Sentimental Analysis to Rank Web Products

Mr. Abhijit J. Borade[1] Mr. Imran R. Momin[2] Mr. Dattatray B. Ghogare[3] Prof. S. Pratap Singh[4]
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]Institute of Knowledge College of Engineering, Pune

Abstract— Nowadays, there is a trend of online shopping as more and more products are being sold by the manufacturers on the internet and customers are expressing their views and reviews about a product on the internet. Traditionally, individuals collect feedback from their friends or relatives before purchasing an item, but nowadays user generated reviews give the information which helps customers in buying their favorable products. So reviews of huge number of individuals around the globe can be analyzed. For that, various review mining techniques and sentiment analysis techniques are currently being used and products are being ranked using various ranking algorithms. We present a product ranking system using review mining techniques. Users can specify product information and get the ranking results of all matched products in return. We consider three issues while calculating product scores: Product reviews, Product popularity and Product release month. We are going to rank the products specified by user not all the products.

Key words: Natural Language Processing, Review Mining, Sentiment Analysis, POS Tagging, Wordnet

I. INTRODUCTION

With the exponential growth of the web there has been explosive increase in the user generated contents in the form of customer reviews, their blogs, discussions, social networks etc. The contents are stored as unstructured or semi-structured data from where distillation of knowledge is a challenging task. Till now, most of the ranking systems are doing reviews and sentiments analysis to decide whether it is positive, negative or neutral [1], [2], [3]. But this is not sufficient to rank. To rank the products we should have some numeric score for each product.

Here we propose a feature wise review mining system which first extracts features from user reviews, then finds the intensity by giving emphasis to the modifier of the words expressing reviews. It finds out numeric score of all the features and sentiments then calculates the overall orientation of the feature to determine how intense the review is for positive and negative features. Identify positive and negative features by extracting the associated modifiers and reviews. Specify the features in descending order of importance to present the summery.

We consider three issues while calculating product scores: 1) Product reviews, 2) Product popularity and 3) Product release month. We are going to rank the products specified by user not all the products. Users will specify product information and get the ranking results of all matched products in return.

Organization of the paper is as following. In the next section i.e. Section II; we review basic concepts and algorithms used in this paper in brief. Then, the framework of a product ranking system is proposed in Section III. Section IV contains the implementation screen samples.

II. BASIC CONCEPTS AND ALGORITHMS

Here in this section we are going to review the basic concepts briefly.

A. POS Tagging:

The POS tagging generally referred to as Part-of-speech tagging [1] (i.e. POST), is also called as word category disambiguation or grammatical tagging, and it is basically the process of giving the correct part of speeches which can be verb, adverb, adjective, noun, pronoun, etc. to each word in a data that is textual and which is based on both of its context and dentition as follows:

Word	POST	Meaning
This	DT	Determiner
Article	NN	Noun, Common, Singular or Mass
Is	VBZ	Verb, Present Tense, 3rd Person Singular
About	RB	Adverb
The	DT	Determiner
Sport	NN	Noun, Common, Singular or Mass
.	SENT	Sentences

Table 1: Part of Speech Tagging

B. WordNet:

WordNet [13], [14] is basically a database which is lexical for the English language [4]. It collects words in English sentences and groups them into collection of synonyms which is basically called as synsets, and gives us general and short concepts and definitions, and records the different relations which are semantic between these synsets.

C. Stopword Removal Algorithm [17]:

```
Algorithm Stopwords Removal  (wordsList [],
stopWordsofwordnet[])
{
    Define Stopwords set i.e. Stopwords word net.
    for i := 0 to wordsList.size do
    {
        If ( BinarySearch ( wordsList[i],
stopWordsofwordnet ) ≠ -1 ) then
            Remove wordsList[i] from wordsList
    }
}
```

Fig. 1: Stopword Removal Algorithm

Each word in the input is scanned. For each word encountered searching is done to check whether the word is in the WordNet stopwords' list or not. If it is there then remove it. A sample set L of stopwords is given bellow.
L = { am, is, are, the, a, an, have, has, had, do, does, did }

D. Stemming Algorithm:

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words

"fishing", "fished", and "fisher" to the word, "fish". Another words, "argue", "argued", "argues" and "arguing" reduce to the stem "argue" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument". [15]

Example: Let us consider the sentence given bellow.

"Product worked absolutely fine."

Here it will encounter two words with some suffix. Now these suffix parts are removed to form a word without suffix.

