International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 03, March – 2017, PP – 84-90

Feedback analysis using Sentiment Analysis for E-commerce

Dr.S.Murugavalli[1], U.Bagirathan[2], R.Saiprassanth[3], S.Arvindkumar[4]

[1]*Head of the Department, Panimalar Engineering College, Chennai 600123, INDIA*

[2,3]*Panimalar Engineering College, Chennai 600123, INDIA*

Abstract: In the current scenario existing systems typically use the star ratings of products to determine its quality, but ignore the tangible feedback. It is problematic because the star ratings may differ from the tangible feedback .We propose exploiting sentiment analysis algorithm (Sentiment analysis refers to the employment of linguistic communication processing, text analysis and computational linguistics to spot and extract subjective data in supply material) to judge product feedback. This issue is addressed by encoding sentiment data of feedbacks in conjunction with contexts of words in sentiment embeddings .Bycombining context and sentiment level evidences, we can identify the sentiment polarity of the product feedback provided by the customer.

Keywords: Sentiment Analysis, Feedback, Data mining

I. INTRODUCTION

E commerce websites are typically used for purchasing and merchandising products and services, over a network system, primarily the web. They are growing at a very fast pace. Nowadays people prefer ecommerce websites over direct purchase. Through these websites people view their desired products, read reviews on them and decide on buying the required products. Here the administrator's role is to manage the inventory of the products based upon the products quality. Now in order to determine the quality of a product, the star ratings given by the user is taken into consideration. If the ratings are good, then the product is promoted within the website. If they are poor the product is removed from the inventory. This is the general process undertaken in an e-commerce website.

The user who has bought a product has an option to provide comments on the products they purchased. These comments can be positive or negative based on the product quality and they have no restrictions on what they want to express. These Comments are not considered by the administrator to manage the inventory since there may be thousands of products each having their own quantity of reviews and it is not possible for the admin to read each comments. Hence the star rating feature was brought into usage. It enables the administrator to find out what the customers actually feels about the product without the hassle of going through thousands of feed backs given by the customer. This will enable the administrator to improve the standard of his e commerce website.

But there is a complexity in the star ratings, there is no quantifiable measurement standard to identify good products from bad based upon the star ratings. It is up to the admin in order to set the standards to identify good products from bad, such as one admin may consider ratings below 1 is considered bad ,other may consider zero star ratings as bad. This variable measurement standards will not be useful when it comes to making important decisions such as removing the product from the e commerce website, switching to a different retailer for the same product so on and so forth. The star ratings can also be automatically given higher using bots. Hence we are in need to use different method in order to receive the customer's feedback which does not involve reading of these customer's feed backs manually one by one.

Figure 1: Sentiment Analysis

Sentiment analysis refers to the process of mining the texts in order to identify the tone of the passage written by the user. There are usually three tones for a passage written by the user: positive, negative and neutral. Sentiment analysis is actually used for this purpose.

Humans as we read the passage we can easily identify the tone of the passage using the context and the words used but for computers it is actually a different scenario, they mainly use the contexts to identify the sentiments of the passage. Here comes the complexity, words with opposite meanings such as good and bad are actually used in the same context but they represent different sentiments, this can't be recognized by the computers. Consequently this method is opted to the utilization of polarity scores for certain words in the passage in order to identify the sentiment of the passage.

Hence the utilization of sentiment analysis in ecommerce websites in order to analyze the customer's feedback on the products will prove to be the best alternative to the process of manually reading the customers feedback, as well as the star ratings. This will provide the details to the administrator on the overall score of each product which is calculated from the feedback given by the customer. This will represent in detail the amount of positive score and the amount of negative score for the product clearly to the administrator, based on which the important decisions on managing the inventory can be made by the administrator.

II. CORRELATED RESEARCH

In 2010, Luciano Barbosa Et.al[1] proposed an effective and robust sentiment detection approach for Twitter messages, which uses biased and noisy labels as input to build its models. Although noisy and biased, the information sources offer labels of cheap quality and, since they need completely different bias, combining them additionally brought some edges. The most limitation of this approach is that the cases of sentences that contain antagonistic sentiments.

In 2010, Takeshi Sakaki Et.al[2] investigated the time period nature of Twitter, particularly for event detection. Linguistics analyses were applied to tweets to categorize them into a positive and a negative class. They considered each Twitter user as a sensor, and set a problem to detect an event based on sensory observation which easily detect and provide estimation for location of the earthquake but multiple events cannot be detected simultaneously.

