Next Word Prediction in Song Lyrics

Rhyma Sharma, Khushi Anand, Prof. Rishabh Kaushal
Department of Computer Science and Engineering
Indira Gandhi Delhi Technical University for Women, Kashmere Gate, Delhi, India - 110006
Email: rhyma070btcse21@igdtuw.ac.in, khushi203btcse21@igdtuw.ac.in, rishabhkaushal@igdtuw.ac.in

Abstract. Next Word Prediction models, as a part of Language Modeling, have been created in the past and widely used in text messaging, emails and documents. The predicted words are in line with the training data. The previously made models in this field that are trained on literature texts. The approach proposed in this paper is focussed on developing a system to predict the next word in song lyrics by training on a Long Short-Term Memory fetching lyrics from Genius.

Keywords: Next Word Prediction, LSTM, song lyrics, Genius

I. INTRODUCTION

Technological advancements aim to enhance efficiency along with reducing human effort in all spheres of life. A lot of recent developments focus on getting things done faster and more accurately. Predictive text is an example of how machines can mimic human language and also be able to suggest the successive words in sentences. It is widely used in text messaging services, mobile keyboards, Microsoft Word, Google Docs, emails etc.

Cameron L [1] has, in great length, written about the growth of artificial intelligence and its upcoming use and limitations in creating music, including composition and songwriting. The N-Gram model [2] is popularly used in Natural Language Processing where it exploits the statistical properties of text and the probability of occurrence of words, but its limitation is that it leads to data sparseness and dimension disaster and does not take into account the semantic structure of the language. Naive Bayes method [3] is based on conditional probability given by Bayes' theorem. But it is not very accurate. Latent Semantic Analysis (LSA) [4] models work on lexical relational Semantic Analysis of text, but the prediction results are only based on the frequency of occurrence and not language grammar. As deep learning techniques are exploited in text analysis [5], several neural networks such as Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), and Long short-term Memory Network (LSTM) are proposed. It has been shown that LSTM [6] can achieve great accuracy with serialized data while studying prediction problems. LSTM [7,8,9] is known for time series forecasting problems as it takes long-term memory and short-term memory into account in its mechanism, and makes use of long-distance temporal information to predict post-sequential time series. Therefore, the predicted words are in line with the specific user's vocabulary habits.

The model proposed in this paper is based on next word prediction in song lyrics using LSTM. Firstly, the data is collected from Genius and is written to a text file. Then this data is trained on the LSTM network and tested on lyrics input by a user.

Following is a summary of the different sections of the paper:
Literature Review deals with the previous works in this field - the models, approaches and results. Proposed Methodology consists of the data collection process

using API and its details along with the preprocessing which includes tokenization of the lyrics. Implementation section talks about the training data, batch size, epochs and accuracy. In Experimental Results and Discussion, the outcomes of the research experiment have been mentioned, including the observations made by using different sets of input. Finally, the Conclusion and Future Works section encapsulates the model's shortcomings thereby providing ideas for future research in this field.

II. LITERATURE REVIEW

In recent years, several researchers have worked on the next word prediction. Various different models have been proposed and tested for the highest accuracy, including LSTM model, RNN model, N-gram model etc.

Tomas M [10] et al. examined the probabilistic feedforward neural net language model (NNLM) and the Recurrent neural net language model (RNNLM) along with the new Log-linear models including Continuous Bag-of-Words model and Continuous skip-gram model. Their work primarily focussed on creating word vectors to preserve the syntactic and semantic regularities in language. Afika R [11] et al. proposed the LSTM model on a dataset containing 180 Indonesian destinations from nine provinces. They managed to achieve 75% accuracy which is more than the Pre Training Federated Text Model by Joel S [12] et al.

Yuerong M [13] et al. also proposed an LSTM model with an input layer, a hidden layer (which further consisted of two LSTM layers and a DROPOUT layer) and an output layer and trained it with industry data set text. The word vector is the input, and the output is the prediction probability vector after Softmax activation function normalization. The Adam optimizer is used to reduce the loss function, and the optimal model is obtained by constantly updating the iterative weight.

