Product Aspect Ranking

Avinash Mote[1], Pratik Madhavi[2], Neha Patil[3], Sonal Choudhari[4]

[1,2,3] Student, Department of Computer Engineering, Datta Meghe College of Engineering. Maharashtra, India
[4] Professor, Department of Computer Engineering, Datta Meghe College of Engineering, Maharashtra, India

---

Abstract — *In this paper we are going to rank a product from different websites according to a customer reviews. Numerous consumer reviews of products are now available on the Internet. Consumer reviews contain rich and valuable knowledge for both firms and users. However, the reviews are often disorganized, leading to difficulties in information navigation and knowledge acquisition. This article proposes a product aspect ranking framework, which automatically identifies the important aspects of products from online consumer reviews, aiming at improving the usability of the numerous reviews. The important product aspects are identified based on two observations: (a) the important aspects are usually commented by a large number of consumers; and (b) consumer opinions on the important aspects greatly influence their overall opinions on the product.*

In particular, given the consumer reviews of a product, we first identify product aspects by a shallow dependency parser and determine consumer opinions on these aspects via a sentiment classifier. We then develop a probabilistic aspect ranking algorithm to infer the importance of aspects by simultaneously considering aspect frequency and the influence of consumer opinions given to each aspect over their overall opinions. The experimental results on a review corpus of 21 popular products in eight domains demonstrate the effectiveness of the proposed approach. Moreover, we apply product aspect ranking to two real-world applications, i.e. document level sentiment classification and extractive review summarization, and achieve significant performance improvements which demonstrate the capacity of product aspect ranking.

Keywords: **Product aspects, Aspect ranking classification, Sentiment classification, Consumer review, Extractive review summarization etc.**

1. INTRODUCTION

Microblogging websites like Twitter have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. . In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on microblogs. Organizations can also use this to gather critical feedback about problems in newly released products. Consumers can use sentiment analysis to research products or services before making a purchase. Nowadays the numbers of reviews, suggestions, feedbacks are increasing in enormous manner. Every person wants to share his views and experience about the product like review on product, review on movie, Person tweets etc. Reviews play vital role in helping and suggesting other person in their decision making. But on the other hand it becomes difficult to read all reviews and make decision as per. Another challenge of microblogging is the incredible breadth of topic that is covered. It is not an exaggeration to say that people tweet about anything and everything. Therefore, to be able to build systems to mine Twitter sentiment about any given topic, we need a method for quickly identifying data that can be used for training. Thus, mining this data, identifying the user opinions this is done by performing detailed sentiment analysis on the data.

The fields of opinion mining and sentiment analysis are distinct but deeply related. Opinion mining focuses on polarity detection [positive, negative, duplicate and sarcastic whereas sentiment analysis involves emotion recognition. Sentiment Analysis, also called opinion mining, is a field of study that analyzes people's opinions, sentiments, emotions, evaluation and attitude towards entities such as

product, services, organizations, individuals, issues, events, topics and their attributes. Sentiment Analysis is a type of Natural Language Processing for tracking the mood of the public about a particular product or topic. There are two main approaches to the problem of sentiment analysis: lexical approach and machine learning approach. The first, a lexicon-based method, uses a dictionary of words with assigned to them semantic scores to calculate a final polarity of a tweet, and incorporates part of speech tagging. Machine Learning is one of the most prominent techniques gaining interest of the researchers due to its adaptability and accuracy. In Sentiment Analysis mostly the supervised learning variants of this technique are employed. It comprises of the following stages: Data Collection, Preprocessing, Training data, Feature Extraction, Classification and Plotting Results. Machine Learning approach uses classification algorithms namely: Naive Bayes, Maximum entropy and SVM.

2. LITERATURE SURVEY

Analysis of on-line opinions became a popular research topic recently. Topics related to the one discussed in this work, have been researched before. Sentiment analysis has been practiced on a variety of topics. For instance, sentiment analysis studies for movie reviews[2], product reviews[3], and news and blogs[4][5]. In this section, Twitter specific sentiment analysis approaches are reported. The research on sentiment analysis so far has mainly focused on two things: identifying whether a given textual entity is subjective or objective, and identifying polarity of subjective texts[3]. The opinion spam problem was first formulated by Jindal and Liu[5] in the context of product reviews. By analyzing several million reviews from the popular Amazon.com, they showed how widespread the problem of fake reviews was. The existing detection methods can be split in the context of machine learning into supervised and unsupervised approaches. The authors had to build their own dataset, and the simplest approach was to use near-duplicate reviews as examples of deceptive reviews. Although this initial model showed good results, it is still an early investigation into this problem. Analysis of on-line opinions became a popular research topic recently. Topics related to the one discussed in this work, have been researched before. Sentiment analysis has been practiced on a variety of topics. For instance, sentiment analysis studies for movie reviews, product reviews, and news and blogs. In this section, Twitter specific sentiment analysis approaches are reported.