In 2013, N. Yang Et.al[3] explores applying deep neural network for word alignment task. Their model integrates a multi-layer neural network into an HMM-like framework, where context dependent lexical translation score is computed by neural network, and distortion is modeled by a simple jump-distance scheme. This method generates a very compact model with much fewer parameters which makes it out perform the existing models, but the stop words are inherently hard to align, which often requires grammatical judgment unavailable to our models; as they are also extremely frequent.

In 2013 T.Mikolov et.al[4]presented many extensions of the first Skip-gram model. They show that sub-sampling of frequent words throughout coaching ends up in a major acceleration and improves accuracy of the representations of less frequent words. The main advantage of this model is training of the Skip-gram model does not involve dense matrix multiplications. This makes the training extremely efficient. To obtain higher accuracy we need larger training samples, at the expense of the training time.

In 2014 D. Tang et.al[5]proposed learning continuous word representations as options for Twitter sentiment classification underneath a supervised learning framework. They learn sentiment-specific word embedding by integration the sentiment info into the loss functions of three neural networks. Sentiment-specific word embeddings outperform existing neural models by large margins. The drawback of this model is it learns sentiment-specific word embedding from scratch which is time consuming.

III. METHODOLOGY

We present the methods for learning sentiment embeddings in this section. We first describe standard context-based neural network methods for learning word embeddings. Afterwards, we introduce our extension for capturing sentiment polarity of sentences before presenting hybrid models which encode both sentiment and context level information. We then describe the integration of word level information for embedding learning.

Notation

We notate the meaning of variables used in this paper. In particular, wi means a word whose index is i in a sentence, hi is context words of wi in one sentence, ei is the embedding vector of wi. In this work, we implement the neural network approaches with some basic neural layers, including lookup, hTanh, linear and softmax. For each neural layer, Olayer means the output vector. The implementations of these layers can be found at: http://ir.hit. edu.cn/~dytang.

Modeling Contexts of Words

Dominating word embedding learning algorithms are on the basis of distributional hypothesis, which states that the representations of words can be reflected by their contexts. In this subsection, we describe a prediction model and a ranking model to encode contexts of words for learning word embeddings. These context-based models will be naturally incorporated with sentiment-specific models for learning sentiment embeddings.

Prediction Model

An effective way to encode contexts of words into word representation is "context prediction. Given a target word w_i and its context words h_i, "context prediction" aims to predict w_i based on h_i, which can be viewed as language modeling. The contexts of a target word could be preceding, following or surrounding words occurred in a piece of text. Since we do not focus on an exact language model, we investigate surrounding words $h_i=\{w_{i-c},w_{i-c-1},...w_{i-1},w_{i+1},...w_{i+c-1},w_{i+c}\}$context in this work. The method can be naturally extend to preceding or following contextwords. We build a prediction model analogous to the representative "context prediction" neural language. They model the conditional probability $P(w_i|h_i)$ of predicting a target word w_i based on its contexts h_i by taking word embeddings as a parameter. The scoring function is a feed-forward neural network consisting of lookup -> linear-> hTanh-> linear ->softmax.

Lookup layer (also referred to as projection layer) contains a lookup table LT $\in R_{dx|V|}$which maps each word to its continuous vector, where d is the dimension of each word vector and |v| is the vocabulary size. The lookup operation can be viewed as a projection function that uses a binary vector idx_i which is zero in all positions except at the ith index

$$ei= LT .idxi\in R1xd$$

We concatenate the embeddings of context words {wi-c,wi-c+1,...,wi-1,wi+1,...wi+c-1,wi+c} as the output of lookup layer, which is formalized as below:

$$Olookup=[ei-c;...ei-1,ei+1...;ei+c]R1xd.2c$$

Afterwards, Olookup is fed to a linear layer for dimension transformation, where $Wl1\in Rlenx2c$ is the position-dependent weights and $bl1\in R1xlen$ is the bias of linear layer, len is the length of output vector Ol1 of linear layer

$$Ol1=wl1.Olookup+bl1$$

The linear layer is followed by a non-linear function layer for adding element wise non-linearity. Standard non-linear layers include hyperbolic tangent, hard hyperbolic tangent(htanh), sigmoid and rectifier. We use hTanh in this work for its computational efficiency and effectiveness. We denote the output vector of hTanh as $O_{htanh}\in R_{1xlen}$

$$hTanh(x)=\{-1, \text{if } x< -1; x, \text{if } -1 \leq x \geq 1; 1, \text{if } x>1.$$

The output layer of standard neural language model is a softmax layer whose output length is vocabulary size. Softmax is suitable for this case as its outputs can be interpreted as conditional probabilities, which is calculated as below:

$$softmax_i=\exp(zi)\Sigma\exp(zi')i'$$

Conditional probability $p(w_i|h_i)$ is the idx_{wi}-th value in the output vector of softmax layer, where idx_{wi} is the index of w_i in vocabulary. The optimizing objective of a given word context pair (w_i,h_i) is to maximize $P(w_i|h_i)$