Chunting Z [14] et al. worked on a novel and unified model called C-LSTM which achieved excellent results outperforming individual multi-layer CNN and RNN models. Tomas M [15] et al. depicted that RNN based speech recognition models could achieve state of the art accuracy. Prasad R [16] developed a correction system to correct misspelled words in text using continuous BOW and skip-gram models.

III. PROPOSED METHODOLOGY

Fig. 1 shows the neural network architecture for the proposed algorithm that is based on the Sequential class of Keras. It consists of an Embedding, two LSTM layers and two Dense layers. The model plot is also shown in Fig. 2.

3.1 Data collection

The term 'data' here refers to the lyrics of songs the model is trained on. Lyrics are easily available on the internet but are not accessible for training. Henceforth, the process of collecting the data begins with identifying a source that provides lyrics which here is Genius (https://genius.com/).

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 5, 10)             42000

 lstm (LSTM)                 (None, 5, 1000)           4044000

 lstm_1 (LSTM)               (None, 1000)              8004000

 dense (Dense)               (None, 1000)              1001000

 dense_1 (Dense)             (None, 4200)              4204200

=================================================================
Total params: 17,295,200
Trainable params: 17,295,200
Non-trainable params: 0
_________________________________________________________________
```

Fig. 1: Neural network architecture

The first step is to set up the API Client (https://genius.com/api-clients) and generate a Client Access Token.

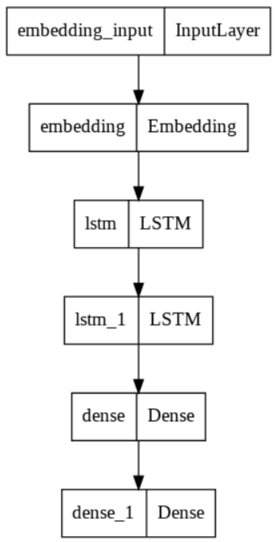

Fig. 2: Model plot

This token is used to send requests to the server and access an artist's Genius profile. This model makes use of the artist's name, ID, albums, songs and lyrics. Lyrics collection is done using LyricsGenius Client of the Genius API (https://lyricsgenius.readthedocs.io/en/master/). It makes it simple to fetch the lyrics to any song by using the song ID from Genius.

After having set up the client, the artist has to be selected. The choice is free, however, not casual. The lyrics of the artist(s) selected have to be used to train the model and thus, language and word choice will vary with various artists. This model is trained on songs by five artists: Taylor Swift, Ariana Grande, Halsey, Maisie Peters and Gracie Abrams, though other artists can be used just as well.

A request is sent to Genius and the artists' Genius homepage data is retrieved and read in .json format. From this data, artistID can be accessed. This artistID is used throughout the process of collecting data. Since Genius has several versions of the same albums and songs (including remixes, rerecordings, translations and annotations), names of albums that have to be fetched should be exclusively mentioned with the same formatting as in the original version. This is optional but it greatly simplifies the process of data collection and model training, since it is ensured that no duplicates are read. If more artists have to be included in training data, their albums can also be hardcoded. The albums are fetched using the API and saved in albumIDs.

