International Journal of Engineering & Technology, 7 (2.20) (2018) 97-100

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Multi-class Emotion AI by reconstructing linguistic context of words

K. Sripath Roy [1], Farhaan Ahmed Shaik [2*], K. Uday Kiran [2], M. Naga Teja[2], Subhani Kurra [2]

[1] Asst Professor Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502
[2] Student, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502
**Corresponding author E-mail: farhaanfsk@gmail.com.*

Abstract

In today’s technological world, Social networking websites like Twitter, Instagram, Facebook, Tumblr, etc. play a very significant role. Emotion AI is about dealing, recognizing and analyzing sentiments or opinions conveyed in a person’s text. In particular Emotion is most frequently called Sentiment analysis. It helps us to understand the people’s point of view. A vast amount of sentiment rich data is produced by Social networking websites in the form of posts, tweets, statuses, blogs etc. Some users post reviews of certain products in social media which influences customers to buy the product. Companies can use such review data analyze it and improve the product. Sentiment analysis of Twitter is troublesome correlated to other social networking websites because of the existence of a lot of short words, misspellings and slang words applying emotion analysis to such data is more challenging. We have classified the sentiment into 5 categories. Machine learning strategies are preferred mostly for analyzing emotion AI. We have used neural network model word2vec with TF-IDF approach to predict the sentiment of the tweet.

Keywords: *Emotion AI, Machine learning, Sentiment analysis, TF-IDF, Word2vec.*

1. Introduction

Technology is developing day-by-day because of the Internet. The Internet is a worldwide communication network with this, usage of social websites like Twitter, Facebook, Instagram, Tumblr etc. have increased. In fact, social websites have become a famous place where everyone expresses their opinion about anything i.e. either it may be a product or article or society issues. Everyone has right to express their feeling and many of them think a social website is the best place to express their feelings in the form of the social networking blogs, Tweets, posts, statuses etc.

Sentiment Analysis or Emotion AI is a mechanism for ‘computationally’ resolving or recognizing if certain text is neutral, negative or positive. Sentiment analysis is again called as **Opinion mining** that is computing the sentiment or opinion of the person. Sentiment analysis is used to establish the person's opinion towards a product or problem with the help of variables such as emotion, tone and context, etc. Business corporations value this approach to compute and investigate the public opinion of their products and company to improve the customer satisfaction towards them. Business corporations further use this opinion mining approach to collect critical opinion about a new product and identifying the problems in it. Sentiment analysis is used by the major multinational companies to obtain the impression on their products and with the help of that, they can design their business strategies. Twitter is one of the platforms where people post the opinion on major real-world subjects, current affairs, etc. The social media websites like Facebook, Twitter and Instagram produce trillions of megabytes of data 24/7. This vast amount of data is interpreted to figure out the sentiment of people on various problems. Although a lot of research is done in sentiment analysis our developed model is one of its first kind.

2. Proposed Techniques

2.1 Euclidean Distance

In order to calculate the distance between two points, Euclidean is used. Euclidean distance will play a significant role to calculate the distance between words. Euclidean distance is used in many of classification algorithm such as K-Nearest Neighbour, Minimum Distance Classifier, TF-IDF etc. let p1 at (x1, y1) and p2 at (x2, y2), in a plane then the Euclidean can be calculated with the help of Pythagoras theorem by:

$$\text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2.2 TF-IDF

Tf-idf is an acronym for term frequency-inverse document frequency, it’s a way to grade the significant terms (or "words") in a document based on how often these terms appear across the whole document and across multiple documents and the tf-idf weight usually used in data retrieval and text data mining.

Computing sum of tf-idf for each term is one of the simplest approaches.

$$TF - IDF(t) = [\ TF(t)\ \times\ IDF(t)]$$

2.2.1 Term Frequency (TF)

It computes how often a word or term appears in a record. Since every record is divergent in length, it is possible that the term appears more times in some documents than others. Term frequency can be calculated by how frequent a term t occurs in a record, normalized by dividing by the total number of terms in that record.

$$TF(t) = \left[\frac{(Number\ of\ times\ term\ t\ appears\ )}{(Total\ number\ of\ terms)}\right]$$