Worked, absolutely
↓
Work, absolute

E. TF-IDF:

TF-IDF stands for Term Frequency-Inverse Document Frequency, and its weight is a weight used for retriving information and text mining. It is a statistical measure which is used to evaluate how much important a word is to a document in a collection. The importance increases as the number of times a word appears in the document but is offset by the frequency of the word in the collection. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

How to Compute? Typically, the TF-IDF weight is composed by two terms: the first computes the normalized Term Frequency (TF). We are not going to use TF. The second term which we are going to use is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

IDF: Inverse Document Frequency, which measures how important the term is. When computing Term Frequency, all terms are considered as they are equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. SO we will weigh down the frequent terms while scale up the rare terms, by computing the following [1]:

$$IDF(t) = \ln\left[\frac{Total\ No.of\ Documents}{No.of\ Documents\ with\ Term\ t\ in\ it}\right]$$

Example: Consider a document containing 100 words wherein the word cat appears 4 times. The term frequency (i.e., TF) for cat is then (4 / 100) = 0.04. Now, assume we have some million of documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., IDF) is calculated as ln(10,000,000/1,000)=4. Thus, the TF-IDF weight is the product of these quantities: 0.04 * 4 = 0.16. [16]

III. SYSTEM FRAMEWORK

In general, ranking is provided for different types of products (e.g. computers, mobiles), based on the analysis of collected user reviews. Therefore, our system is designed to provide relevant product ranking for different products. First, through the Internet, downloaded HTML page data from Amazon product review pages is to be given to the system.

Then, it would extract the required product reviews and information from downloaded data, and split them into sentences. Next, it identifies the polarity of the review words in each sentence. Finally, the product information and review polarity would be integrated. Afterwards, product scores are generated and ranking results are obtained.

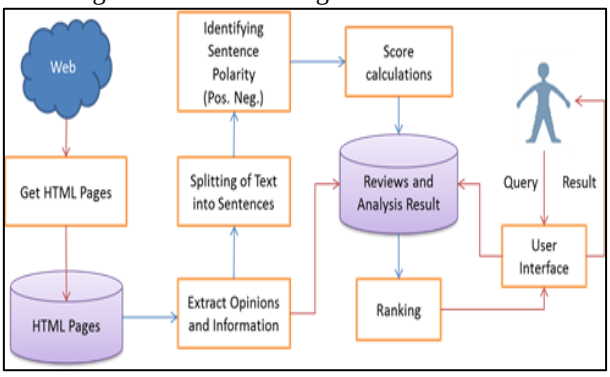

Fig. 2: System Framework

A. Download Product Web Pages:

When user searches for product we download all matched products' web pages. These pages contain information and reviews of each product. For example a user searches for Nokia Lumia phones. The search results will contain Nokia Lumia mobile phones of various models, prices, colors, etc. Web pages for all these mobile phones will be downloaded.

B. Reviews and Information Extraction:

This is to identify relevant portion of the web documents. We use Markup Language (ML) tag filter which divides the individual documents in individual record size chunks and presents them as individual unstructured record documents for further processing. The cleaned document is then converted into numeric-vectors using unigram model for the purpose of subjectivity/objectivity analysis. In document vectors a value represents the likelihood of each word being in a subjective or objective sentence [3].

Subjective sentences are expressive of the reviewer's sentiments about the product and objective sentences do not have any direct or obvious bearing on or support of that sentiment. Therefore, the idea of subjectivity analysis is used to retain segments (sentences) of a review that are more subjective in nature and filter out those that are more objective.

In this way we will extract name, brand, release date, price, number of reviews, reviews, etc. Now for each review we will find out how many people found that review helpful, rating, review post date, etc.

C. Splitting Review Text into Sentences:

We parse a product's reviews to split texts into sentences, and produce POS tags like noun, verb, adverb, adjective, etc. for each word [1].

D. Identify Sentence Polarity (Pos./Neg.):

Polarity of a sentence can be calculated by using review strength (OS_p), IDF_p and degree of adverbs ($Degree_p$) that can modify polarity [1].

$$Sentence_p = OS_p \times IDF_p \times Degree_p$$

1) Review Strength:

$$OS_p = Sign\left(SET\ (p)\right)\frac{|CS(p)|}{|SET(p)|}$$

Where SET(p) is positive or negative set based on polarity of adjective p. CS(p) is closed set which is extended by p using synonyms and OSp is review strength of adjective p which is between -1 and 1. If we take an adjective "good" as an example, it's polarity is positive. The positive set consists of 4308 elements; CS(good) is extended by "good" and has 3879 elements. Then, OSgood can be calculated as 3879 / 4308 = 0.900418.

