Comparison of VADER and LSTM for Sentiment Analysis

Adarsh R, Ashwin Patil, Shubham Rayar, Veena K M

Abstract*— Sentiment analysis is one of the trending topics at present. It has a vast scope from analysing the mood of the person based on his tweet, to predicting the stock prices. But this field is quite challenging. It is not easy to make a machine understand what exactly the person is saying. In this paper, we are going to demonstrate two different methods that can be used in sentiment analysis and its comparison. The two methods used in this paper are: i) VADER-Valence Aware Dictionary for sEntiment Reasoning ii) LSTM model (Long Short-Term Memory). VADER uses a lexicon-based approach, where the lexicon contains the intensity of all the sentiment showing words. The intensities are fetched, the sentiment score is calculated and based on this sentiment score, the review is classified as either positive or negative. We used VADER from NLTK module of python for our study. Recurrent Neural Network has proved its results in a variety of problems like speech recognition, language modelling, and translation. We used LSTM which is an extension of RNN for our study. LSTM networks are very effective for sequential data like texts because they can relate the context of the sentence very well. We preferred LSTM over RNN as LSTM supports Long-term dependency which will help us predict our reviews better. We implemented the LSTM model using keras.

***Index Terms*— GloVe, Lexicon approach, LSTM, Sentiment Analysis, VADER**

1. INTRODUCTION

Sentiment analysis is a very powerful tool. It can be used to find the tone of tweets in Twitter, used in recommender system for suggesting articles based on reader’s interest, classify reviews as positive or negative. It can also be used to predict stock prices, based on conflicting tweets and comments. Analysing sentiments is very important to all industries. It is not easy to analyse the sentiments in the text. Many a time the words used in a text may have a spelling error or maybe a slang word which model cannot identify. The reviews or texts must be cleaned before they are analysed. In this paper, we will analyse the dataset of mobile reviews using two different approaches. Firstly we began with the VADER, which is a lexical based approach. It has a dictionary of sentiment words with their corresponding intensities, which range between -4 to +4. For Example, the word good has a polarity score of +1.9 and nice has polarity score of +1.8. VADER not only classifies the review as positive or negative, but it also tells us the intensity of positiveness or negativeness in the text. But the problem with VADER is that it can only find sentiments of the words that are present in the lexicon. Any new slangs, if used in a review will not have any effect on classifying the reviews as those slang words are not part of the lexicon and hence do not have polarity.

LSTM is an extension of RNN. It uses a machine learning approach to classify the reviews as positive or negative. The best thing about LSTM model is that it can relate the context, which cannot be achieved by other neural network models. When we talk about machine learning, we talk about mathematical equations. But what about the text then? To use texts in the machine learning model, it should be converted to appropriate word vectors. We have used GloVe for getting a vector representation of the word. These Vectors are then passed to models for training the model and also to obtain the prediction. This approach is quite different from the lexical approach where the words are mapped to the lexical dictionary; corresponding intensities are obtained and then applied through a formula to obtain the overall sentiment score.

II. TASK DEFINITION

TASK A: Study and analyse mobile review dataset using VADER (lexical based approach)

TASK B: Study and analyse mobile review dataset using LSTM (Machine learning approach)

III. RELATED WORK

Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative or neutral. The most common use of Sentiment Analysis is that of classifying a text to a class. We started our study from work presented by [1]. In this paper, the author explained about various sentiment lexicons including VADER. The broad overview of all existing work was presented by [2], in their survey. The authors describe existing techniques and approaches for opinion-oriented information retrieval. Early works in sentiment analysis started with the seminal work and considered review as bag-of-words, and focused on classifying them as positive, negative, or neutral using classifiers. Later works developed more sophisticated features based on phrasal and dependency relations, narratives, perspectives, lexicons.

The [3] provides a detailed survey of the existing sentiment analysis methods. It also discusses the detailed overview of the sentence-level sentiment analysis available in the literature. An extension of LSTM is used [4] to perform target

Revised Version Manuscript Received on March 08, 2019.

