IOT Data Mining:A New approach with Classification Model

B.Mallikarjun Rao*[1] Dr B.V.Ramana Murthy*[2]
*[1]Research Scholar, *[2]Professor & HOD (CSE)
*[1]Rayalaseema University-Kurnool, *[2]Stanley College of Engineering-Abids.

Abstract – **Emojis can be expressed to show the emotional and sentimental feelings in terms of graphical symbols which can add weightage to the text statement. Emojis have been widely used in social media and chat conversation on both web as well as mobile applications. With usage of emojis, the textual communicating statements will be more expressive & effective. Therefore, it is important to mine the relationship between the text & emoji symbols. By using the advanced machine learning techniques, related emojis can be suggested automatically while understanding the real context of the statement. In this paper, I present a neural network methodology to predict appropriate emoji symbol for a given textual statement and the machine learning model is built by using word vector representations & deep learning frameworks. Model is created by using Global Vector (GloVe) embedding representation, combination of long short-term memory (LSTM) network and convolutional neural network (CNN) and softmax layers. Algorithm is able to generalize and associate words in the test set**

& finds the proper emoji symbol even. The model becomes an accurate classifier mapping from sentences to emoji symbols, even if the words do not appear in the training set.

***Keywords*—Emojis, LSTM, Deep Learning, Neural Networks, Sentimental Analysis**

I. INTRODUCTION

Emojis usage is increasingly popular on both social media applications and corporate communicators such as Twitter, Facebook, Whatsapp and Lync. Emojis are usually used with the combination of textual information to convey the emotions & feelings. It is an important task to establish the correlation between the emojis & the textual content. Typically, studies on usage of emojis mainly focus on analyzing the semantics of the text based messages. Wijeratneet al. presented the including the emojis along with the text description to enhance the quality of text expression. However, these approaches cannot establish the real context of full statement, instead, the focus is only on the individual words. Thus, the performance may be sub-optimal. Peijun Zhao presented on training of the Neural network and is based on Twitter text messages which are, most of the times, of the combination multiple languages, unknown words. Hence, for the unseen words, this model cannot correlate to the emojis.

In order to overcome the incorrect prediction problems, I propose deep learning based approach which is based on GloVe database, Bi-directional LSTM layers to perform on unseen text words. This also increases the speed of model creation as the minimal training data set is sufficient to verify the accuracy of the prediction. GloVe database is very much handy to finalize the hyper parameters to conclude the model architecture.

II. PROPOSED APPROACH

In this section, I introduce the details of my deep learning based neural model. The architecture of the model is shown in Figure 1.

Global Vector embedding Layer

Global Vectors are used for word representation & more than 2000 papers are published. GloVe provide useful information about the meaning of individual words and used to calculate the correlation between the different words. Semantic vector formatted representation of a language evaluates each word with a real-valued vector. These vectors are of specific weighted least squares model which trains on a data set with word-word co-occurrence and frequency counts and hence makes efficient use of statistics. The model produces a word vector space with different dimensional substructure like 50, 100 and 150 dimensions. For example, cosine_similarity(father - mother) = 0.890903844289 and cosine_similarity(ball, crocodile) = 0.274392462614. Words father-mother are having higher similarity where as ball-crocodile are having lower similarity. Having this co-relational vector representation of all of the English dictionary words, the proposed model can perform well on unseen or new words. The input statement will be converted into GloVe formatted vector representation and the average of the embedding layer as a single encoded vector will be fed to LSTM layer.

Long-short term memory layer (LSTM)

LSTM layer has great feature of remembering the pattern of the statement for long durations of time to evaluate the context of textual information. Major advantage of LSTM is that this network is comprised of various memory blocks called cells. The cells are responsible for remembering the things. Cell state and hidden state are the two states which get transferred to the next cell. There are three gates through which, manipulations to the memory blocks are being done. Forget gate is used to remove the information from the cell state. Input gate is used to add the information to the cell state. Output get is used to select the useful information from the current cell and giving as an out to next cell. All the three gates work together sequentially to process the information and identify the real context of the input statement. Based on the evaluated context or important word, the proposed model predicts the proper emoji symbol.

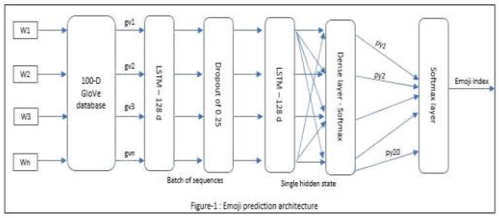

Figure-1 : Emoji prediction architecture

III. MODEL CREATION

The input of the model is a string corresponding to a sentence (e.g. "I love you") and gets converted into lower case for the sake of simplifying the model creation. The first step is to convert the input sentence into the word vector representation, which then get averaged together. As shown in Figure (1), I have considered pre-trained 100-dimenstional GloVe embeddings. I created word_to_index, index_to_word and word_to_vec_maps for all of the vocabulary available in the English dictionary. GloVe dataset incorporated vectors for 400,001 words indexing from 0 to 400,000. Thus, each word of the input text gets represented as a 100 valued vector and all these vectors get averaged. Since I consider for 20 classifications of emoji symbols, created OneHotEncoding representation for output labels.

```
for w in input_sentence:
avg += word_to_vec_map[w]
avg = avg / len(words)
```

Word embedding layer is used as input to LSTM layer & the training of the data is being done as mini-batches. Since LSTM has constraint that length of all test & train dataset should be consistent, I am considering the length of statement as 10 which is decent size to calculation the model performance. If the length of input statement is of lesser than 10, then padding is used to make it to fixed size. As part of initial phase, convert all training sentences into list of indices and then zero-pad if the length is lesser than 10. The Embedding layer takes an integer matrix of size (batch size, max input length) as input and these sentences converted into lists of integers formatted indices as shown in the Figure 2.

