International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Emoticon Suggestion with Word Prediction using Natural Language Processing

Rahman Mahte[1], Ranjith Nair[2], Vysakh Nair[3], Athira Pillai[4], and Prof. Manasi Kulkarni[5]

[1-5]*Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, India-410206*

Abstract - *Emojis are a very important part of communication in today's world. It is used to express emotions during a conversation. Building a system which can suggest emoticons based on the text provided can be very useful. It can be used to express emotions efficiently and easily. While dealing with the semantics of the sentence it can be used to predict the emotion in the sentence and emojis can be predicted accordingly. Typing each and every word to complete a sentence is also a very time consuming task with the help of word prediction models this task can be made much easier. So in our project combining two models i.e Word prediction and emoji suggestion will improve textual communication. Emoticons add life to sentences and word prediction helps in framing correct sentences. It makes sentences more understandable and appealing. A system which can help in quoting the correct emoji into a sentence easily can be very useful in today's world.*

Keywords: **Emoticon, Suggestion, Semiotic, Recommendation, Expression, N-Gram.**

1. INTRODUCTION

Living in the world of social media conversation through text and messages play a very important role. In each and every aspect of life for communication messages are used. It is very important to frame the messages correctly so that the meaning of the message is conveyed to the point. At this stage the role of word prediction comes into existence. With an accurate word prediction model sentence framing can be achieved easily. Once correct sentences are framed, words from these sentences can be picked to suggest emoticons. At this stage the role of emoticons comes into existence. With the use of accurate emojis the meaning of the message can be conveyed easily. Picking the correct emojis from a list is a time consuming task. So an emoticon suggestion system can be of great help for effective communication. Emoticons make sentences more lively and appealing. With text based communication being a very important part of our day to day life a suggestion system can help in making this work easier.

1.1 Fundamentals

The main purpose of the system is to frame correct sentences and to suggest emoticons for the same. The user could select any other word of his or her choice than from the predicted word. Emoticons will be suggested for keywords existing in that sentence. Emoticons relevant to the context will be suggested.

The system will make use of bigram trigram model to achieve word prediction and help in forming sentence which will be further provided as input to the emoticon suggestion model

1.2 Problem Statement

Building a system to complete a sentence or incomplete words with the help of given predictions and using the formed sentence to finally suggest emoticons to make the sentence more appealing. While predicting word for sentence completion a lot of parameters have to be considered. Provided input by the user may be wrong, most similar words to the wrong input should be predicted and to make the complete text more appealing with the help of emoticon suggestion model this can be achieved.

2. LITERATURE SURVEY

2.1.Emoticon Recommendation System Reflecting User Individuality [2]

Taichi Matsui and Shohei Kato, 2017 in their survey, Emoticon Recommendation System Reflecting User Individuality: A Preliminary Survey of Emoticon done in this paper shows that the way of using emoticons for each individual differs widely. Dividing the emoticons into different clusters and predicting the emojis further according to the cluster the selected emoji belongs to makes the prediction easier. Each individual has a different way of using emoticons. With the help of division of clusters the way of use by different individuals can be predicted. The survey done in this paper shows how a limited number of people prefer using emojis by observing the statistics of their connection circle.

2.2 RNN based Emoticon Suggestion [3]

Dineshika Dulanjalee Wijerathna, 2016 in Emoticon Suggestion based on Recurrent Neural Network talks about the application of RNN in suggesting emoticons which can be applied in chat applications. Recurrent neural networks are used for the prediction of emoticons. In this approach there would be a recurrent neural

network which would consist of the first layer, the hidden layer and the last layer. When the user uses any emoticon, its sequence would be recorded and thus one of the neurons in the first layer would hold the value of that emoticon. Based on the number of times the emoticon is used in that particular sequence, the weightage of that neuron would increase. Many such neurons would be holding different values or different sequences in which emoticons were used. When the user types a sentence, its sequence would be used to predict the emoticon. Out of all the sequences of sentences and emoticons used previously, the most likely emoticon would be suggested.

2.3. Semiotic based Sentiment Score [4]

Darsha Chauhan, Kamal Sutaria and Rushabh Doshi ,2018 in their paper Impact of Semiotics on Multidimensional Sentiment Analysis on Twitter include various methods to determine sentiment score of a statement with semiotics. Sentiment Analysis plays a very important role in any text based prediction system. In this paper the research is done on sentiment analysis in which text based input is processed to find the sentiments related to the input. In this method the keyword extraction of the given text is done. Each keyword shows the sentiment related to it from the dataset. Analysis based on the sentiments extracted from the input is further used to describe the complete sentiment of the given input.

