Grammar checker & Text prediction for Kids

Avani Kasat
19BCE1058
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
avani.kasat2019@vitstudent.ac.in

Shreya Priyadarshini Roy
19BCE1779
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
shreya.priyadarshini@vitstudent.ac.in

Abstract

As technology advances overs the years, it has assisted us in so many ways possible which would have never been thought of before. Ranging from medicine, agriculture, music, finance, gaming, and various other domains, it has slowly become an intricate part of our lives. In this paper we would discuss how we used NLP techniques in the implementation of Error correction and text Prediction. The software developed has been integrated with a UI suitable for kids as they would be our target audience.

In our Grammar error correction algorithm, we used the Norvig's Approach and for the Text Generation also known as next-word prediction we implemented a LSTM model.

Keywords

NLP, Norvig's Algorithm, LSTM, RNN, TensorFlow, Grammar-Checker, Next-word Prediction

1. Introduction

Kids nowadays have been surrounded by so much technology that every task that they do needs some or the other kind of gadget assistance. Throughout the lockdown all students have attended their classes through online mode, completed their exams and assignments online itself.

This has created a habit which would be very difficult to reverse. This Technology presents children with situations and problems which are interactive and also helps them to learn how to make calculated decisions and solve problems on their own. Apps and games on technological gadgets could help give these children the practice they might need to find success on their own based on their capabilities. When students wisely use technology, they can reap huge rewards. So why not use this habit and help educate them meanwhile.

Through our work we were able to use NLP techniques and implement a Spell-checker and a Next-word prediction Model. We integrated these two models with a simple UI which would be more attractive and friendly for the children.

NLP is sub-branch of AI (Artificial Intelligence) which enables machines to comprehend and understand the complex human language. The whole goal of NLP is to build machines than can read, understand and react to the human text automatically and perform intelligent tasks on the text and help improvise it, in this case error detection and next word prediction.

Error correction is when you use an automated technology which can detect and correct the grammatical or spelling errors contained in text written by the user.

Next-word prediction basically would predict the next word/sentence depending on your input text. It is amongst the most

widely used techniques which help in enhancing the communication quality and rate in augmentative and alternative communication. This is making the life of the users convenient as it helps them to type without errors and increases their speed too. Hence, a personalized text prediction system is a vital analysis topic for all languages, especially for kids.

In the past years many autocorrect software such as “Grammarly” have come to the limelight. But most of them are to be used as plugins and don’t provide an interactive UI. They are usually made keeping the corporate crowd in mind. For our project we chose children to be our main audience. And as mentioned before in today’s world every kid uses technology. So, why not use that technology in a fun and interactive way which would benefit them.

2. Related work

In an Error Correction paper by Shanchun Zhou and Wei Liu [1], they used classification model for their algorithm. They used the rules generated by the section modules to be used on the corpus and combined it with the limited back-off algorithm. According to their results, there would be a continuous increase in the progress of the learning process with the continuous increase in the training samples

In a paper [2] by Khrystyna Shakhovska, Iryna Dumyn, Natalia Kryvinska, and Mohan Krishna Kagita: They chose three algorithms for their Text-prediction model for the Ukrainian language. Markov chains, LSTM and the hybrid.For their pre-processing they only used tokenization with removal of fast words as thye didn’t wanna loose the integrity of the structure of the sentence .According to their models they observed that, The Markov Chains performed the best amongst the three and gave the faster results with a high accuracy value. Their system unlike the T9 does not only find the next one word but also a phrase/sentence.

In the Paper [4] Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM. They used the same RNN-LSTM Model to generate new stories based on inputted stories. They divided the stories into two types: Stories with the same storyline, Different volumes of the same stories. The results generated were analysed based on their grammar, linkage of events, interest level and uniqueness. The accuracy rate they reached was 63%. These stories were also verified and evaluated by humans too.

In an article written by Sivasurya Santhanam [3] , he implemented LSTM units on a sequence based model. The network learns from the input-output function and generates texts irrespective of pragmatics. To improve the semantic consistency the model was tried with various other variations. The best result was obtained using clustering method with suitable embeddings. The system generated could be implemented in Question answering systems , Chat-Bots and even to generate random stories or reports having a format based on any topics.

3. Proposed System

The proposed system is to design an interactive web application for kids where they can read stories and novels in different languages such as ENGLISH, HINDI, TELGU.

The website would have various books from which the kid can choose any, based on his/her preference. On choosing the book they would be redirected to a page where the whole story would be there along

with interactive pictures which would appeal to the kids.

This system will allow kids to improve their grammar as it will automatically find the spelling errors and suggest the correct spelling to them. Also, can predict the next word or sentence by directly using the next-word predictor integrated as well.

