UNDERSTANDING INFORMAL TEXTS, ABBREVIATIONS AND EMOTICONS USING NLP FOR SENTIMENT ANALYSIS

J Anuradha[1] and Pranab Ray[2] Sushmita Ghosh[3]

[1] Associate Professor, School of Computer Science & Engineering, Vellore Institute of Technology, India.
januradha@vit.ac.in
[2] Computer Science & Engineering, Vellore Institute of Technology, India.
[3] Computer Science & Engineering, Vellore Institute of Technology, India.

Abstract. Nowadays, the usage of social media has grown to a large extent. And the use of Informal texts and emoticons is common to express their feelings in chatting, blogs etc. Understanding abbreviations/ shorthand and emoticons from the normal texts are one of the challenging tasks. An abbreviation is a shortened form of a word or phrase, with locations, cultures etc. People use different abbreviations of the same word and understanding those abbreviations with respect to proper semantics is one of the backbreaking tasks by a computer. In this paper, our objective is to detect shorthand/abbreviations and emoticons meaning from the normal conversation with the help of our new proposed system which is using the Longest Common Subsequence algorithm

Keywords: emoticons, shorthand message, abbreviation

1 INTRODUCTION

With the innovation of smart mobile gadgets and advancements in this era has given rise to social media. Approximately over 60% of populace owning a smart mobile gadget associated with the web. There has been a gradual increase in usage of social media which provides an open stage of communication in modern time. Shorthand message means messaging using the short form of the words used while chatting, blogs etc to reduce time while using social media. Everyone has its own style of writing words/slags present in the shorthand message[1]. So, it is difficult for the computer to understand to message as it is in short form like “good morning” will be written as “gm” or “gud mrng” or “gd mrng” or “gd mrg”. Therefore, to make the computer understand that all the mentioned short forms are the same is a challenging task.

Emoticons[2] are the short form for the emotion icon, used for the pictorial representation of the emotions usually used for describing person mood, feeling etc. Emoticons are popular these days among the people in social media. It’s very common to see them in the chatting, blogs, status, text, etc. to express their feeling and representing the content of the text[3, 1].

Sentiment analysis[4] is an automated process in which a machine analyses and classifies the sentiments, emotions, conclusion etc present in the text data. In this project, we are going to understand the abbreviation, shorthand messages, emoji present in the text message using natural language processing.

2 RELATED WORK

For the determination of the abbreviations and shorthand messages, the n-gram algorithm is used which predict the given sample word on the basis of the first few continuous sequences of the letter.

It analyses the n-1 letter and then predicts the probability of the next nth letter conditional over them. We will use maximum likelihood estimation which is a natural way to estimate probability. For the given sample (n-1) letters we have to predict nth letter for that the probability equation will be

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w (w_{n-1}w_n)} \tag{1}$$

Here, the "$C(w_{n-1}w_n)$" is the number of occurrence of "w_n" in the dataset, and the "$w(W_{n-1}W)$" is the all possibilities starting with "w_n".

For example, we are taking "smiling" now its given "smilin" we have to predict the last letter. Now, for this, we will use maximum likelihood estimation which will be as follows

$$p(g|lin) = \frac{\#(ling)}{\#(lin.)} = \frac{\#(ling)}{\#(lina) + \#(linb) + \#(linc) + \dots} \tag{2}$$

Here, its 4 gram because its using 4 letters for predicting the word "smiling".

The major drawback of the n-gram algorithm is that it will predict the value on the basis of the previous n continuous sequence of the letters. If the shorthand message contains the non-continuous letters of any word then it will give less accuracy for predicting. For example "college" is written as "clg" in the shorthand message. Here the accuracy of the algorithm will be very less because the "clg" is a non-continuous word which represents "college". Hence, in this paper, we introduced a new approach to predict abbreviations and shorthand messages.

3 LITERATURE REVIEW

An early study for the prediction of the text word done by Daniel Jurafsky & James H. Martin [5] in which they have given detailed study about the n-gram model which is used for prediction of words. This model uses the maximum likelihood estimation method for estimating probability. In that, they have used Laplace smoothing, which is the simplest way to perform smoothing by adding 1 to all bigram counts before the normalization for probability. This paper has given us the motivation to analyse further and work with a new algorithm having more accuracy.