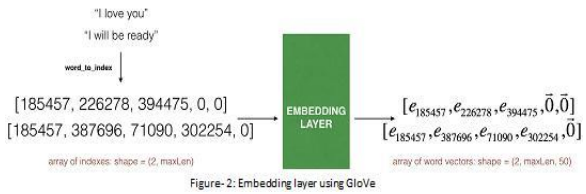

Figure- 2: Embedding layer using GloVe

The textual information is represented as list of words which
is denoted as W = [w_1, w_2, ..w_{10}] and these words get translated into GloVe formatted numerical data as E = [e_{xx}, e_{yy},..].

LSTM -1 layer is constructed with 128-dimensional hidden state with returning as batch of sequences. Output of LSTM-1 is given to Dropout layer with the probability of 0.25. Another LSTM – 2 is integrated with 128-dimensional hidden state with returning output as a single hiddent state. Now the output of LSTM-2 is an input to Dropout layer with the probability of 0.25. The output is then fed to Dense layer with Softmax activation function and size of 20 which is the final output label which are emoji classifications. Finally Softmax layer is given the input with size of 20 which are probabilities. Based on higher probability, softmax layer finalizes the predicted emoji index which is then mapped to corresponding emoji symbol.

Number of hyper parameters involved in model creation:

vocabulary size: 400,001
non-trainable parameters: 400,001*100 = 40,000,100
Trainable params: 223,877
Total params: 40,223,977

After creating the model, it needs to be compiled with various specifications like loss, optimizer and metrics (accuracy). Model is trained with batch size of 32, 50 epochs and with shuffle enabled.

The softmax layer taken works as normalized exponential function which is a generalization of the logistic function.

IV. PERFORMANCE EVALUATION

In this section, I compare the performance of my proposed approach with the available neural network techniques.

- SVM, using K independent word data set, model to predict for each emoji.
- CNN, using Convolution Neural Network as the word encoder and predict the emojis.
- LSTM with GloVe, proposed model.

According to my experiments done, observations as below:

SVM model perform well as it takes independent words
& starts mapping word to emoji. Even though, SVM model is giving good accuracy, this cannot consider the real context of the full statement & performance is poor in case of negative words like 'I did not win the match', 'I am not happy'.

Similar to SVM performance, CNN model works on only individual words and cannot remember the earlier important words to derive the real meaning of the full statement.

My proposed approach gives decent accuracy with both on unseen or new words as well as by considering the full context of the textual statement to derive the real meaning of the full statement.

Apart from default accuracy calculation with below parameters, I also evaluated the performance with F1 score as mentioned in Table 1.

```
loss='categorical_crossentropy',
optimizer='adam',
metrics=['accuracy']
```

Method	Precision	Recall	F1-score
SVM	0.325	0.326	0.235
CNN	0.354	0.379	0.304
LSTM – GloVe	**0.380**	**0.402**	**0.379**

Table 1: F1 score comparison

V. CONCLUSION

In this paper, I introduce neural network model with LSTM with combination of GloVe embeddings to predict the emoji symbol while understanding the real context of the full statement & to create the model in simple and faster way. With the use of GloVe embeddings, tweaking the hyper parameters has become drastically faster to evaluate the model performance as well as on fixing the hyper parameters.

VI. ACKNOWLEDGMENT

This work is guided by Dr. M. Akkalakshmi, M.Tech, Ph.D, Professor & HOD and Sai Leela, M.Tech, Ph.D, Assistant Professor, Information Technology department, GITAM, Hyderabad.

REFERENCES

[1] Yoonjung Choi, Youngho Kim, Sung-Hyon Myaeng Domain-specific sentiment analysis using contextual feature generation (2009)

[2] Mikalai Tsytsarau, Sihem Amer-Yahia, Themis Palpanas Efficient sentiment correlation for large-scale demographics (2013).

[3] Jeffrey Pennington, Richard Socher, Christopher D. Manning GloVe: Global Vectors for Word Representation (2015).

[4] Anastasia Giahanou, Ida Mele, Fabio Crestani Explaining Sentiment Spikes in Twitter (2016)

[5] Yen-Jen Tai, Hung-Yu Kao Automatic Domain-Specific Sentiment Lexicon Generation with Label Propagation (2013)

[6] Shubhamoy Dey, Anuj Sharma
An artificial neural network based approach for sentiment analysis of opinionated text (2012).

[7] A. Koumpouri, I. Mporas, and V. Megalooikonomou Evaluation of Four Approaches for "Sentiment Analysis on Movie Reviews": The Kaggle Competition (2015)

[8] Jürgen Schmidhuber, Sepp Hochreiter Long Short-Term Memory (1997)

[9] Peijun Zhao, Jia Jia, Yongsheng An Analyzing and Predicting Emoji Usages in Social Media (2018).

[10] Thomas Dimson. 2015. Emojineering Part 1: Machine Learning for Emoji Trends (2015).