However, directly predicting the probability of $P(w_i|h_i)$ is time-costly as the length of output softmax layer is vocabulary size, which is typically hundreds of thousands. To speed-up the training process, we use noise contrastive estimation, and transfer the problem of "context prediction" to distinguish a word-context pair as real case or artificial noise by means of logistic regression. The probability that the given sample came from the data is $P(D|w,\theta)$ $P(D|w,\theta)=\exp(f\theta(wi,hi))\exp(f\theta(wi,hi))+k.\exp(f\theta 9wn,hi))$

Where, wn is an artificial noise such as a randomly selected word from vocabulary. The scaling factor k accounts for the fact that noise samples are k times more frequent than data samples .The score function $f\theta(w,h)$ quantifies the compatibility between context hi and target word wi, which can be naturally defined as a feed forward neural network consisting of lookup -> linear -> hTanh ->linear. The input of lookup layer is the concatenation of the current word w and context words h. The output is a linear layer with output length as 1, which stands for the compatibility between context h and word w. We implement $P(D|w,\theta)$with a softmax layer and maximize the log probability of the softmax for parameter estimation

$$loss_{cPred}=\Sigma logP(D|w,w\in T\theta)$$

Ranking Model

Collobert and Weston use a pairwise ranking approach to capture the contexts of words for learning word embeddings. It holds the similar idea with noise contrastive estimation but the optimizing objective is to assign a real word context pair (w_i,h_i) a higher score than an artificial noise (w_n,h_i) by a margin. They minimize the following hinge loss function, where T is the training corpora:

$\Sigma max(0,1-f_{(wi,hi)\in T\theta}(w_i,h_i)+f_\theta(w_n,h_i))$

The scoring function $f_\theta(w,h)$ is achieved with a feed forward neural network. Its input is the concatenation of the current word w_i and context words h_i, and the output is a linear layer with only one node which stands for the compatibility between w and h. During training, an artificial noise w_n is randomly selected over the vocabulary under a uniform distribution.

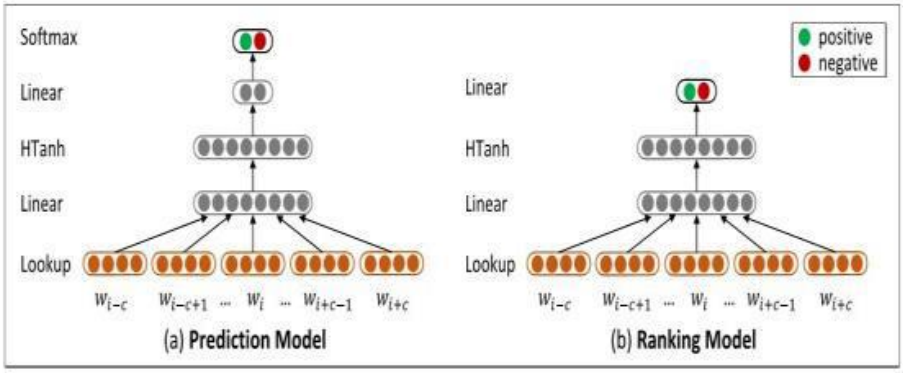

Fig. 2 An illustration of two neural networks that model the sentiment polarity of sentences for learning sentiment embeddings

IV. EXISTING SYSTEM

In the current scenario in order to determine the product quality e-commerce websites only consider star ratings .Where a product with higher Star ratings are considered to be a superior product. But these star ratings are not reliable since they are imprecise and could also be easily manipulated using bots to modify the actual ratings given by the previous users who had actually purchased the product.

V. PROPOSED SYSTEM

The proposed system uses sentiment embeddings and performs sentiment analysis on the product feedback given by the user on the e commerce site. It uses contexts and sentiments to identify the polarity score of the feedback and assigns them to the product. This method maps the words from the feedback to the embedding space. The nearest neighbors in the embedding space are not only semantically similar but also favor to have the same sentiment polarity, so that it is able to separate good and bad feedbacks to opposite ends of the spectrum. Then the polarity scores from the feedbacks of the particular product is added to obtain overall score for the product, which is then displayed as a graph to the administrator based upon which he can manage the inventory in order to improve the overall quality of the website.

