International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 5

ISSN: 2454-4248
159 – 161

An Efficient Approach of Review Sentiment Analysis

Mohit Chaturvedi
Department of Computer Science & Engineering
Feroze Gandhi Institute of Engineering & Technology
Raebareli(UP), India
mohitcck@gmail.com

Shruti Tripathi
Department of Computer Science & Engineering
Maharshi University of Information & Technology
Lucknow(UP), India
shru_tri@yahoo.com

***Abstract*—** This paper investigates the utility of linguistic feature for detecting the sentiments of review messages. We take a supervised approach to the problem and 50,000 movie reviews for building our training data. We investigate an efficient method to build a strong model for extracting the features that contain sentimental information.

Keywords- *Logistic Regression, Stochastic gradient descent, Tokenization, Normalization ,Stop Words.*

*****

I. INTRODUCTION

Sentiment analysis is becoming one of the most profound research areas for prediction and classification. Automated sentiment analysis of text is used in fields where products and services are reviewed by customers and critics. Thus, sentiment analysis becomes important for businesses to draw a general opinion about their products and services. Our analysis can help concerned organizations to find opinions of people about any product from their reviews, if it is positive or negative. One can in turn formulate a public opinion about a product.

Our goal is to calculate the polarity of sentences that we extract from the text of reviews. We will experiment to model sentiment from reviews and try to find an efficient and reliable method to categorize according to its polarity "positive or negative".

II. LITERATURE REVIEW AND PROBLEM IN SENTIMENT ANALYSIS

The informal and specialized language used in Reviewing products, as well as the very nature of the microblogging domain make Review sentiment analysis a very different task(e.g.,(Kouloumpis, E., Wilson, T., Moore, J., 2011))[4]. It's an open question how well the features and techniques used on more well-formed data will transfer to the microblogging domain. Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification to learning the polarity of words and phrases (e.g., (Hatzivassiloglou and McKeown 1997; Esuli and Sebastiani 2006)).Given the text data, classifying the sentiment of text messages is most similar to sentence-level sentiment analysis (e.g., (Yu and Hatzivassiloglou 2003; Kim and Hovy 2004))[3].

Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010; Bifet and Frank 2010). Exploit movie review for collecting training data (A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A.Y. Ng, and C. Potts. Learning Word Vectors for Sentiment Analysis. In the proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics) [1]. And also use hashtags for creating training data (Davidov, Tsur, and Rappoport 2010), we experiment to find an efficient way of using supervised learning for 2-way sentiment classification.

The calculation to find the polarity of text review can be difficult and tricky because of the ambiguous nature of text. Consider a sentence "the movie interstellar was visually a treat but the story line was terrible". Now one can clearly see how categorizing this sentence as negative, positive or neutral can be difficult. The phrases "visually a treat" and "story line was terrible" can be considered positive and negative respectively but the degree of their 'positiveness' and 'negativeness' is somewhat ambiguous. We use a score for common positive and negative words and use this score to calculate the overall sentiment of a sentence.

III. PROPOSED WORK

A. *Data used during Expriment*

We use the large dataset of movie reviews from the **Internet Movie Database** (**IMDb**) that has been collected by Maas et al. (A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A.Y. Ng, and C. Potts. Learning Word Vectors for Sentiment Analysis. In the proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics)[1]. The movie review dataset consists of 50,000 polar movie reviews that are labelled as either *positive* or *negative*; here, positive means that a movie was rated with more than six stars on IMDb, and negative means that a movie was rated with fewer than five stars on IMDb. A compressed archive of the movie review dataset (84.1 MB) can be downloaded from

 ISSN: 2454-4248
159 – 161

"*http://ai.stanford.edu/~amaas/data/sentiment/"* as a gzip-compressed tarball archive[5].