2.2.2 Inverse Document Frequency (IDF)

It computes how common a word is among all records, at the same time measuring Term frequency, every word is scrutinized as equally essential. Despite, there are certain common words like "the", "for", "is", "of" and "that" which appear in many documents very frequently but have little significant. Therefore we compute IDF by:

$IDF(t) = log_e \left[\frac{(Number\ of\ times\ term\ t\ appears\ )}{(Total\ number\ of\ terms)}\right]$**.3**

WORD2VEC model

Word2Vec is a cluster of models which helps derive associations between a term and its contextual terms or words. The cluster models are hollow, two-layer neural networks that are schooled to reestablish linguistic contexts of words. The two important model architectures inside are Skip-grams and CBOW. Word2vec model can be built with the help of Tensor flow, it is an open source machine learning framework developed by Google for research purpose.

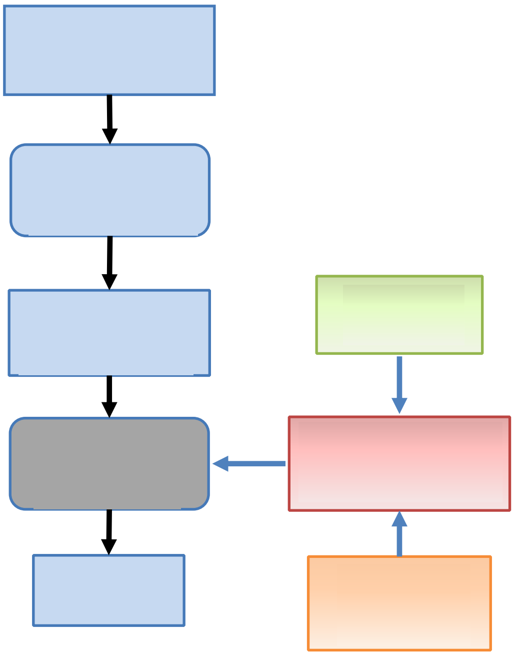
2.3.1 Skip-gram

In Skip-gram model, a center word and a window of context (neighbor) words are chosen and context words are predicted out to some window size for each center word. So, the model goes to outline a probability distribution i.e. probability of a word occurring in the context given a center word and vector representations is chosen to maximize the probability.

2.3.2 Continuous Bag of Words (CBOW)

In Actual terms, this is a mirror of skip-gram. In CBOW, we predict center word by summing vectors of surrounding (Neighbour) words. Essentially, we begin with small random initialization of word vectors. In Word2vec, we have a large matrix with each row for the "words" and columns for the "context". CBOW is not sequential and does not have to be probabilistic.

3. Flow of process



4. Implementation

4.1 Data Collection

Collection of data is the major task in sentiment analysis. Twitter data is abundantly available on twitter official website. Twitter provides a standard search API for research purpose. Access_token_ secret, Access_token, Consumer_secret and Consumer_key is required to access the Twitter API, to obtain these keys an application is to be created in dev.twitter.com. We have developed a simple python script with the Twitter API access keys and used it to download the specific tweets of a topic from the Twitter API to form the dataset. This raw dataset obtained through python script is cleaned and processed for prediction. We have downloaded tweets on airlines since they are more suitable.

4.2 Data Pre-processing

In Sentiment analysis, preprocessing of data plays a vital role as data always may not be in the exact (meaningful) form for analysis. Twitter data preprocessing is more significant as it consists of slang words, misspellings, #tags, @tags, hyperlinks and shortcut words. So we use various processing techniques like:

I. Removing duplicates i.e. retweets and repeated tweets that are downloaded.
II. Replacing the Null values with Zeros.
III. Tweet of length less than 3 are deleted.
IV. Tags and hyperlinks have been eliminated.
V. Stemming or lemmatization to obtain the base word.
VI. Removing stop words from each tweet.

Caution is required when removing stop words and lemmatization because some words like Not, Very, Limit etc. make sense in many statements. For overcoming this problem we use bigrams or trigrams instead of unigrams. Sometimes with Stemming or lemmatization, the base word obtained is not relevant to the tweet, such problems should be specifically checked.

