International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 05 | May -2020 ISSN: 2582-3930

Twitter Sentiment Analysis

Ashish Dalmia[1], Ananya Upadhyay[2]

Department of Computer Science and Engineering, RKGIT, Ghaziabad

---

Abstract -Sentiment Analysis aims at gathering information about public opinion for any product. It is the process of analysing the sentiment of people and extracting subjective information. We take in consideration the twitter data as it provides us with an extremely valuable insight into human opinions as well as the new challenging Big Data problem. This problem includes the processing of massive volumes of streaming data and getting insights of the short messages. These insights help various groups of people or organizations to understand the customer opinion towards their product. We have taken the default twitter dataset from kaggle dataset. This paper describes a deep learning sentiment analysis of tweet. The sentiment analyser can classifies the tweet into two categories- positive or negative, based on various aspects. As various other research papers have already published, this paper tends to provide an aspect that RNN based LSTM can provide better results.

Keywords:Natural Language Processing, Twitter Sentiment Analysis, RNN, LSTM, Twitter Dataset.

1. INTRODUCTION

The social media platform and micro-blogging sites have evolved over the period of time and with this evolution they now provide a variety of information. This is because of the nature of these sites. The people all around the world, share their opinions, real-time emotions, reviews and feelings on a variety of topics, discuss current issues, express their sentiments related to a company or a product. Often these companies study the user reactions and emotions and reply to the users on their micro-blogs. People around the world end up showing their positive or negative sentiments for various political and regional issues which can also help in nation building. People have different views about a single product or topic. One challenge is to build a technology to detect and summarize an overall sentiment. The sentiments or opinions are classified into three types:- Positive sentiment,negative sentiment and neutral sentiment.

We will be using a dataset formed of collected messages from Twitter. Twitter contains a really sizeable amount of very short messages created by the users of this micro-blogging platform. The contents of the messages may vary from personal opinions, feelings or thoughts to public statements. This paper implements the method to distinguish the tweets based on the sentiments of the tweet. This is also useful as it analyses a large chunk of data at a time and also removes the intervention of manual labour consuming more time and is not cost effective and accurate. Generally the tweets on micro-blogging sites like twitter, may not be in a thoughtfully composed form of review but still on analysing the tweets, companies gather additional feedbacks about their products, from the users.

As there are many solutions proving that the emoticons on the tweets had better represent the sentiment, we have also taken into account these emoticons for better predictions.

There are also Twitter APIs, which can help to extract large amount of tweets with emoticons in them, but we are currently using kaggle dataset for generalization.

2. Defining Sentiment

Sentiment can be undserstood to be the personal feelings, views or opinions towards anything. These feelings can be positive or negative. Many times it may not be necessary that every tweet generally gives a positive or negative sentiment about a person, product or place. Tweets differ depending on the context so twitter also uses a unique sorting algorithm to classify a tweet for better utilization. We have also ignored the neutral tweets. The training set and test dataset consist of positive and negative sentiments. A tweet can be sentimentally neutral like- The 10 years old Greek gymnast Dimitrios Loundras is the youngest ever recorded athlete at an olympic games. Neutral tweets prediction is currently a challenge which has not yet been overcome.

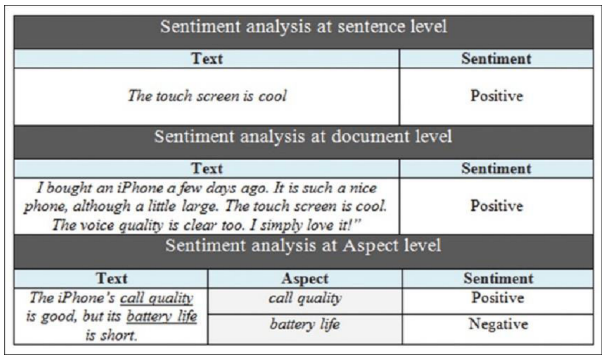
The sentiment mining or the opinion mining, can be done in three levels:

a) Document level - the entire document is categorized as positive, negative or neutral.

b) Sentence level - the sentiment of each sentence is categorized as positive , negative or neutral.

c) Aspect level or featurelevel - more commonly known as ”perspective level ssessment grouping”, this level comprises of categorizing the sentiment of sentences/documents on the basis of certain aspects.

The following image gives examples of the three levels of sentiment analysis.