2) Inverse Document Frequency:

The formulae given above in section II (E) will calculate the IDF value but here we are identifying polarity. So we need to normalize IDF such a way that it will give us a value either 0 or 1.

$$IDF_p = \ln\left(\frac{R}{RCA_p}\right) \times \gamma, \quad \gamma = \frac{1}{\ln(R)}$$

Where RCAp denotes number of product reviews containing adjective p, R is the number of all product reviews, and γ is a normalization formula making IDFp value in the range of [0, 1].

3) Degree of Adverbs:

Adverb can modify adjective (Adverb + Adjective) and enhance or weaken the adjective strength or even change the polarity (not + Adjective). The Degreep denotes the degree of an adverb modifying an adjective p. Four levels of adverbs are shown in table 2 bellow. Where the weights of high (0.6), medium (0.5), low (0.4), and negative levels (-1); if no adverb is used to modify an adjective, the weight would be 0.5. [1]

The polarity is saved into the database along with the extracted information.

High (0.6)	**Medium (0.5)**	**Low (0.4)**	**Negative (-1)**
very, incredibly, much, so, too, completely, etc.	fairly, pretty, rather, as, almost, partly, half, etc.	slightly, a little, a bit, somewhat, etc.	Not

Table 2: Degree of Adverbs

E. Calculate Final Score:

To calculate the score we need Average Polarity of Reviews (APR), Popularity Weight (PW) and Weight of Product Release Month (WPRM).

$$Score_i = APR_i \times PW_i \times WPRM_i$$

1) Average Polarity of Reviews APR could be calculated as follows.

$$APR_i = \frac{\sum_{j=1}^{n}\left[\left(Polarity_j \times HF_j\right) \times WRPD_j\right]}{n}$$

Where n is number of review, polarity is polarity of review, HF is a value between -0.99 to 0.99, WRPD is the weight of review post date.

1) Polarity of a Review Is Calculated As:

$$Polarity_j = \frac{\sum_{p=1}^{k} Sentence_p}{k}$$

2) Helpful (HF) Can Be Calculated As:

$$HF_j = \frac{\alpha \times [\ln(\left|Help_j - NotHelp_j\right| + 2)]}{10}$$

$$\begin{cases} -1, if\ 1)\ Polarity_j > 0\ and\ Help_j - NotHelp_j < 0 \\ \quad 2)\ Polarity_j < 0\ and\ Help_j - NotHelp_j > 0 \\ 1, if\ 1)\ Polarity_j > 0\ and\ Help_j - NotHelp_j > 0 \\ \quad 2)\ Polarity_j < 0\ and\ Help_j - NotHelp_j < 0 \end{cases}$$

Where Helpj is number of users marked the review helpful and NotHelpj is the number of users felt that the review is not helpful. HF value is in the range -0.99 and 0.99.

3) WRPD Has To Be Calculated As:

$$WRPD_j = \exp\left(\frac{RPD_j - t}{30 \times \beta}\right),$$

$$\beta = \left\lceil \frac{t - \min(RPD_j)}{30} \right\rceil$$

Where RPDj is the post date of review j, t is the current date, and β normalizes WRPD value in the range (0.36, 0.99). [1]

2) Popularity Weight:

Popularity Weight is how much a product is popular. A product with more discussion is more popular.

$$PW_i = \frac{\ln(m_i + 1)}{\ln(\max(m) + 1)}$$

Where the number of reviews for product i are mi, and max(m) is the maximum number of reviews among all products.

3) Weight of Product Release Month

WPRM is calculated in the similar way we calculated WRPD which is calculated as following:

$$WPRM_i = \exp\left(\frac{PRM_i - t}{12 \times \beta}\right), \quad \beta = \left\lceil \frac{t - \min(PRM_i)}{12} \right\rceil$$

Where PRMi is release month of product i, t is the current month, and β normalizes WPRM value.

IV. SOME IMPLEMENTATION SCREEN SAMPLES

After a successful file upload the file is parsed to get the review section and other product features such as its name, the date when it was released, how many reviews it has, etc.

In the screen sample 1 bellow the reviews are shown there in the iframe with its score, the average polarity and opinion strength. All extracted reviews are displayed in the iframe. You need to scroll down to the bottom to see the score. The file upload can be done by admin only. Normal user do not have that option.