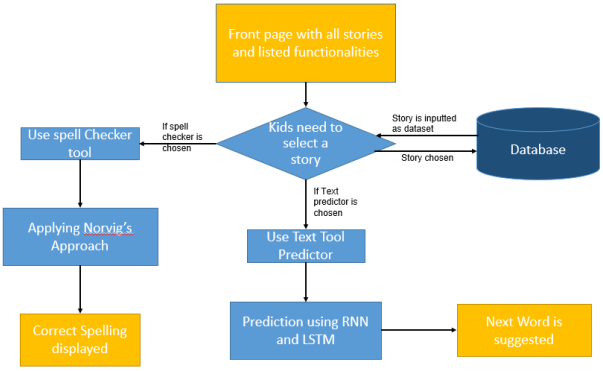

Fig. 1 System Architecture

3.1 Methodology

3.1.1 Grammar Checker:

As mentioned earlier the grammar checker would check the spelling and of the text inputted by the user. We used Norvig's approach for implementing this algorithm.

Peter Norvig who is the Director of Research at Google implemented this methodology in Google's error correction and hence the name Norvig's Approach. This method achieves over 92% accuracy at a processing speed of at least 10 words per second. [6]

This works based on a probability theory. For every given word in the corpus there is a brute force done, with all possible edits to the word. We would delete, transpose, insert and split all possible combinations. Every new word created throughout the process would be inserted into a candidate list. The probability value assigned to each word in the candidate list is divide into four parts:

Out of all the words possible the one with the maximum pro ability is chosen, the four factors which are a part of the expressions for calculating the final probabilities are:

- **Selection Mechanism**: The candidate with the highest combined probability is chosen
- **Candidate Model**: This would tell you which candidate corrections should be considered
- **Language Model**: The probability that the word appears in that language model. Example: the occurrence of "the" in English Text is 7% so P(the)=0.7.
- **Error Model:** The probability that the user would have types the given word across the candidate word found respective to it.

To find the edit distance bigger than obtained previously the procedure is repeated for second time. Each word and edit distance are obtained through unigram language model. Based on whole corpus, word frequencies are pre calculated. And the word with highest frequency is chosen as the most suitable answer.

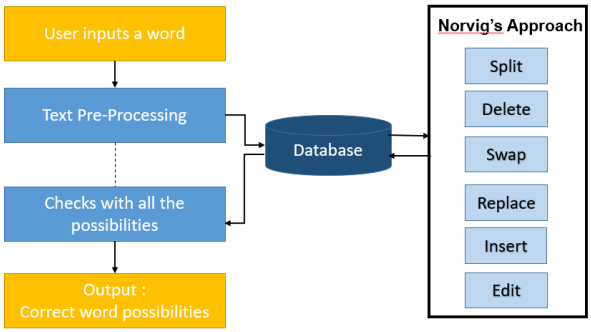

Fig. 2 Grammar-Checker Block Diagram

3.1.2 Text Prediction

We created a text predictor, character by character using LSTM recurrent neural networks in Python with Keras. Recurrent neural networks can be used as generative methods apart from being predictive models. Hence, they can learn sequences and patterns of sentence formation and help generate entirely new plausible sentences.

LSTM recurrent neural networks are slow to train but the results are much better compared to all the other methods. We first started by creating the whole setup i.e., we developed a small LSTM-RNN. We imported all the libraries required for it and then read the input data. Text processing was done on the data. Text processing includes functions like removing excess characters and vectorizing them. Before training them, we needed to convert them into numerical representation, as we cannot model the characters directly.

The pre-processing layer can convert each character into a numeric ID. It starts by splitting it into tokens first. The input to the model will be sequence of characters and we would train the models in such a way that it would predicts the following character at each step. We would then divide the text into example sequences and each input will contain length of characters from text. For the target sequence we would increase this length by one character to the right.

For training the model we needed a dataset of (input, label) pairs, where input and label are sequences and at each step input is the current character and label is the preceding character. But before feeding the data into the model we would also need to shuffle the data and pairs into batches.

Next, we would setup the Model. Our model consists of three layers. The input layers, a trainable lookup table that would map every character-ID to a vector. A RNN/LSTM type of layer, where all the processing and internal operations would go on. An output layer, which would display every next character in the vocabulary for the respective input.

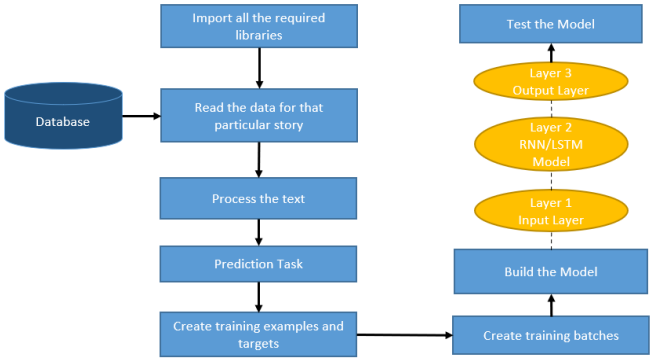

Fig. 3 Text Prediction Block-Diagram

4. Result & Discussion

Though our project we implemented two NLP based Models to obtain a spell checker and a Text-Predictor.