Adarsh R, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India (E-mail: adarsh.ravikumar7@gmail.com)

Ashwin Patil, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India (E-mail: ashwin.mpatil98@gmail.com)

Shubham Rayar, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India (E-mai: shubham.rayar@gmail.com)

Veena K M, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. (E-mail: veena.gv@manipal.edu)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

dependent sentiment classification and it found that the extension LSTM outperforms standard LSTM, adaptive recursive neural network, feature based SVM, lexicon enhanced neural network.

3.1 Machine Learning Approach

Classification model of machine learning can also be used to classify the text into binary class (positive or negative) and multi-class (positive, negative and neutral). Two Popular models that were for Sentiment analysis is SVM and Naive Bayes.

A support vector machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space, which can be used for classification. An SVM training algorithm builds a model that assigns new examples to one category or the other based on the support vector.

Naive Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The main problem with these models is that it cannot relate to the context of the text. So, we decided to choose a model that is sequential and can relate the context. So, we choose LSTM for our study.

IV. MOTIVATION

Sentiment Analysis is growing wider day by day. This field has application in various domains like Business Intelligence. It helps the company identify why people do not prefer their product and how can they improve it? Apart from this, we humans are subjective creatures and our opinion matters a lot. Being able to interact with people on that level, will help promote business. To better understand people and their opinions was our main motivation.

V. PROBLEM DOMAIN

In this paper, we are focusing on the review analysis. This comes under the domain of text analysis and Sentiment analysis.

VI. PROBLEM DEFINITION

Comparison of VADER and LSTM models (Lexicon based model vs. Machine learning model) for classification of reviews as either positive or negative, against a mobile review Dataset. The reviews are gathered from [6].

VII. INNOVATIVE CONTENT

For a given class of review, Mobile reviews, comparing and determining the best approach among LSTM and VADER. The best performing approach could be for future predictions.

VIII. METHOD

In this paper, we bring out the comparison between two different models that can be used in sentiment analysis. We collected our dataset on mobile reviews.

8.1 Dataset

The dataset for Mobile reviews was collected from Kaggle [8]. The dataset was cleaned (removed all unwanted columns). The Dataset finally contained only 2 columns (Review and Rating). It was an unsupervised dataset. We converted the dataset to supervise (labelled) using the following conditions if the ratings>=3, then the review is labelled as positive else: It is labelled as negative.

Finally, we collected 5000 positive samples and 5000 negative samples (Total size of the dataset used for the experiment is 10000). We split this data into a training set and testing set in the ratio (8000) 80%, (2000) 20% respectively.

8.2 Pre-Processing

The texts (reviews) were pre-processed. All the characters except [! a-zA-Z] were removed from the reviews. This was done with the help of regular expressions. All the words in the reviews were converted to lower case. The stop words (such as a, an, in, the) were removed using NLTK's corpus module.

8.3 Task A

Using Python's NLTK Framework, we implemented the VADER Lexicon module. Here the complete dataset (all 10,000 reviews) was used to test the VADER Lexicon module. The output obtained from the model was in the form of a dictionary containing positive, negative, neutral and compound scores. In Vader Lexicon, when a sentence (review) is passed through the model, it extracts sentimental words and its intensity. The polarity score of a word lies in a range of -4 to 4, where -4 being extremely negative and +4 being extremely positive. But overall sentiment score of statement ranges between -1 to 1. This is obtained through normalizing the sentiment scores of the words. This normalization results in a new metric called the compound score. The compound score tells us about the intensity and polarity of the review. If there is more than one sentiment word in a sentence, the sentiment scores are added and then normalized to obtain the compound score.

Calculation of compound Score

$$CompoundScore = \frac{x}{\sqrt{x^2 + alpha}} \qquad (1)$$

In equation (1), Alpha is a constant and usually chosen as 15, x is the sum of polarity scores of all the words. Let's look into an example. "The food here is good and service is nice." Here good and nice are two sentiment words with polarity scores of 1.9 and 1.8. Now calculating the compound score, our x value will be (1.9+1.8) =3.7.So our compound score comes out to be 0.6907. We then classified the reviews based on the compound score obtained by the VADER tool. If Compound Score >=0, then the review is classified as positive, otherwise it is classified as negative.