B. Data Preprocessing

Data preprocessing consist of three steps:
1) Data Cleaning
2) Tokenization
2) Normalization

The text Data we downloaded contains HTML markup as well as punctuation and other non-letter characters. While HTML markup does not contain much useful semantics, punctuation marks can represent useful, additional information in certain NLP contexts. We remove all punctuation marks but only keep **emoticon** characters such as ":)" since those are certainly useful for sentiment analysis. To accomplish the data cleaning task, we will use Python's **regular expression** (**regex**) library, re.

```
>>> import re
>>> def preprocessor(text):
...     text = re.sub('<[^>]*>', '', text)
...     emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)', text)
...     text = re.sub('[\W]+', ' ', text.lower()) + \
' '.join(emoticons).replace('-', '')
...     return text
```

For tokenization we used the process of word stemming which is the process of transforming a word into its root form that allows us to map related words to the same stem. The original stemming algorithm was developed by Martin F. Porter in 1979 and is hence known as the **Porter stemmer** algorithm[5]. The Natural Language Toolkit for Python implements the Porter stemming algorithm. Using PorterStemmer from the nltk package , we tokenize our data to reduce words to their root form, like the word running stemmed to its root form run, we use this tokenizer just to reduce no of tokens we also remove all the stop words like *is ,and ,has and the* removing these token will be good for our normalization as these word have nothing to do with the sentiment of review.

IV. IMPLEMENTATION OF PROPOSED WORK

Our goal for these experiments is two-fold. First, we want to evaluate whether our training data with labels derived from movie review is useful for training sentiment classifiers. Second, we want to evaluate the effectiveness of our model in order to deploy it in bigger projects.

A. Training a Logistic Regression Model

The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a *logit transformation* of the probability of presence of the characteristic of interest.

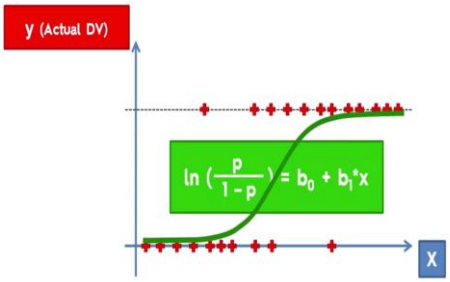

Figure 1. Logistic regression

We train a logistic regression model to classify the movie reviews into positive and negative reviews. First, we divide the DataFrame of cleaned text documents into 25,000 documents for training and 25,000 documents for testing.

After training the logistic regression model using sklearn gried search we obtain the testing accuracy of about 0.899 means we were able to categorize reviews with the accuracy of about 90 percent but the problem of using the Grid Search is that it could be computationally very expensive to construct the feature vector for 50000 review data. It takes about 40 minutes to complete on a standard computer, the model is useful but not efficient for real world application.

B. Using Stochastic Gradient Descent

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima by iteration.

It is an optimization algorithm that updates the model's weights using one sample at a time. we make use of the partial_fit function of the SGDClassifier in scikit-learn to stream the documents directly from our local drive and train a logistic regression model using small minibatches of documents.

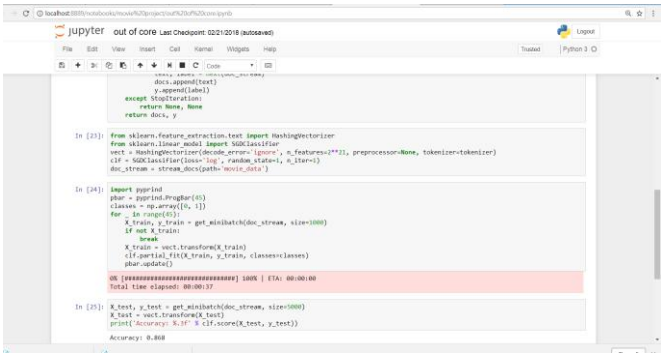

Figure 2. Accuracy testing

As we can see, the accuracy of the model is 87 percent, slightly below the accuracy that we achieved in the previous section using the grid search for hyper parameter tuning. However, SGDClassifier is very memory-efficient and took less than a minute about 37 sec to complete.

 ISSN: 2454-4248
159 – 161

V. CONCLUSION

Our experiments on review sentiment analysis shows that using logistic regression may not be very efficient for analysis of large amount of data however model give a good result in our prediction but SGD model is more efficient because of its iterative nature. Persisting the emotions icons in our token proved to be useful in finding sentiment of sentence. In our future investigation we are planning to see the effects of deep learning technique in order to find sentiments of text data.