The research on sentiment analysis so far has mainly focused on two things: identifying whether a given textual entity is subjective or objective, and identifying polarity of subjective texts. The opinion spam problem was first formulated by Jindal and Liu[5] in the context of product reviews. By analyzing several million reviews from the popular Amazon.com, they showed how widespread the problem of fake reviews was. The existing detection methods can be split in the context of machine learning into supervised and unsupervised approaches. The authors had to build their own dataset, and the simplest approach was to use near-duplicate reviews as examples of deceptive reviews. Although this initial model showed good results, it is still an early investigation into this problem. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best feature and most efficient classification techniques.

3. METHODOLOGY

3.1 Pre-processing:

Keyword extraction is difficult in twitter due to misspellings and slang words. So to avoid this, a pre-processing step is performed before feature extraction. Pre-processing steps include removing url, avoiding misspellings and slang words. Misspellings are avoided by replacing repeated characters with two occurrences. Slang words contribute much to the emotion of a tweet. So they can't be simply removed. Therefore a slang word dictionary is maintained to replace slang words occurring in tweets with their associated meanings. Domain information contributes much to the formation of slang word dictionary.

Steps for preprocessing :

i. Negation:

Dealing with negations (like "not good") is a critical step in Sentiment Analysis. A negation word can influence the tone of all the words around it, and ignoring negations is one of the main causes of misclassification. In this phase, all negative constructs (can't, don't, isn't, never etc)

are replaced with "not". This technique allows the classifier model to be enriched with a lot of negation bigram constructs that would otherwise be excluded due to their low frequency.

ii. Dictionary:

This module uses the external python library PyEnchant6 , which provides a set of functions for the detection and correction of misspelled words using a dictionary. As an extension, this module allows us to substitute slang with its formal meaning (i.e., l8 → late), using a list. It also allows us to replace insults with the tag "bad word". The motivation for the use of these functions is the same as for the basic preprocessing operation, i.e., to reduce the noise in text and improve the overall classification performances.

iii. Stemming :

Stemming techniques put word variations like "great", "greatly", "greatest", and "greater" all into one bucket, effectively decreasing entropy and increasing the relevance of the concept of "great". As in the case of emoticons, with the use of this technique it is possible to combine features with the same meaning and reduce the entropy of the model.

iv. Stopwords :

Stop words are words which are filtered out in the preprocessing step. These words are, for example, pronouns, articles, etc. It is important to avoid having these words within the classifier model, because they can lead to a less accurate classification. Stopword removal enhances the system because it removes words which are useless for the classification phase. Analysis of on-line opinions became a popular research topic recently. Topics related to the one discussed in this work, have been researched before.

3.2 Feature Extraction:

There are various methods of feature extraction such as Vector space model, Principal Components Analysis (PCA), Latent semantic Analysis (LSA), etc. From all these methods vector space model is used because it is simple model based on linear algebra, allows computing a continuous degree of similarity between queries and documents and also allows ranking documents according to their possible relevance. A document is nothing but a group (more than one) of tokens. Every supervised machine learning algorithm requires each and every textual document to be represented in the form of a vector to start training on these documents, this is done through Vector Space Model (VSM).This is an algebraic model for text representation.

It consists of three stages:

Stage 1: Indexing of the documents where the content bearing terms [6] are extracted from the document text. The terms having very high or very low frequency distract the learning and hence are eliminated. Such words are known as function words [6,7,8]. These include the highly occurring stop words like "a, an, the, on". For eg: "New York is using sand-filled trucks to protect Thanks giving parade". Here, the words in bold are the content bearing words.

Stage 2: Weighting of the indexed terms for the enhancement of the retrieval of relevant document. There are many ways to give weight to the terms depending upon the application. Dj = (w1,j, w2,j….wn,j) is the representation of document in terms of weights. Here, each dimension corresponds to an independent term. Zero shows absence of any term from the document.

Stage 3: Ranking of the documents and taking the similarity measure into consideration to get the closet words from query document. The most popular similarity measure is the cosine coefficient, which measures the angle between the document and query vector[7].