In[6], the author discusses the importance of understanding emotions which helps in the development of human's intelligence. So , an emerging technology called sentiment analysis which helps the machine to understand and categories the emotions, conclusion etc present in the sample data. To understand the emotions out of the text is a difficult task.

In[7], the authors gave a neural architecture to the model substance of the emoticons by developing the relationship between the word and emoticons. They used twitter data for analyzing the most appropriate emoticons related to tweets. They trained the model using Long Short Term Memory networks algorithm. The experimental analysis appeared that the proposed demonstrate beats, two models, to be specific BoW and SG Vector Average .

In[8], the authors have a proposed model for emoji2vec, which consists of 1661 emoji's instead of word2vector's skip gram model. Their approach is not only limited to emoji's, it directly works on the Unicode description.

4 METHODOLOGY

Our system architecture is shown in the Figure 1

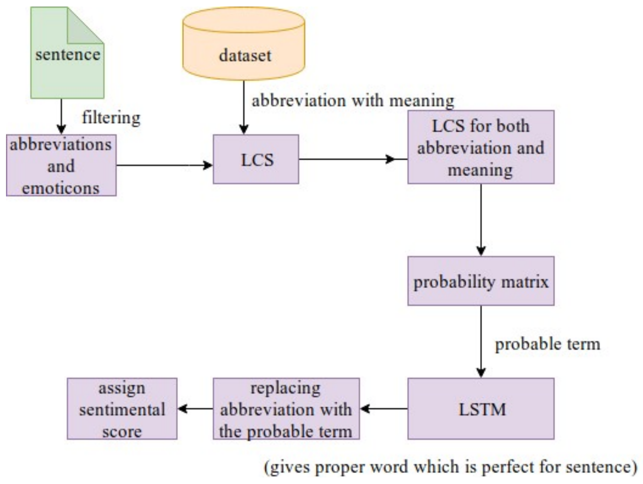

sentence
dataset
filtering
abbreviation with meaning
abbreviations and emoticons
LCS
LCS for both abbreviation and meaning
probability matrix
probable term
assign sentimental score
replacing abbreviation with the probable term
LSTM
(gives proper word which is perfect for sentence)

Fig. 1. Proposed system

4.1 Shorthand/abbreviation detections

In our methodology we are going to match the number of characters in abbreviation word with the full english word using Longest Common Subsequence.

1. We have taken I matrix of size equal to number of rows/columns which is shown below.

$$I_1 = [1], I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ldots, I_n = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

2. Apply Longest Common Subsequence:
   The algorithm for the Longest common subsequence[6] is shown below-

Algorithm 1: LCS(first_string,Second_string, m, n)

```
if both string size is zero then
    return nothing
else
    if match first_string[m-1] and Second_string[n-1] then
        return 1+LCS(first_string,Second_string , m-1, n-1)
    else
        return maximum(LCS(first_string,Second_string , m, n-1),
          LCS(first_string, Second_string, m-1, n))
    end
end
```

3. Probability calculation:
   Matched String:number of test string characters matched with string training character . Here we have to calculate LCS for both meaning and acronym after wards find the average probabilities i.e

$$Probability_x = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

   $A \cap B$ is the LCS between acronym (A) and (B). $|B|$ is length of meaning full string,$|A|$ is length of string acronym.
   Here we have to calculate LCS for both meaning and acronym after wards find the average probabilities i.e

$$AverageProbability = \frac{Probability_{Meaning} + Probablity_{Acronym}}{2}$$

4. Build probability matrix:
   After calculation of average probability, we mapped the average probability value with its acronym and meaning.
   Retrieving meaning of emoticon with similar emoticon
   (a) We have taken I matrix of size equal to number of rows/columns.

(b) We compare test string unicode with the column ,which comes equal that's our required emoji.
(c) Apply Longest Common Subsequence: We apply LCS among retrieved emoji meaning with rows(meaning) .
(d) Build probability matrix After calculation of probability we mapped probability value with its emoji unicode and meaning.In the table which one is the highest value above particular threshold value that are the required similar emoji.

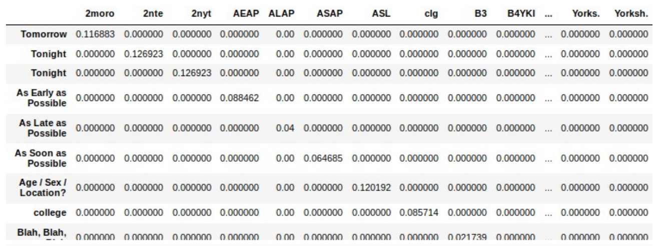
5 RESULT AND ANALYSIS

Shorthand/abbreviation detections:

We have taken here data set having 'Acronym' and 'Meaning' As two columns.In our data set we have taken 1594 rows.

Let's take example test string as tmrw so with acronym LCS's are $clg(LCS : 0), Yrs.(LCS : 1), 2mrow(LCS : 3)$ and with meaning LCS's are college(LCS:0), Years(LCS:1), Tomorrow(LCS:4).

Now, to calculate probability for meaning and Acronym i.e. (for Tomorrow) probability_meaning is 0.5 and probability_acronym is 0.6 which implies average probability is 0.55, like that for Acronym clg ,yrs. average probabilities are 0 , 0.267.



	2moro	2nte	2nyt	AEAP	ALAP	ASAP	ASL	clg	B3	B4YKI	...	Yorks.	Yorksh.
Tomorrow	0.116883	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
Tonight	0.000000	0.126923	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
Tonight	0.000000	0.000000	0.126923	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
As Early as Possible	0.000000	0.000000	0.000000	0.088462	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
As Late as Possible	0.000000	0.000000	0.000000	0.000000	0.04	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
As Soon as Possible	0.000000	0.000000	0.000000	0.000000	0.00	0.064685	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
Age / Sex / Location?	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.120192	0.000000	0.000000	0.000000	...	0.000000	0.000000
college	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.085714	0.000000	0.000000	...	0.000000	0.000000
Blah, Blah,	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.021739	0.000000	...	0.000000	0.000000

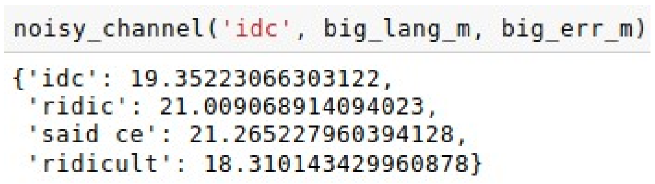
Fig. 2. Probability Matrix

If we compare n-Gram model with our model the results given below in the figure 3.

The figure 3 is the n-Gram model which cannot detect idc/IDC acronyms (I DON'T CARE) whereas our model can predict it accurately which is shown in the figure 4.

Where as in our model we are able to detect 'I DON'T CARE' acronyms with probability chances of 62.5%.

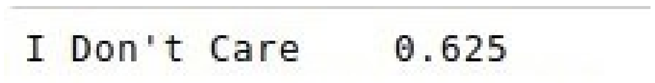
In the table 1 gives the comparison between n-gram with our proposed model. where the words are highlighted that means the system correctly recognised Acronym. Special case for our system if the acronym doesn't exist in our data base then our system cannot detect the acronym.



```
noisy_channel('idc', big_lang_m, big_err_m)

{'idc': 19.35223066303122,
 'ridic': 21.009068914094023,
 'said ce': 21.265227960394128,
 'ridicult': 18.310143429960878}
```

Fig. 3. Results for n-Gram



```
I Don't Care    0.625
```

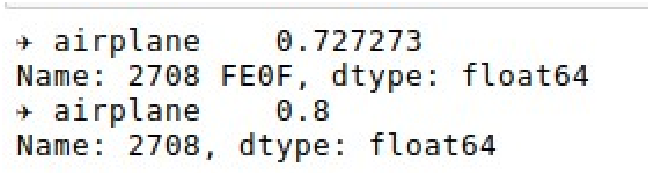
Fig. 4. Results for our approach on I don't care