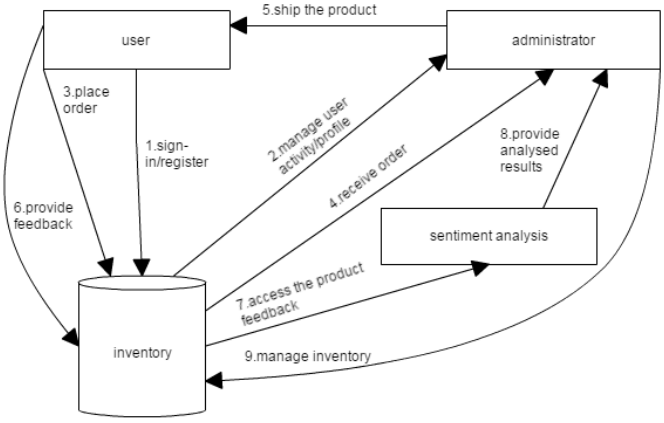

Fig.3 Architecture diagram of Feedback Analysis Using Sentiment Analysis

The Architecture Diagram [Fig 3] has the following modules involved

1. User
2. Administrator
3. Sentiment analysis
4. Inventory

User:

User profiling is the foremost part of this method which covers the single sign on user and also registers new user to the e-commerce site. This involves maintaining user's personal information which is required to ship the products to the user at a destined time.

Administrator:

This involves the administrator side work where he deploys the products, maintains the site, tracks user walkthrough towards the site and captures end user reviews into inventory Database. Product administration involves adding new products, updating product information, deleting obsolete products from the site and other common activities.

Sentiment analysis:

This module extracts the feedback from the inventory and performs sentiment analysis on it. It extracts the important sentiment words from the feedbacks computes the score and finally aggregates the scores of all feedbacks of the product to provide its overall score.

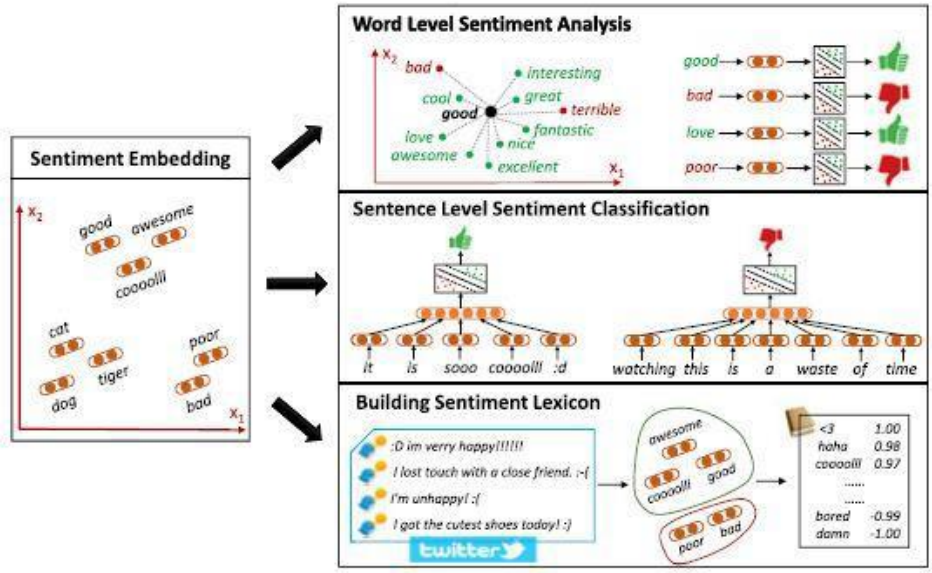

Fig.4 Sentiment embeddings

It basically splits the words into positive and negative space embeddings. Here positive words have positive score and negative words have negative score such as the word "good" is positive and it has a score of +1.0 while the word "bad" is negative and so it has -1.0 as the sentiment score. Now a product quality is determined by the overall sentiment score.

If the product has positive score then the positive reviews outweigh the negative reviews, consequently the product is considered to be good .If the product has negative score then the product is bad in quality or in its shipping. Other than these the product can also have a neutral score i.e., zero if the positive and the negative reviews are equal. In such case the product is considered to be neutral product.

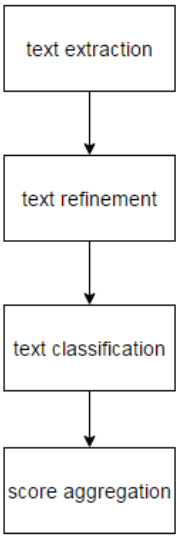

Fig.5: Steps in Sentiment Analy

Text extraction:
This is the first step in which the sentiment analysis module extracts the words from the feedback that could potentially affect the outcome.
Text refinement:
This step involves the refinement of the extracted words from irrelevant words and phrases.
Text classification:
Here the extracted and the refined text are classified into positive and negative classes based upon their polarity.
Score aggregation:
This step computes the total score from the classifier and then aggregates it to produce the final sentiment score. Finally the positive and the negative scores of the product are shown in the form of bar graph to the user.