2.4. Emoticon Recommendation System[5]

Yuki Urabe, Rafal Rzepka and Kenji Araki, 2013 in Emoticon Recommendation System for Effective Communication describes the development of an emoticon recommendation system that allows users to express their feelings with their input. This paper discusses the usage of the ML Ask approach. In this approach the system separates the emotive utterances from non-emotive utterances. This utterance is then used for determining the emotive utterance. The different emotive utterance that it uses are joy, delight, anger, excitement, sadness, gloom, liking, fear, relief, dislike, surprise, amazement and shyness. This system is said to produce an accuracy of 71.3%.

3. EXISTING WORK

In an ordinary system every keyboard has a list of emoticons in it. In these keyboards words are predicted for efficiency but in certain conditions users prefer emoticons over words. In this condition a prediction system can be very useful. In the existing system the normal keyboards contain a section of emojis. If a prediction system is developed it can be integrated with the keyboard which predicts words. After typing a sentence, the user has to manually select an emoji from the list of emojis available. This reduces the use of available emoticons.

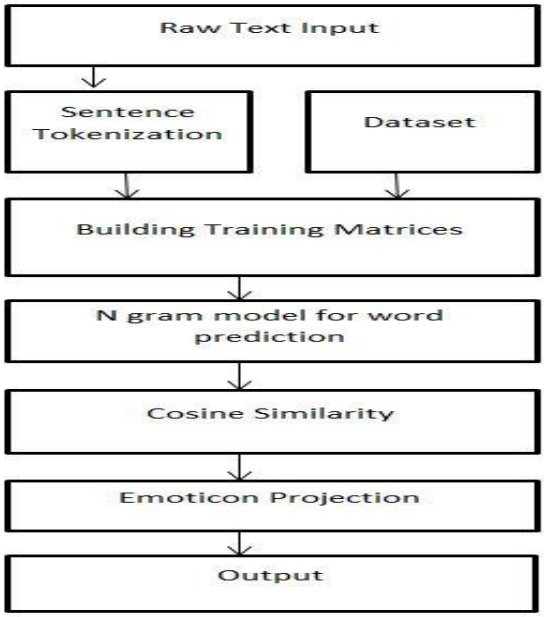
With the existing systems the statistics showing the use of emoticons is less. The usual system makes the task of selecting emoticons tedious. The system records the use of emoticons by each individual and gives it in the form of recently used. This can be helpful if a similar type of communication is done. With the use of new types of sentences this system grows less efficient.

A little advance system gives the feature of text analysis by considering limited emotions. A list of smiley emojis are presented based on the past data and past use of emoticons by the user.

4. PROPOSED WORK

Our System consists of two models, the first part contains word prediction and the second part consists of emoticon prediction. Word prediction model uses brown corpus which contains data from different sources e.g. news articles, novels etc. Implementing bigram and trigram word prediction is achieved. This data is fed to the second part i.e emoticon suggestion which uses 'GloVe Vector' and cosine similarity.

The system architecture is given in Figure 1. Each block is described in this Section.



Fig(1): Proposed System Architecture

4.1 Raw Text Input:

This will be the input which is given by the user of the framework. It can consist of sentences which are utilized in everyday life. This content will go about as the contribution to the framework. To test the framework the info should comprise of watchwords which can be spoken to utilizing existing emojis. Despite the fact that in

ordinary life situations content information can be any information however for a framework to give emoji information ought to be significant information.

4.2 Tokenization:

Tokenization is a key step in NLP. In lexical analysis, tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.

The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.

4.3 Input Corpus:

GloVe file is utilized to condition and guide emoji depiction provided in emoji dataset with the glove vectors. It is an unsupervised learning algorithm for getting vector representation for words. Training is performed on aggregated worldwide word-word co-event insights from a corpus, and the subsequent portrayals grandstand intriguing straight foundations of the word vector space. A dataset of emojis comprises emoji depiction and their individual unicodes. The emoticon dataset has a high bias towards american culture.

4.4 N-Grams Model:

Provides the capacity to autocomplete words and proposes prediction for the following word. This makes typing faster, more intelligent and reduces effort. Probabilistic models are used for computing the probability of an entire sentence or for giving a probabilistic prediction of what the next word will be in a sequence. This model involves looking at the conditional probability of a word given the previous words

4.5 Emoticon Projection:

The input data is vectorised to compare with the map of glove vectors and emoticons. The system uses cosine similarity method to compare the data with each vector in the map and then find the vector to which it is closest to i.e having the highest cosine similarity. The data is converted to emoticons if its cosine similarity crosses the provided threshold value.