REFERENCES

[1] Maas, A.L., Daly, R.E., Pham, P. T., Huang, D., A.Y. Ng, and Potts, C.2011, Learning word vector for sentiment Analysis, Computational Linguistics pp.399–433.

[2] Wilson, T., Wiebe, J. and Hoffmann, P. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3),pp.399–433.

[3] Yu, H., and Hatzivassiloglou, V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proc. Of EMNLP*.

[4] Kouloumpis, E., Wilson, T., Moore, J., 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG!. ICWSM.

[5] Sebastian Raschka,2015.Python Machine Learing –Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics. Packt

[6] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representatives," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[7] Jalaj S. Modha, Gayatri S. Pandi, Sandip, J. Modha, "Automatic Sentiment Analysis for Unstructured Data", IJARCSSE, Volume 3, Issue12,December 2013.

[8] Erik Cambria, Marco Grassi, Amir Hussain and Catherine Havasi," Sentic Computing for social media marketing", Springer Science+ Business Media, LLC 2011.

[9] Shichang Sun, Hongbo Liuand Ajith Abraham, "Twitter Part-Of- Speech Tagging Using Pre-classification Hidden Markov Model", IEEE International Conference on Systems, Man, and Cybernetics October 14-17,2012,COEX,Seoul,Korea 2012.

[10] Catherine Havasi, James Pustejovsky, Robert Speer and Henry Lieberman "Digital Intuition: Applying Common Sense Using Dimensionality Reduction", IEEE intelligent systems , July/August 2009.

[11] Cambria, Erik, Yangqiu Song, Haixun Wang, and Amir Hussain. "Isanette: A common and common sense knowledge base for opinion mining."In Data Mining Workshops (ICDMW), 2011IEEE 11th International Conference on, pp. 315-322.IEEE,2011.

[12] F.Suchanek, G.Kasneci,and G.Weikum, "YAGO:A Core of Semantic Knowledge, " Proc. Int"l World Wide Web Conf., 2007, pp. 697–706.

[13] Partha Sarkar and Bipul Syam Purkayastha, "A Study on Deep Linguistic Processing with Special Reference to Semantic and Syntactic Levels," International Journal of Computer Applications(0975– 8887) Volume88 –No.13,February2014.

[14] Im, TanLi, Phang Wai San, Chin Kim On, Rayner Alfred,and Philip Anthony. "Analyzing market sentiment in financial news using lexical approach. "In Open Systems (ICOS), 2013 IEEE Conference on, pp. 145-149.IEEE,2013.

[15] Zhongwu Zhai, Bing Liu, Hua Xu and Hua Xu, Clustering Product Features for Opinion Mining,WSDM'11, February 9–12, 2011, Hong Kong, China. Copyright 2011 ACM 978-1-4503-0493

[16] Singh and Vivek Kumar, A clustering and opinion mining approach to socio-political analysis of the blogosphere, Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference.

[17] V. S. Jagtap and Karishma Pawar, Analysis of different approaches to Sentence-Level Sentiment Classification, International Journal of Scientific Engineering and Technology (ISSN : 2277-1581) Volume 2 Issue 3, PP : 164-170 1 April 2013

[18] Erik Cambria, Bjorn Schuller and Yunqing Xia, "New Avenues in Opinion Mining and Sentiment Analysis", 1541-1672/13 IEEE 15 Published by the IEEE Computer Society, 2013.

[19] G.Vinodhini and RM.Chandrasekaran, Sentiment Analysis and Opinion Mining: A Survey, Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[20] Havasi, Catherine, Erik Cambria, Björn Schuller, BingLiu, and Haixun Wang."Knowledge-based approaches to concept-level sentiment analysis."IEEE Intelligent Systems 28,no.2(2013): 0012- 14.

[21] Tsai, Angela Charng-Rurng, Chi-En Wu, Richard Tzong-HanTsai, and Jane Yung-jen Hsu. "Building a concept-level sentiment dictionary based on common sense knowledge." IEEEIntelligentSystems28, no.2 (2013):22-30.