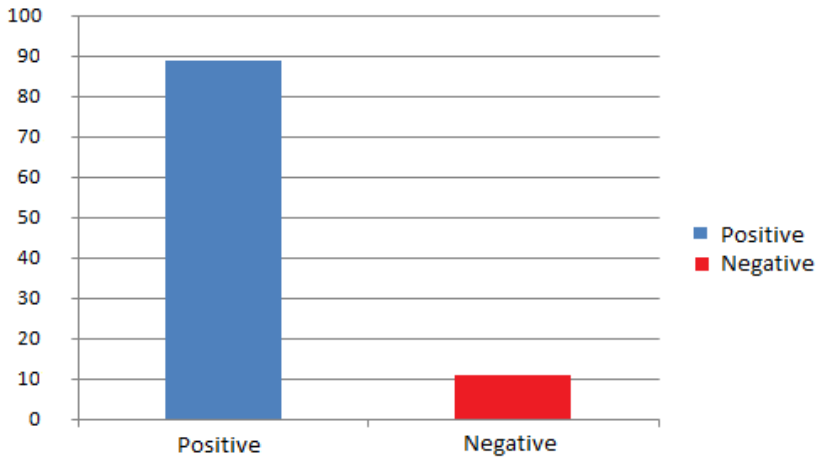

Fig.6: Product Score Graph

Inventory:
Inventory in other words is the database, which contains the details about the products such as a product id, brand, pricing etc. It also contains the user profile such as name, address, payment method etc. This method involves storing the product feedback provided by the customer. It also contains the table that has the sentiment scores for each word in use and the admin is provided with the rights to add new words into the sentiment score table or remove the words from the table.

VI. CONCLUSION

A merchandise quality in E-Commerce websites depends upon the reviews given by the user who have purchased and used it. Since the prevailing star ratings aren't entirely confirmative, by using the conception of Text mining hand to hand with Sentiment analysis brings out the required information for a user who desires to grasp others opinion on that product. Sentiment data of texts in conjunction with its actual context is encoded in sentiment embedding to support sentiment analysis algorithm. As a result the written feedback of users is taken into thought to work out the particular quality of the merchandise from which it will be either promoted or removed by the administrator.

VII. REFERENCES

[1]. Luciano Barbosa, Junlan Feng,"Robust Sentiment Detection on Twitter from Biased and Noisy Data",2010

[2]. Takeshi Sakaki, Makoto Okazak, Yutaka Matsuo"Earthquake shakes twitter users: Real-time event detection by social sensors ()", 2010

[3]. N. Yang, S. Liu, M. Li, M. Zhou, and N. Yu
"Word alignment modeling with context dependent deep neural network", 2013

[4]. T.Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean "Distributed representations of words and phrases and their compositionality", 2013

[5]. D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin "Learning sentiment-specific word embedding for twitter sentiment classification", 2014

[6]. D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu"Building largescale twitter-specific sentiment lexicon: A representation learningapproach"**,** 2014

[7]. J. Pennington, R. Socher, and C. Mannin "Global vectors for word representation", 2014

[8]. Rui Xia,Feng Xu,Chengqing Zong,Qianmu Li,Yong Qi ,Tao Li "Dual Sentiment Analysis: Considering Two Sides of One Review" , 2015

[9]. Shulong Tan, Yang Li, Huan Sun, Ziyu Guan , Xifeng Yan, , Jiajun Bu, , Chun Chen, , and Xiaofei He "Enterpreting the Public Sentiment Variations on Twitter", 2015

[10]. D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A deep learning system for twitter sentiment classification", 2014

[11]. R. Feldman, "Techniques and applications for sentiment analysis", 2013

[12]. T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis", 2005.

[13]. P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T.Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter", 2013.

[14]. A. Severyn and A. Moschitti, "On the automatic learning of sentiment lexicons", 2015.

[15]. J. Li and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding", 2015.
A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C.Potts, "Learning word vectors for sentiment analysis",2011.

[16]. X. Hu, J. Tang, H.Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals", 2013.

[17]. A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining", 2010.

[18]. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentimentclassification using machine learning techniques" , 2002.

[19]. A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classificationusing distant supervision", 2009.

[20]. R. L. Rosa, D. Z. Rodríguez, and G. Bressan, "Music recommendation system based on user's sentiments extracted from social networks", 2015.