Next, each of these albumIDs is used to fetch all songs from these albums. The names of songs are appended to songNames and their IDs are appended to songIDs. The albums fetched and the number of songs in each album are depicted in Table 1.

Table 1: Albums used for training model (July 2022)

S.No.	***Album Names***	***Total Songs***
1.	Red (Taylor's Version)	31
2.	Fearless (Taylor's Version)	27
3.	evermore	15
4.	folklore	16
5.	Lover	18
6.	reputation	19
7.	1989 (Deluxe)	19
8.	Speak Now	15
9.	Taylor Swift	16
10.	Positions	14
11.	thank u, next	12
12.	Sweetener	15
13.	Dangerous Woman	15
14.	My Everything	18
15.	Yours Truly	12
16.	Manic	16
17.	You Signed Up For This	14
18.	Trying: Season 2 (Apple TV…	9
19.	It's Your Bed Babe, It's Your…	6
20.	Dressed Too Nice for a Jacket…	6
21.	This Is What It Feels Like	12
22.	minor	7
	Total :	332

The 'lyrics' method of LyricsGenius is used to read the lyrics of each of these songs using songIDs and subsequently write them to a text file, which is saved as 'lyrics.txt'. The rest of the data processing is performed using text from this file. Fig. 3 summarizes the entire process of data collection.

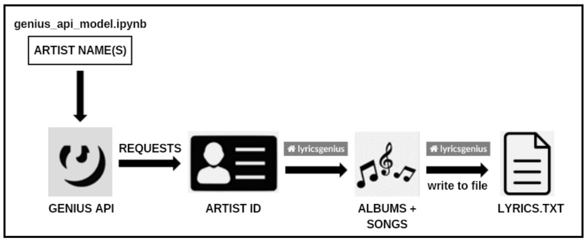

Fig. 3: Workflow diagram

3.2 Data preprocessing and Tokenization

Data cleaning has been done to replace all new line, carriage return and unicode characters with an empty string. Any other major changes in data, such as removing punctuation would affect the prediction model. Therefore, it has been avoided.

The Tokenizer class of Keras is used for vectorizing the text corpus. The fit_on_texts and texts_to_sequences methods are used to update the vocabulary and convert the text corpus into a sequence of integers respectively. Lastly, the word_index method has been used to calculate the vocabulary of the training data. The vocabulary here has 5, 945 words.

IV. IMPLEMENTATION

The proposed model predicts the next word by mapping the sequence of integers of the previous five words with the integer representing the word following them. Thus if any number of instances of one or more of those previous words are present in the input sentence, the model shall predict the next word from the mapping. Fig. 4 depicts how this mapping system works.

The training weights are saved to an h5 file. Then the model is trained on 30 epochs with a batch_size of 32 and the loss is improved to 0.1952. The accuracy is found to be 0.9406.

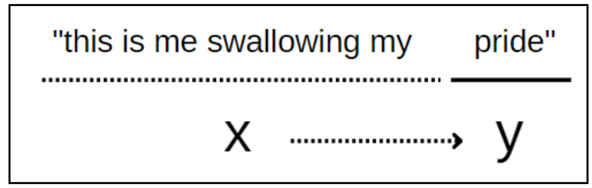

Fig. 4: Mapping of the last five words to the next

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The trained model is tested on lyrics from lyrics.txt. When the input lyrics are accurate, the predicted words are correct. By experimenting with different artists' songs, the slight overfitting can be reduced and optimum results can be achieved. More than one word prediction can also be done by either introducing more layers to the existing model or choosing a new model altogether (using transformers or N-grams).

Fig. 5 shows the prediction results for input lyrics and Fig. 6 shows the actual lyrics. As can be seen, the prediction on the existing data is fairly accurate. This strengthens the claim made by the accuracy of the model (0.9406). It has been tested multiple times with different lyrics from the file and the predicted word is found to be the same as the actual one in most cases.

```
Enter your line: Cause you know I love the
players
Enter your line: And as much as I want you to
stay
Enter your line: I stare at your clothes in the
drawer
Enter your line: the train was cold, we left
to
Enter your line: To make me feel like this would last
forever
```

Fig. 5: Prediction results

```
Cause you know I love the players
And as much as I want you to stay
I stare at your clothes in the drawer
The train was cold, we left Connecticut
To make me feel like this would last forever
```

Fig. 6: Actual lyrics

As long as it is ensured that the length of input is more than or equal to five words, the model works well. However, there may be some cases where the predicted word does not fit in a sentence. That is shown in Fig. 7 where a new set of sentences is input to the model.

Although, if a few words from the input sentence are known, prediction is fairly accurate.

```
Enter your line: painting pictures in my
head
Enter your line: i should've begged you to
see
Enter your line: how could you be so cruel with my
friends
Enter your line: in another life i will be your
time
Enter your line: let it go and run
away
```

Fig 7. Prediction results for unseen data

If new artists have to be included in the training data, their albums have to be exclusively mentioned in the code in the correct case. This is because several versions of the same albums exist on Genius and we would not want duplicate instances.

The input entered to predict the next word should have more than or equal to five words. If the user is unaware of this, prediction might fail. If the input data is random, that is, not from the training file, the chances of sensible prediction are low. This is because for the model there is little to no contextual meaning to the lyric. It predicts the next one word on the basis of previously seen words from the input sentence. To predict more words, changes will have to be made in the model along with retraining.

Since the code is written in Google Colab and uses its drive features, a lot of code will have to be changed to run it outside Colab. The paths to all the files are also mentioned in the code and will have to be taken care of.

V. CONCLUSION AND FUTURE WORKS

The predictive results of the training data are satisfactory for the words that the model has been trained on. Prediction of more than one word can be done by appropriate methods and changes in the source code. Then it can be used in creative writing extensively. Also, training on more artists' data, even from different regions and languages can greatly vary the results and make them more flexible.

As is previously stated, next word prediction models' results vary with the type of data they are trained on. This fact can be exploited to create a songwriting bot that would be an asset to songwriters in the music industry. It could also be used by creative writers to write poetry, quotes and books. This model is beginners' attempt to develop an intelligent system that is capable of writing complete songs. It often predicts words that make lyrics poetic by implying an underlying meaning. The proposed approach could be improved upon to incorporate semantics of language. For it to be applicable on real time data, that aspect is indispensable.

Creativity can be developed by training the model on advanced English tools and poetic devices like idioms, metaphors and phrases. Concepts of rhyming schemes can also be introduced to write poetry and songs. After having developed a robust songwriting model, melodies and vocals can be added and the idea can be extended to singing as well.

REFERENCES

[1] Cameron Lange, A Study of Artificial Intelligence for Creative Uses in Music, Spring, 2020.

[2] Hamarashid H K, Saeed S A, Rashid T A, et al. (2021). Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji [J]. Neural Computing and Applications, 33(5): 4547-4566.

[3] Le C C, Prasad P, Alsadoon A, et al. (2019). Text Classification: Naive Bayes Classifier with Sentiment Lexicon[J]. IAENG International journal of computer science, 46(2):141-148.

[4] Landauer TK, Foltz PW, Laham D. (1998). An Introduction to Latent Semantic Analysis[J]. Discourse Processes, 25, 259-284.

[5] Atlinar F, Ayar T, Darrige A, et al. (2020). Masked Word Prediction with Statistical and Neural Language Models[C]// 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 1-4

[6] Hochreiter S, Schmidhuber J. (1997). Long short-term memory[J]. Neural Computation, 9(8):1735-1780.

[7] Y. Liu, Y. Wang, X. Yang and L. Zhang. (2017). Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models[C]// 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 1-8.

[8] Nicole M, Klinton B. (2019). Using LSTMs to Assess the Obligatoriness of Phonological Distinctive Features for Phonotactic Learning[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 1595-1605.

[9] Ang Zifeng, Zeng Zhiqi, Wang Ke, et al. (2020). Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions [J]. Journal of thoracic disease, 12(3): 165.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781v3 [cs.CL] 7 Sep 2013.

[11] Rianti, Afika & Widodo, Suprih & Ayuningtyas, Atikah & Hermawan, Bima. (2022). NEXT WORD PREDICTION USING LSTM.

[12] Stremmel, J., Singh, A. (2021). Pretraining Federated Text Models for Next Word Prediction. In: Arai, K. (eds) Advances in Information and Communication. FICC 2021. Advances in Intelligent Systems and Computing, vol 1364. Springer.

[13] Yuerong Ma, Yating Wang, Chengjie Wan, Tian Qiao (Vol. 8 No. 8; December 2021), College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai, China.

[14] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis C.M. Lau (2015) A C-LSTM Neural Network for Text Classification.

[15] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan “Honza” Cernocky, Sanjeev Khudanpur (26-30 September 2010, Makuhari, Chiba, Japan), Recurrent neural network based language model.

[16] Prasad R, Dept. of ISE, BMS college of Engineering, Bangalore, India – 560 019, Next Word Prediction And Correction System Using Tensorflow, Copyright© 2016, IERJ.