5. IMPLEMENTATION

5.1 Word Prediction

5.1.1 Designing a keyboard interface:

The initial undertaking was to structure a console interface as a web application. The keyboard comprises all keys which are available on a physical console. The console's interface will provide the best three words for a given sequence of words and also recommend word-completion. This interface was created by using HTML and CSS. The model takes Dynamic input and output of words by using XAMPP, JavaScript and AJAX.

5.1.2 Using Bigram and Trigram model to suggest predictions on the software keyboard:

To predict words in a sequence, a bigram and a trigram module were generated in python. The bigram module computes the probability of a word after a given sequence. This is accomplished by saving all the potential words in the corpus, inside a variable(in python), which can occur after a given past word. The count of this bigram is a key-value pair in a hash map. The probability can be determined by dividing the value(count) by the total number of times the given word happens in the corpus. Similarly, the trigram module is used as a hashmap of hashmaps consisting of the potential words preceding a sequence of words (two words) with their particular count. Faster lookups can be accomplished by using Hash maps.

5.1.3 Using Minimum Edit Distance Module for auto-completion:

Often in real world typing, a user might make typing errors, for which a clever typing assistant must be capable of making suggestions for. This is implemented by utilizing the Minimum Edit Distance concept which tries and makes predictions as to what the user wanted to type. This is implemented by using dynamic programming that finds the least number of addition, subtraction and substitution needed to make one word completely the same with another. This module however, while using an enormous number of words in the corpus, doesn't provide a very time-efficient performance. Thus in our implementation, the nltk function which finds the Levenshtein distance between given words is used. We additionally permit unit transposition cost to factor in situations where the user may have typed "draem" rather than "dream", as it is so much common while typing quickly. We take all the potential predictions after the last word, and store them in a dataframe. We then find the similarity between every prediction and what the user has typed. Then a lambda function is used to sort these. The prediction with the highest similarity is shown first, then the second, then the third.

5.2 Emoticon Suggestion

5.2.1 Emoticon Suggesting model

The model takes the user input and tokenizes it. It changes over every single word to an emoticon on the basis of similarity (cosine similarity). The emoticon unicodes are

processed with 300 dimensional glove vectors and mapped on it. A comparison is made between each word and their corresponding guide. A minimum value of similarity is set. Any time a word has a similarity higher than the minimum similarity value, its emoticon is printed. The models utilize 300 dimensional Global Vector environment available on the Wikipedia corpus as word embeddings.A aggregate of 3415 emoticons are utilized in this framework. On finding the similarity of crossing the threshold value, the framework separates the emoticon depiction to utilize the emojis capacity of python for the conversion. This project lays a greater emphasis towards Americans and their way of speaking. Better word embeddings with a bigger corpus must have sufficient data that covers all types of speaking behaviour like slang language, formal language and informal language. An example of the working of our model:

star boy - star

the pizza is great - the is great

5.3 Algorithm

Step 1: Input Text from user

Step 2: Word prediction and auto-completion of the word

1. Using Bigram and Trigram model to suggest predictions on the software keyboard
2. Using Minimum Edit Distance Module for auto-completion
3. Building a simple python server in flask

Step 3: Tokenize the input text.

Step 4: Implementing emoticon suggesting model

1. Emoticon and glove vector mapping is done
2. Tokenized inputs are converted to vectors
3. Calculates the cosine similarity between tokens and map
4. Highest cosine similarity word crossing the threshold is chosen

Step 5: The chosen word is converted to emoticons.

6. RESULT ANALYSIS

Following graph depicts the aggregated outcome by using the bigrams and trigrams.

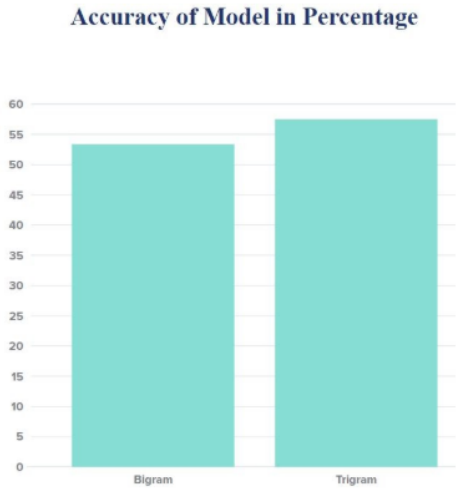


Fig 6.1 Accuracy of the models.

A sample of ~675,000 words were taken from the test dataset. The time was calculated as the total time required by the NGram models to go through all the n-grams. The Bigram model accurately predicted the next word 54% of the times and the trigram model predicted the next word 59% of the times. The next table shows the hit (correct predictions) and the overall predictions estimated for each N-Grams model.

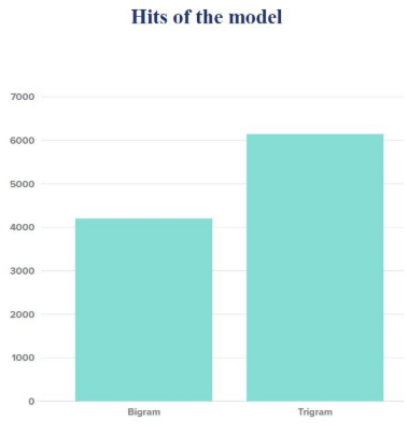


Fig 6.2 Accuracy of the models.

We made a total of 7873 attempts for Bigrams out of which a total of 4251 correct predictions were made. 10682 attempts were made for Trigrams out of which 6303 correct predictions were made.. The accuracy figures

look simplistic at the first look, but keeping in mind the vast range of vocabulary used in the corpus(~675,000) and the nature of the bigram and trigram model, it seems as the accuracy of the model is quite good. Even the time complexity is good, considering the vast set of matrices that are generated by traversing the entire dataset.

7. CONCLUSION

Developing a system helps to make some tasks easier. The same way an emoticon suggestion system can be of great use in the world of texting. It has the ability to autocomplete and predict the next word based on the previous words which will increase the speed of typing and increase the efficiency of the system hence making the system more intelligent. The system will increase the statistics of use of emoticons in conversation. With a prediction system the manual task of selection of emojis is reduced. This increases the use of unused emoticons. With the use of accurate emojis the meaning of the message can be conveyed easily. Picking the correct emojis from a list is a time consuming task. So an emoticon suggestion system can be of great help for effective communication.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Prof. Manasi Kulkarni**, our project in charge, for her guidance and constant supervision as well as her support and help regarding project related information which helped us in the completion of the project. We are grateful to **Dr. Sharvari Govilkar,** HOD Dept. of Computer Engineering, for encouraging and allowing us to present the project on the topic Emoticon Suggestion with Word Prediction. We express our gratitude to **Dr. Sandeep M. Joshi** Principal, PCE New Panvel for providing us with a platform to utilise this golden opportunity to expand our knowledge and experience.

REFERENCES

1. Urabe, Yuki & Rzepka, Rafal & Araki, Kenji, "Emoticon Recommendation System to Richen Your Online Communication", International Journal of Multimedia Data Engineering and Management (IJMDEM),Vol. 5,No.1, June 2014

2. Taichi Matsui and Shohei Kato,'Emoticon Recommendation System Reflecting User Individuality-A Preliminary Survey of Emoticon Use', ICAART, 2017

3. Dineshika Dulanjalee Wijerathna, 'Emoticon Suggestion based on Recurrent Neural Network', University of Moratuwa, 2017.

4. Darsha Chauhan, Kamal Sutaria and Rushabh Doshi,'Impact of Semiotics on Multidimensional Sentiment Analysis on Twitter: A Survey', IEEE,2018

5. Yuki Urabe, Rafal Rzepka and Kenji Araki, 'Emoticon Recommendation System for Effective Communication', IEEE , August 2013

6. Bin Wen, Ping Fan, Wenhua Dai, Ling Ding, 'Research on Building Chinese Micro-Blog Semantic Lexicon', MEC, 2013

7. Yasutaka Toratani, Makoto J. Hirayama, 'Psychological Analysis of Emoticons Used for E-mails on Cellular Phones', IEEE, October 2011

8. Jeffrey Pennington, Richard Socher & Christopher D. Manning, 'GloVe: Global Vectors for Word Representation'[Online].Available:https://nlp.stanford.edu/projects/glove/

9. Medium.com,'Using Perplexity to Evaluate a Natural-Language Model',[Online].Available: https://medium.com/@davidmasse8/using-perplexity-to-evaluate-a-word-prediction-model-8820cf3fd3aa

10. Jaysidh Dumbali, Nagaraja Rao A. ,'Real Time Word Prediction Using N-Grams Model', International Journal of Innovative Technology and Exploring Engineering (IJITEE) , Volume-8,Issue-5, March 2019

11. Xuying Meng, Suhang Wang, Huan Liu and Yujun Zhang,'Exploiting Emotion on Reviews for Recommender Systems'

12. Shatha Ali A Hakami,'The Importance of Understanding Emoji: An Investigative Study', 2017

13. Alison P. Ribeiro and Nadia F. F. da Silva,'#TeamINF at SemEval-2018 Task 2:Emoji Prediction in Tweets ',2018

14. Ruobing Xie1, Zhiyuan Liu1 , Rui Yan2 and Maosong Sun1,'Neural Emoji Recommendation in Dialogue Systems',2016