Table 1. Comparison Between n-gram and Proposed model

Acronym	n-gram Model	Our proposed Model
idc	not detected	**I don't Care(0.625)**
Zeitschr.	zeits chair	**Zeitschrift (0.8)**
tmrw	**Tomorrow**	**Tomorrow(0.45)**
aeap	not detected	**As Early as Possible (0.6)**
clge	cleague, coldge	**college(0.78)**
mrng	**morning**	not detected

Emoticons meaning Retrieval :

We have given input as emoji and try to retrieve similar emoji, for example here we gave airplane emoji and first try to detect the meaning.After retrieving meaning of the emoji we try to get similar related emojis.

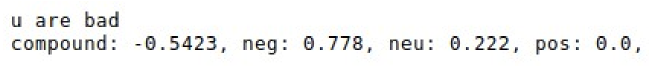


```
✈ airplane    0.727273
Name: 2708 FE0F, dtype: float64
✈ airplane    0.8
Name: 2708, dtype: float64
```

Fig. 5. Emoji detection

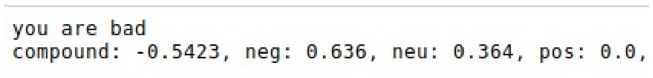
Sentimental analysis:

Using built in nltk package we able to analysis the sentiment given below in figure 6 We can see nltk is unable to detect the abbreviations due to that the value to sentiments changed in figure 6 ,it is given in figure 7 that after replacement of abbreviation the sentiment analysis result is right. Where 'neg' is the



```
u are bad
compound: -0.5423, neg: 0.778, neu: 0.222, pos: 0.0,
```

Fig. 6. Sentimental analysis by using nltk



```
you are bad
compound: -0.5423, neg: 0.636, neu: 0.364, pos: 0.0,
```

Fig. 7. Sentimental analysis using our approach

percentage of word/sentence is negative.'neu' is the percentage of word/sentence is neutral and 'pos' is the percentage of word/sentence is positive.compound says about the complexity of sentence.

6 CONCLUSION

In this work, we have used LCS algorithm and LSTM for the word and shorthand message to predict the meaning associated with it. Our model can work accurately for prediction of abbreviations which can't done by the n-gram algorithm, it will predict the value on the basis of the previous n continuous sequence of the letters. If the shorthand message contains the non-continuous letters of any word then it will give less accuracy for predicting. Meaning of emoticons similar emoji can be found by this method. One drawback of our model is that, the processing speed becomes slow if take long strings for the prediction.

7 FUTURE WORK

This project can be extended by using by Bidirectional LSTM. LSTM can be use to predict next possible word of the sentence. Which abbreviation will be replace by correct meaning full word based upon sentimental analysis that's a big questions for machine but combination of Bidirectional LSTM and our proposed model can be a good abbreviation replacer system.

References

1. Li, X., Yan, R., Zhang, M.: Joint emoji classification and embedding learning. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, Springer (2017) 48–63
2. Hogenboom, A., Bal, D., Frasincar, F., Bal, M., De Jong, F., Kaymak, U.: Exploiting emoticons in polarity classification of text. J. Web Eng. **14**(1&2) (2015) 22–40
3. Tang, D., Wei, F., Qin, B., Zhou, M., Liu, T.: Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In: Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers. (2014) 172–182

4. Zhao, Y.Y., Qin, B., Liu, T., et al.: Sentiment analysis. Journal of Software **21**(8) (2010) 1834–1848
5. Cambria, E.: Affective computing and sentiment analysis. IEEE Intelligent Systems **31**(2) (2016) 102–107
6. Sorokin, A.: Using longest common subsequence and character models to predict word forms. In: Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology. (2016) 54–61
7. Barbieri, F., Ballesteros, M., Saggion, H.: Are emojis predictable? arXiv preprint arXiv:1702.07285 (2017)
8. Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., Riedel, S.: emoji2vec: Learning emoji representations from their description. arXiv preprint arXiv:1609.08359 (2016)