4.3 Tweet Annotation

Generally, the output of sentiment analysis of any statement falls into three categories, they are positive, negative and neutral. Since the chosen dataset is on Airlines review tweets, we add two more essential categories question and suggestion. Since new categories added auto-annotation of emotion is not possible. So, each tweet is manually annotated. Tweets sometimes consist of both positive and negative or a negative and question but general sentiment analysis only assigns a single outcome to the tweet. As an improvement to this, multiple assignments to each tweet is done, For example:

1) *"@united I have such a love-hate relationship with you. Some days you're good, the others you are so terribly awful it's saddening"*. We annotate such tweet as both **Positive** and **Negative**.

2) *"@VirginAmerica, you're doing a great job adding little luxuries/aesthetics that improve the air travel experience. Thank you. Keep it up !"*. We annotate such tweet as both **Positive** and **Suggestion**.

3) *"@SouthwestAir, been on hold for an over an hour now - when can we expect some customer service? #disappointed"*. We annotate such tweet as both **Negative** and **Question**.

4.4 TF-IDF word2vec model

Word2vector is a super-powerful representation of words and it somehow catches the semantic relationship in d-dimension vector. Word2vector gives Dense network i.e. most of the cells are non-zero.Word2vector learns the Data corpus through Word-Content or context. Suppose $⍰_1$, $⍰_2$, $⍰_3$, $⍰_4$, $⍰_5$ are the words in a corpus then in order to compute $⍰_⍰$ i.e. $⍰_⍰$ occurs in the context of the set of 5 words chosen. Let $⍰_1$, $⍰_2$, $⍰_3$, $⍰_4$, $⍰_5$ are the words in a tweet. So, We form 300 dimension vector for every word from Google's model and we add all these vectors (V_1+V_2+V_3+V_4+V_5) and divide that with total no of words let it be K ((V_1+V_2+V_3+V_4+V_5)/K). With this, we get a single vector of size 1*300 dimension for every tweet. For this algorithm, we need a large data corpus i.e. in billions or millions.

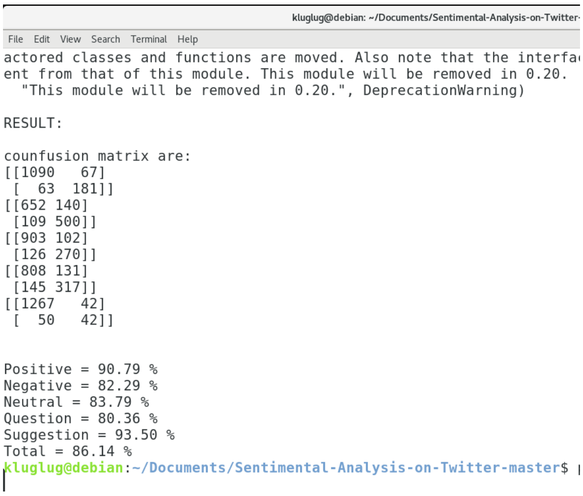
Since only limited data is available it is not feasible to build a word2vec model of our own. Instead, it is preferred to use Google's word2vector model. It consists of 3 billion words and each word is associated with 1X300 size vector. By using this model we create a combined model of **"word2vec"** and **"TF-IDF"** to form **TF-IDF word2vec** model.

5. Training & Testing dataset

After annotation of the dataset, we break the dataset into a couple of parts, Training data and Testing data. The number of tweets in the training data should be always greater than the testing data for the better result. Each tweet from test dataset is matched with a similar tweet from train dataset with the help of the constructed model "TF-IDF Word2vec" and various distance calculation algorithms like Euclidean distance, Manhattan, Minkowski, Hamming, cosine similarity and cosine distance. It is found that Euclidean distance is far better than others after multiple trails on the airline's dataset. We predict the class variables by using the above technique i.e. Euclidean distance and assign the same values as the near tweet consists of the test tweet. By using these similarity-based techniques we predict the sentiment of each user data or tweet.