3.3 Classification:

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviours given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases. In machine learning technique, algorithms like Naïve Bayes, maximum entropy, etc. From this we are using Naïve Bayes algorithm.

Naive Bayes Classifier: The Naïve Bayes classifier is based on Bayes rule, a practical Bayesian

learning model that is easy to understand and implement. The Bayes rule allows us to determine this probability of any event. It is the probabilistic approach to the text classification and can learn the pattern of examining a set of documents that has been categorized. Equation(1) compares the contents with the list of words to classify the documents to their right category or class. Let d be the tweet and c* be a class that is assigned to d, where

$$C^* = \arg\, mac_c P_{NB}(c|d)$$

$$P_{NB}(c|d) = \frac{(P(c))\sum_{i=1}^{m} p(f|c)^{n_{i(d)}}}{P(d)}$$

From the above equation (1) 'f' is a "feature", count of feature (fi) is denoted with ni(d) and is present in d which represents a tweet. Here, m denotes no. of features. Parameters P(c) and P(f|c) are computed through maximum likelihood estimates, and smoothing is utilized for unseen features. To train and classify using Naïve Bayes Machine Learning technique, we can use the Python NLTK library .

Steps for classification:

Proposed Naive Bayes classifier
Input: messages m= {m1, m2, m3.....mn},
Database: Naive Table NT
Output: Positive messages p= {p1, p2...}, Negative messages n= {n1, n2, n3....},
Duplicate messages nu= {d1, d2, d3...}
Sarcastic messages nu= {s1, s2, s3...}
M= {m1,m2,m3.....}
Step: 1 Divide a message into words
mi ={w1, w2, w3.....},i=1,2,....,n Step 2: if wi NT
Return +ve polarity and -ve polarity Step 3: Calculate overall polarity of a
word=log(+ve polarity)-log(-ve polarity) Step 4: Repeat step 2 until end of words
Step 5: add the polarities of all words of a message i.e. total polarity of a message.
Step 6: Based on that polarity, message can be positive

4. CONCLUSION

Sentiment analysis has become an important factor in decision making process in a particular field in this system we discussed the technique for preprocessing like negation, dictionary, stemmer and stop word, and retrieval of tweet through twitter. From study we can conclude that SVM acknowledges some properties of text like High Dimensional feature space, few irrelevant feature, and sparse instance vector. There are various approaches for sentiment analysis using machine learning techniques like Naïve Bayes and Maximum entropy and there is some feature extraction technique like Vector Space Model. Depending on the tweet's content, would be able to delegate, to one of the methods, Naïve Bayes have the responsibility of classifying and would use the maximum entropy for double validation and in some cases for training. Hence we can conclude that more the cleaner data, more accurate results can be obtained. Hence On the basis of results the tweets are categorized in positive, negative, sarcastic or duplicate tweets in the system.

REFERENCES :

[1] Salma Farooq, Hilal Ahmad Khanday: Opinion Spam Detection: A Review
[2] B. Pang and L. Lee: Using very simple statistics for review search: An exploration. In Proceedings of the International Conference on Computational Linguistics (COLING), (2008)
[3] K. Dave, S. Lawrence, and D. M. Pennock,: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews (2003) 519-528 6. N. Godbole, M. Srinivasaiah, and S. Skiena,: Large-scale sentiment analysis for news and blogs (2007)
[4] B. Pang and L. Lee,: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, vol. 2, no. 1-2 (2008) 1-135
[5] Jindal, N., & Liu, B. , "Opinion Spam and Analysis", In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 219–230), New York, NY, USA: ACM, 2008.
[6] Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. , "Finding Deceptive Opinion Spam by Any Stretch of the Imagination" , In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (pp. 309–319), 2011.
[7] Mukherjee, A., Liu, B., & Glance, N., "Spotting Fake Reviewer Groups in Consumer Reviews", In Proceedings of the 21st International Conference on World Wide Web (pp. 191–200), New York, NY, USA: ACM, 2012.
[8] Mukherjee, A., Venkataraman, V., Liu, B., & Glance, " Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews", UIC-CS-03-2013. Technical Report, 2013.
[9] Wang, G., Xie, S., Liu, B., & Yu, P. S., "Identify Online Store Review Spammers via Social Review

Graph", ACM Trans. Intell. Syst. Technol., 3(4), (pp.61:1–61:21), 2012.

[10] Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R., "Exploiting Burstiness in Reviews for Review Spammer Detection", In Seventh International AAAI Conference on Weblogs and Social Media, 2013.