Sentiment analysis at sentence level	
Text	**Sentiment**
The touch screen is cool	Positive
Sentiment analysis at document level	
Text	**Sentiment**
I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is clear too. I simply love it!"	Positive

Sentiment analysis at Aspect level		
Text	**Aspect**	**Sentiment**
The iPhone's call quality is good, but its battery life is short.	*call quality*	Positive
	battery life	Negative

$$\mathbf{c}_{\text{pooled}} = \begin{bmatrix} \text{pool}(\alpha(\mathbf{c}_1 + b_1 * \mathbf{e})) \\ \dots \\ \text{pool}(\alpha(\mathbf{c}_n + b_n * \mathbf{e})) \end{bmatrix},$$

Fig -1: Different levels of sentiment analysis

3. About the Model

We have used a deep convolutional neural network to implement sentiment analysis. The architecture of this model is similar to deep learning system presented in [1,2] that has been published earlier.While training a convolutional neural network it becomes very difficult task to rely on multiple lexicons. It also becomes even more useless when we have a huge amount of data in an unlabelled manner.So we have implemented the training process in through three steps. Providing good initialization parameters can benefit us to increase the accuracy of the trained model-

a) Word Embedding are initialized on a neural network

b) Using CNN, refine the embedding creating a large corpus

c) The word embedding initialize the network.

We have applied the deep learning models on two tasks-

a) Phrase Level

b) Message level

4. NETWORK ARCHITECTURE

Our network is mainly based on the paper [2,3] which was published to feature sentiment classification task. The Neural network only consists of a single layer rather than being based on several layers as in the paper[2,3].

In the following we have also described the main components of the network : sentence matrix, activation, convolutional, pooling and softmax layers.

4.1 Input

The input to the models are tweets that are groups of words arranged in a sequence :[wi,…ws]. Words are represented by distributional vectors w 2 R1_d looked up in a word embedding matrix -: W ∈ 2 R^ (d x |V|). These matrices formed by simply concatenating is embedding of all words in V. For convenience and for the ease of look-up operations in W, words are mapped into indices 1; : : : ; |V|.

4.2 Convolution Map

The convolution layer is used to extract pattern, like discriminative words sequence found within the tweets that remains common throughout the training instance.

4.3 Activation units

To enable the learning of non-linear decision boundaries, each convolutional layer is typically followed by a non-linear activation function. The most common choices of activation functions are: sigmoid (or logistic), hyperbolic tangent tanh, and a rectified linear (ReLU) function defined simply as max(0; x) to ensure that feature maps are always positive. We use ReLU in our model since, as shown in [6], it speeds up the training and at times produces more accurate results.

4.4 Pooling

The output from the convolutional layer (passed through the activation function) is then passed to the pooling layer, whose goal is to aggregate the knowledge and reduce the representation. The result of the pooling operation is:

where c_i is that the ith convolutional feature map with added bias (the bias is added to every element of c_i and e is a unit vector of an equivalent size as c_i) and is then passed through the activation function .

The most popular choices for the pooling operation are: max and average pooling. Recently, max pooling has been generalized to kmax pooling [2], where instead of a single max value, k values are extracted in their original order. We use max pooling in our model, which simply returns the value that is maximum. It operates on the columns of the feature map matrix C returning the largest value: pool(ci) :R1(jsj+m□1) ! R (also shown schematically in Fig. 1).

The convolutional layer utilizing the activation function and the pooling layer acts as a non-linear feature extractor. Given that multiple feature maps are utilized in parallel to process the input, deep learning networks are then able to build rich feature representations of the input.

4.5 Softmax

The output of the penultimate convolutional and pooling layers x is passed to a fully connected softmax layer. It computes the probability distribution over the labels:

$$\begin{aligned} P(y = j|\mathbf{x}, \mathbf{s}, \mathbf{b}) &= \text{softmax}_j(\mathbf{x}^T\mathbf{w} + \mathbf{b}) \\ &= \frac{e^{\mathbf{x}^T\mathbf{w}_j + b_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T\mathbf{w}_k + b_k}}, \end{aligned}$$

where wk and bk is the weight vector and bk is the bias of the k-th class.

4.6 Phrase-level sentiment analysis

To perform the phrase-level sentiment analysis, we provide the network with an additional input sequence specifying the location of the target phrase in a tweet. The elements are encoded using merely two word types: the tokens spanning the phrase to be predicted are encoded with 1s and then all the others with 0s.

Every word type is related with its own embedding. So, when tackling the phrase-level sentiment classification, we form a sentence matrix S as follows: for every token in a tweet, we have to pick up its corresponding word embedding in the word matrix W, and the embedding for one among the two word types. Therefore, the input sentence matrix is augmented with an additional set of rows from the word type embeddings.Other than that, the architecture of our network remains unchanged.

5. Approach to implement and train the network

The Convolutional Neural Network is easy to train but training the network on small dataset can be tricky and can lead to overfitting. In the following, we describe the various steps to train the network.

5.1 Network Parameter & Training

We have Stochastic Gradient descent to train the neural network and backpropagation to adjust the weights and calculate the gradient decent.

5.2 Regularization

As we described earlier training network on small and medium dataset can be extremely tricky and can lead to overfitting. To mitigate this problem of overfitting we have suggest to use the twelve regularisation with the cost function . Regularisation prevents overfitting by smoothening the regression lines.

Dropout is also used as another parameter, as dropout feature, co-adaptation by setting to zero (dropping out) a portion of hidden units during the forward phase when computing the activation at the softmax output layer.

5.3 Initialization

Convolutional Neural Network are trained carefully to avoid optimum minima. Therefore it is necessary to optimize to successfully train the network.

The word matrix W is the largest parameter on behalf on which the network is to be trained. so it is necessary to feed the network with high quality embedding like word2vec neural model which help the network to learn the network to lean on the embeddings.

We have used the dataset from Stanford library which contains more than 1 million tweets data. We have also performed minimal preprocessing , tokenising the tweets, normalizing the urls.

After performing this step still the network is unable to understand the sentimental values of the tweets and is required to make them understand the tweets. Hence we use a distant supervision approach [1] to further refine the embeddings. (We used a large number of positive tweets to link them with the tweets present in the main datset).

Finally the obtained parameters (theta) are used to initialize the network.

6. Literature Survey

Sentiment analysis is been handled as a Natural Language Processing task at many levels of granularity. In the beginning it was a document level classification task (Turney, 2002; Pang and Lee, 2004), then it began to be handled at the sentence level (Hu and Liu, 2004; Kim and Hovy, 2004) and lately at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Users now post real time reactions to and opinions about “everything”,on microblogging sites like twitter. This poses newer and different challenges. Some of the former and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to obtain sentiment information. It also uses the fact that tweets ending in positive emojis represent positive sentiment and negative emojis represent negative sentiment. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). With regard to feature space, they try a Unigram, Bigram model in addition with parts-of-speech (POS) features. They observe that the unigram model outperforms all other models. The bigrams and POS features especially do not support. Following a similar distant learning paradigm, Pak and Paroubek (2010) collect data. They perform a special classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in a similar way as Go et al. (2009). For objective data they crawl twitter accounts of popular newspapers like “New York Times”, “Washington Posts” etc. They report that POS and bigrams both help (in conflict with the results presented by Go et al. (2009)). Both of these approaches, were primarily based on ngram models.Further, the data used by them for training and testing is biased as it is collected by search queries. Unlike the data collected using particular queries, the information collected by us is attained from a random sample of streaming tweets. The size of our hand-labeled data helps us to do cross validation experiments and examine the variance in performance of the classifier across folds. An additional notable effort for sentiment classification on Twitter data is by Barbosa and Feng (2010). They use polarity predictions from three websites as noisy labels to train a model and use thousand manually labeled tweets for tuning and another thousand manually labeled tweets for testing. Although they do not remark about how they gather their test data. They suggest the use of syntax features of tweets such as retweet, hashtags, link, punctuation and exclamation marks in addition with features like prior polarity of words and POS of words. We will increase their approach by using real valued prior polarity, and then by combining prior polarity with POS. The results may exhibit that the features that intensify the performance of our classifiers to the greatest extent are features that merge prior polarity of words with their parts of speech. The tweet syntax features help but only marginally. Gamon (2004) carry out sentiment analysis on feedback data from Global Support Services survey. Our aim is to observe and examine the role of linguistic features. They execute extensive feature analysis, feature selection, and reveal that abstract linguistic analysis features adds to the classifier accuracy.