6. Result

In the dataset of 7000 Airline review tweets, 80% of the tweets are trained and 20% are tested or predicted. Below is the bar-chart (fig.3) which shows the overall number of tweets that are predicted correctly and the number of tweets that are predicted wrong. These prediction values are obtained by adding the principal diagonal elements of the obtained confusion matrix. Principal diagonal elements indicate correct prediction and other diagonal indicate wrongly predicted values. Percentage of prediction can be obtained by dividing correct predicted value with number of tweets that are tested * 100. We have obtained five confusion matrices for each emotion positive, negative, neutral, question and suggestion respectively. The combination of TF-IDF and word2vec resulted in an 86.14% classification model accuracy.


kluglug@debian: ~/Documents/Sentimental-Analysis-on-Twitter-
File Edit View Search Terminal Help

```
actored classes and functions are moved. Also note that the interfa
ent from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)

RESULT:

counfusion matrix are:
[[1090   67]
 [  63  181]]
[[652 140]
 [109 500]]
[[903 102]
 [126 270]]
[[808 131]
 [145 317]]
[[1267   42]
 [  50   42]]


Positive = 90.79 %
Negative = 82.29 %
Neutral = 83.79 %
Question = 80.36 %
Suggestion = 93.50 %
Total = 86.14 %
kluglug@debian:~/Documents/Sentimental-Analysis-on-Twitter-master$
```

Fig.2 Output in Terminal.

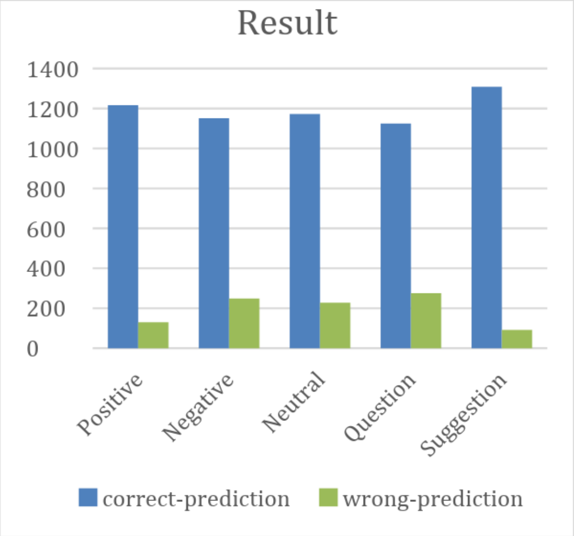


Fig.3 Result showing the prediction values

7. Conclusion and Future Scope

In this paper, we have implemented Sentiment analysis or Emotion AI with two-layered neural network model Word2vec and deep learning model TF-IDF. We have added two more opinions Question and suggestion with the basic sentiments to form a 5 vector model and have manually annotated 7000 tweets and with the help of Google's predefined model which is a set of 3 billion words associated with 1X300 size vector. Word2vec input is a data corpus i.e. dataset and the output it produces is set of vectors for the words in the dataset. The accuracy can be improved if more and more data is trained.

References

[1] Y. Singh, P. K. Bhatia, and O.P. Sangwan, "A Review of Studies on Machine Learning Techniques", International Journal of Computer Science and Security, Volume (1): Issue (1), pp. 70-84, 2007.

[2] E. Boiy, P. Hens, K. Deschacht and M. Moens, "Automatic sentiment analysis in on-line text", 11th International Conference on Electronic Publishing, vol. 349360, 2007.

[3] Vincentius Riandaru Prasetyo, Edi Winarko. "Rating of Indonesian sinetron based on public opinion in Twitter using Cosine similarity", 2016 2nd International Conference on Science and Technology-Computer (ICST), 2016

[4] R. Feldman, "Techniques and Applications for Sentiment Analysis", Communications of the ACM, Vol. 56 No. 4, pp. 82-89, 2013.

[5] J. Kamps, M. Marx, R. Mokken and M. De Rijke, 'Using wordnet to measure semantic orientations of adjectives', 2004.

[6] Gautam, Geetika, and Divakar Yadav. "Sentiment analysis of twitter data using Machine learning approaches and semantic Analysis", 2014 Seventh International Conference on Contemporary Computing (IC3), 2014.

[7] Neethu, M. S., and R. Rajasree. "Sentiment Analysis in twitter using machine learning techniques", 2013 Fourth International Conference on Computing Communications and Networking Technologies (ICCCNT), 2013.

[8] E. Boiy, P. Hens, K. Deschacht, and M.-F. Moens, "Automatic sentiment analysis in on-line text," in Proceedings of the 11th International Conference on Electronic Publishing, pp. 349-360, 2007.

[9] Mondher Bouazizi, Tomaki Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter", IEEE Xplore journal volume 5, 2017.

[10] https://www.tensorflow.org/tutorials/word2vec.

[11] https://www.tfidf.com/.