The results obtained for the Text-Prediction Model is an accuracy rate of 99.2%.

The Model Summary is as given below with three layers:

Input, RNN Layer & Output Layer.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 60, 128)           114176
_________________________________________________________________
dropout (Dropout)            (None, 60, 128)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 128)               131584
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 94)                12126
=================================================================
Total params: 257,886
Trainable params: 257,886
Non-trainable params: 0
```

While fitting the last layer of the model we received improved the loss from 0.77814 to 0.76901.

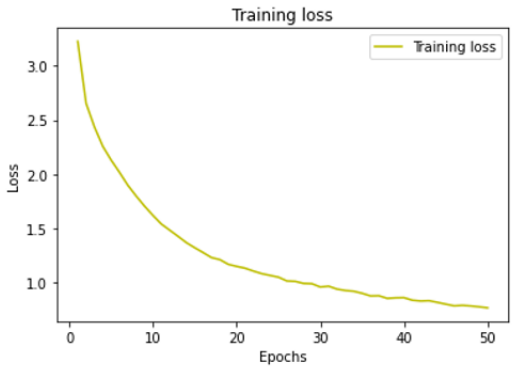

Fig. 4 Training Loss for Text Prediction Model

As seen in the graph the number for epochs is increasing while the loss is decreasing.

For the Grammar Checker we attained an accuracy Rate of 91.8% at 10 words per second.

The model we implemented was Novig's approach where we did the following permutation for every input word.

Split characters from the root word and try permutations of all alphabets. *Delete* Character(s) and replace them with all alphabets and save the new produced words in the candidate dataset. The other operations also followed were *Swap, Replace, Insert* and *Edit.* Once these were all done, the new words generated were added to the candidate dataset.

Memory Usage	Accuracy(f1)	Speed
6 GB	67.7%	230 words/sec
8 GB	73.6%	180 words/sec
13 GB	79.3%	100 words/sec
11 GB	91.8%	10 words/sec

Fig. 5 Accuracy for Spell Checker

5. Conclusion

This paper shows the basic implementation of Text prediction using LSTM-RNN network and Grammar checker implementation using Norvig's Approach.

Both Word Prediction and Grammar Checker are very helpful tools in today's world. It not only boosts up the user's typing speed but also helps in omitting mistakes. An autocorrect predictive text system personalized for ones-self is a relevant research topic that can be implemented in all languages as of now we have implemented this for three languages: English, Hindi and Telegu.

We received a fairly high value of accuracy for both i.e. the spell checker with an accuracy of 91.8% at a processing speed of 10 words per second, And the Text Prediction with a loss of 0.7% i.e. precision is 99.2% .

We believe that this would benefit kids and improve their grasp over the language, If used and implemented wisely.

6. Future Works

As of now we have implemented this for a definite corpus (the stories selected and uploaded by us). In future we can use a dynamic corpus which can be uploaded and updated instantly.

Another addition that can be done is using other languages. We have implemented our project in three languages, but it can always be expanded for many more.

7. References

[1] English Grammar Error Correction Algorithm Based on Classification Model- Shanchun Zhou1 and Wei Liu

[2] An Approach for a Next-Word Prediction for Ukrainian Language- Khrystyna Shakhovska , 1 Iryna Dumyn , 1 Natalia Kryvinska , 2 and Mohan Krishna Kagita

[3] Context Based Text-Generation Using LSTM Networks- Sivasurya Santhanam https://arxiv.org/pdf/2005.00048.pdf

[4] Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM- Mrs. Dipti Pawade, Ms. Avani Sakhapara, Ms. Mansi Jain, Ms. Neha Jain, Ms. Krushi Gada

[5] S. Mangal, P. Joshi, and R. Modak, "Lstm vs. gru vs. bidirectional rnn for script generation,"2019, https://arxiv.org/abs/1908.04332

[6] Using the Web for Language Independent Spellchecking and Autocorrection- Casey Whitelaw and Ben Hutchinson and Grace Y Chung and Gerard Ellis

[7] Text prediction systems: a survey Published online: 8 December 2005 Springer-Verlag 2005- Nestor Garay-Vitoria & Julio Abascal.

[8] Treat the system like a human student: Automatic naturalness evaluation of generated text without reference texts- Proceedings of The 11th International Natural Language Generation Conference.2018. -Ye Tian Ioannis Douratsos Isabel Groves

[9] CrossCheck -a Grammar cheker for second language writers of Swedish – Professor Viggo Kann-2001

[10] Chunk-based Grammar Checker for Detection Translated English Sentences- International Journal of Computer Applications August 2011.- Nay Yee Lin, Khin Mar Soe, Ni Lar Thein.

[11] Secondary School Students' English Writing Aided by Spelling and Grammar Checkers- Odette Radi, Australia.