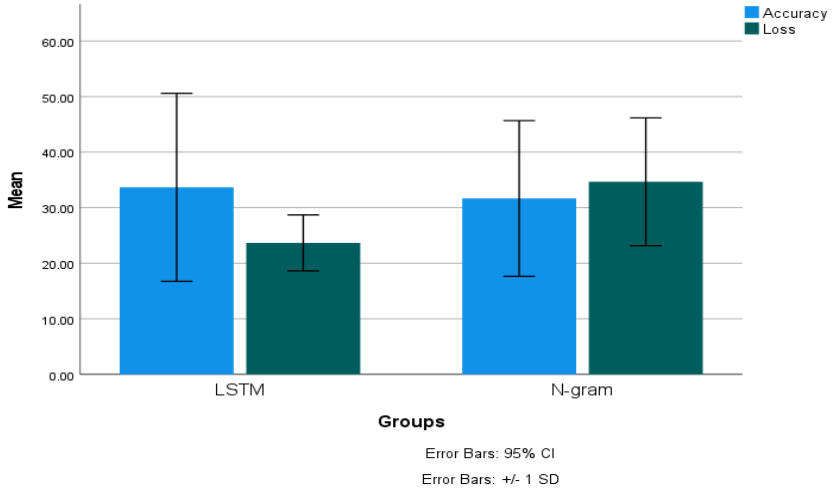

Fig. 1. Comparison of mean accuracy and mean loss of both LSTM and N-gram. The standard error appears to be less in LSTM compared to N-gram also the standard error appears +/- 1SD . X-axis: LSTM vs N-gram algorithm. Y-axis: mean accuracy and mean loss.