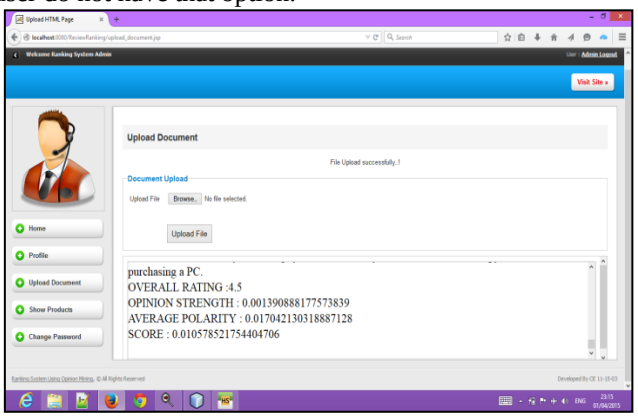

Fig. 3: Screen Sample 1

The screen sample 2 bellow shows the actual ranking page. User has to select a product category to get ranking of that category.

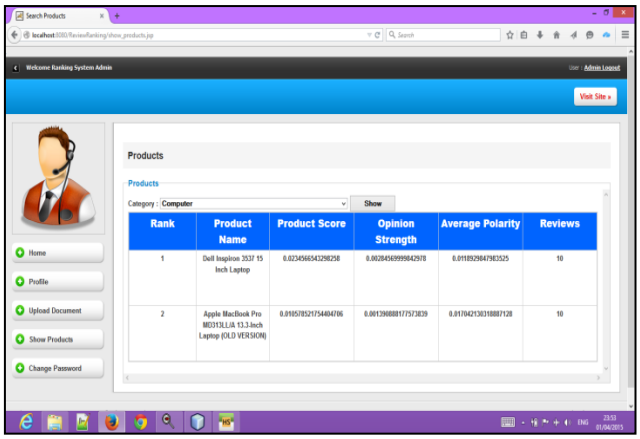

Fig. 4: Screen Sample 2

V. ACKNOWLEDGMENT

We thank Dr. Damodar J. Garkal (Principal IOKCOE Pune) for providing necessary facilities to carry out the work. We are very thankful to Prof. S. Pratap Singh (Assistant Professor and Dean of Student Welfare IOKCOE Pune) and our Head of the Department Prof. Ritesh Thakur for their useful guidance.

REFERENCES

[1] Yin-Fu Huang, Heng Lin, "Web Product Ranking Using Opinion Mining", IEEE 2013 Symposium on Computational Intelligence and Data Mining (CIDM), Pages 184-190
[2] Jalaj S.Modha, Prof & Head Gayatri S. Pandi, Sandip J. Modha, "Automatic Sentiment Analysis for Unstructured Data", International Journal of Advanced Research in Computer Science and Software Engineering December, Volume 3, Issue 12, December 2013, Pages 91-97
[3] Tanvir Ahmad , Mohammad Najmud Doja, "Ranking System for Opinion Mining of Features from opinion Documents", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012, Pages 440-447
[4] Keke Cai, Scott Spangler, Ying Chen, Li Zang, "Leveraging Sentiment Analysis for Topic Detection" IEEE 2008
[5] Balakrishnan Gokulakrishnan, Pavalanathan Priyanthan, Thiruchittam-palamRagavan, Nadarajah Prasath, AShe han Perera, "Opinion Mining and sentiment analysis on a twitter data", The International Conference on Advances in ICT for Emerging Regions – ICTER Presentation, 2012
[6] Ralf Krestel, Nima Dokoohaki, "Diversifying Product Review Rankings: Getting the full picture", IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2011, Vol 1, Page 138-145
[7] Thomas Y. Lee, Simon Li, RanWei, "Needs-Centric searching and ranking based on customer", 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008
[8] Hady W. Lauw, Ee-Peng Lim, "A Multitude of Opinions: Mining Online Rating Data", National Science Foundation Symposium, October 2007
[9] Edison Marrese-Taylor, Juan D. Vel´asquez, Felipe Bravo-Marquez, Yutaka Matsuo, "Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach", 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems – KES 2013, Pages 182-191
[10] Rudy Prabowo, Mike Thelwall, "Sentiment Analysis: A Combined Approach", Journal of Informetrics, Vol 3 No 2, 2009, Pages 143-157
[11] G.Vinodhini, RM.Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, June 2012, Volume 2, Issue 6
[12] Dongjoo Lee, Ok-Ran Jeong, Sang-goo Lee, "Opinion Mining of Customer Feedback Data on the Web", International Conference on Ubiquitous Information Management & Communication (2008), Pages 230-235
[13] WordNet: (en.wikipedia.org/wiki/WordNet/)
[14] WordNets in the world : (http://www.tfidf.com/)
[15] Stemming: (en.wikipedia.org/wiki/Stemming/)
[16] TF-IDF: (en.wikipedia.org/wiki/Tf–idf/)
[17] StopwordsRemoval:(http://stackoverflow.com/questions/25575152/remove-stop-words-from-array-and-return-array-in-java)