7. EXPERIMENTS AND EVALUATION

7.1 Data and setup

We test our model over two subtasks from Semeval-2015 Task 10: the first being at phrase-level (subtask A) and the second at message-level (subtask B). Train and dev from Twitter’13 are used for training and Twitter’13-test is used as a validation set. The other datasets are used for testing, whereas Twitter’15 is used to establish the official ranking of the systems.

For assessment we use the official scorers from Semeval 2015, which compute the average between F-measures for the positive and negative classes.

To pre-train the weights of our network, we use a large unsupervised corpus containing 50M tweets for training the word embeddings and a 10M tweet corpus for distant supervision. The latter corpus was built similarly to [1], where

tweets with positive emoticons, like ':)', are assumed to be positive, and tweets with negative emoticons, like ':(', are labeled as negative. The number of positive and negative tweets in the dataset is same. The parameters of our model were (selected on the validation set) as follows: the width m of the convolution filters is set to 5 and the number of convolutional feature maps is 300. We use ReLU function and a simple max-pooling. The dimensionality of the word embeddings d is set to 100. For the phrase-level subtask the size of the word type embeddings, that encode tokens that span the target phrase or not, is set to 10.

7.2 Evaluation

We tested our model on different aspects first being the word level and second being the phrase level. We have mentiond the accuracy of the model and the confusion matrix.

The accuracy of the model was 79.11% when tested on phrase level.

The Confusion matrix has been mentioned below

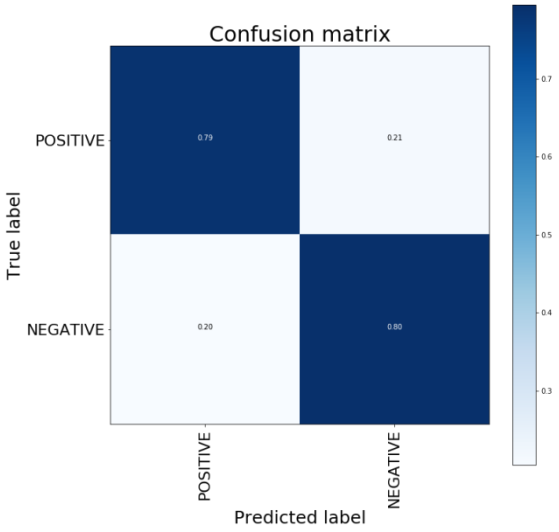


Fig -2: Confusion Matrix

8. CONCLUSIONS

We described our deep learning approach to sentiment analysis of tweets for predicting polarities at both message and phrase levels. We give a detailed description of our 3-step process to train the parameters of the network that is the key to our success. The resulting model sets a new state-of-the-art on the phrase-level and is 2nd on the message-level subtask. Considering the average rank across all test sets our system is 1st on both subtasks. Our network initialization process includes the use of distant supervised data (noisy labels are inferred using emoticons found in the tweets) to further refine the weights of the network passed from the completely unsupervised neural language model. Thus, our solution successfully combines together two traditionally important aspects of IR: unsupervised learning of text representations (word embeddings from neural language model) and learning on weakly supervised data. In the future we plan to apply deep learning approach to other IR applications, e.g., learning to rank for Microblog retrieval and answer reranking for Question Answering.

REFERENCES

[1] A. Go, R. Bhayani, and L. Huang. Twitter sentimentclassification using distant supervision. In CS224N Project Report, Stanford, 2009.
[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In ACL, 2014.
[3] Y. Kim. Convolutional neural networks for sentence classification. In EMNLP, 2014.
[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, 2013.
[5] S. M. Mohammad, S. Kiritchenko, and X. Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets.In Semeval, 2013.
[6] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.
[7] J. W. Ronan Collobert. A unified architecture for natural language processing: deep neural networks with multitask learning. In ICML, 2008.
[8] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. CIKM, 2014.
[9] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In WWW, 2014.
[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.
[11] S. M. M. Xiaodan Zhu, Svetlana Kiritchenko. Nrc-canada-2014: Recent improvements in sentiment analysis of tweets, and the Voted Perceptron. In SemEval, 2014.
[12] M. D. Zeiler. Adadelta: An adaptive learning rate method. CoRR, 2012.
[13] Aliaksei Severyn, Alessandro Moschitti: Twitter Sentiment Analysis with Deep Convolutional Neural Networks