Evaluation Metric Used

The evaluation metric that was used to conclude our result was an Accuracy Score.

$$AccuracyScore = \frac{1}{n}\sum_{k=1}^{n}(y == \hat{y}) \qquad (2)$$

In equation (2), Accuracy Score is a fraction of correct predictions over n samples. Here y represents the actual value of kth review and y^ represents the predicted value of kth review.

8.4 Task B

LSTM is a machine learning model. We cannot directly pass the text to the LSTM model; we need to convert the text to vectors before passing it to the model.

Converting Words to Word Vector

We used NLTK's tokeniser to tokenise the reviews such that every unique word gets a sequence number. This sequence was then padded so that all reviews are of equal length. Now we split our processed data into train and test data (80%, 20%) respectively.

GloVe

GloVe [5], coined from Global Vectors, is a model for distributed word representation. The model is an unsupervised learning algorithm for obtaining vector representations for words. We used the GloVe file (GloVe.6B.50d.txt) which consisted of (400K words, 50d vectors). In this file, the first element of every line is a word and remaining elements are vectors for that word. We converted the data of this file to dictionary format where the key is a word and the value is a vector. We then created embedding matrix which has a shape of (k*50), where k is a number of unique words obtained using tokenizer. In this, the index was the sequence number we obtained from the tokenizer, and the corresponding vector was the value inserted at that index.

Training and Evaluating Model

We used keras deep learning framework to evaluate the model. We passed this word vectors to the LSTM embedding layer (The first layer of our network) and obtained the predictions. The optimizer we used for this model was 'Adam' optimizer. We ran 100 epochs with a batch size of 128.

Evaluation metric

True Positive - Total number of reviews that the model has correctly predicted as positive.

True Negative -Total number of reviews that the model has correctly predicted as negative.

False Positive – Total number of reviews that the model has incorrectly predicted as positive.

False Negative- Total number of reviews that the model has incorrectly predicted as negatively.

Precision

Precision is a fraction of predicted positive that is actually positive. It is calculated using the equation (3).

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \qquad (3)$$

Recall

The fraction of positives predicted correctly. It is calculated using the equation (4).

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \qquad (4)$$

f1 Score

The f1Score is the harmonic mean of Recall and Precision, with a higher score as a better model. It is computed using the equation (5).

$$f1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (5)$$

IX. RESULT

9.1 Results of Task A

After collecting the compound score from all the reviews using eq. 1, we classified the reviews based on their compound score as positive or negative. Then, we compared the result obtained using VADER to the original result and calculated the accuracy. We used accuracy score from SK-learn module to find out the accuracy. The result is shown in Table I.

Table I: Results Of Task A

Parameter	Score
Accuracy	73.48%

9.2. Results of Task B

The model predicted the output 1 for a positive review and 0 for a negative review. The predicted values and actual values were compared to find accuracy score using accuracy score from SK-learn module. The results are shown in TableII.

Table II: Results Of Task B

Parameter	Score
Precision	0.8947
Recall	0.8994
f1Score	0.8970
Accuracy	89.89 %

9.3. Discussion

Table III shows the comparison of accuracy attained by models used in tasks A and B. It is clear that the LSTM model over performs the VADER model. Whenever we need high accuracy prediction, we can make use of LSTM. If enough dataset is not available and accuracy is not of the concern, then we can go with VADER because for LSTM we need a sufficient amount of training data and also more computation time.

Table III: Comparison Of Models

Model	Accuracy
VADER	73.48%
LSTM	89.89%

X. CONCLUSIONS

In this paper, we have presented two models that can be used for sentiment analysis and provided the details of how the experiment was conducted. As per the results obtained on the dataset, the accuracy obtained by LSTM is more compared to VADER. So, LSTM is classifying the reviews better and LSTM model has learned better.

REFERENCES:

1. Hutto, C.J. ,Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.
2. Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and TrendsÂő in Information Retrieval 2.1âĂŞ2 (2008): 1-135.
3. Ribeiro, Filipe N., et al. "Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods." EPJ Data Science 5.1 (2016): 1-29.
4. Tang, Duyu, et al. "Effective LSTMs for target-dependent sentiment classification." arXiv preprint arXiv:1512.01100 (2015).
5. Pennington, Jeffrey, Richard Socher, and Christopher Manning."GloVe: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
6. https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones.