Accurate Recommendation Based on Opinion Mining

Xiu Li, Huimin Wang, and Xinwei Yan*

Tsinghua University, Department of Automation, Lishuistr. 2279, Shenzhen, China

Abstract. Current recommender systems are mainly based on customers' personal information and online behavior. We find that those systems lack efficiency and accuracy. At the same time, we observe the large amount of review data with exponential growth. Based on this observation, we propose a recommender system based on opinion mining. With text mining method we extract the opinion related information from the massive reviews. We analyse the linguistic information and design a two-layer selection algorithm to find the most suitable products for customers. The experiment shows our method has great accuracy, fleasibility, and reliablity.

Keywords: e-business, opinion mining, recommender system.

1 Introduction

The Internet Era brings convenient information service, promotes the rapid development of E-commerce, and also has a profound impact on people's way of life. Both information acquisition and shopping consumption have generally turned to the online. China has the largest online market in the world, but consumers are always limited by cognitive ability and the immature information search behavior confronted with the complicated online market. As one of the most appropriate ways, accurate recommendation can help customers to quickly find the products they need. Opinion mining has great potential to be used in personal recommender system to significantly improve: i) accuracy by analyzing individual review data, and ii) reliability by considering numerous opinions of customers. However, designing a recommender system based on opinion mining is quite difficult, which should:

- Correctly recognize the sentiment expression in complicated Chinese text, extract feature words, adjectives and adverbs, and match them up.
- Accurately calculate quantified sentiment strength of semantic features, and output a normalized numerical value in certain dimensions.
- Appropriately recommend products according to customers' review information.
- Obtain good generalization ability, high computational efficiency.

To the best of our knowledge, no prior system can satisfy all these goals simultaneously.

* Corresponding author.

H. Sun et al. (eds.), *Genetic and Evolutionary Computing*,
Advances in Intelligent Systems and Computing 329, DOI: 10.1007/978-3-319-12286-1_41

In this paper, we propose a new recommender system based on opinion mining. Firstly we establish a customer comments feature library and extract the semantic features by exploring the linguistic information. Next, we put up with a new method on fine-grained sentiment analysis to summarize opinion mining and statistics. Finally, indicators are generated and products to be recommended are selected by two-layer selection. The experimental results show that our system efficiently improves the accuracy and practicability of recommendation.

2 Related Works

The current recommender system of E-commerce shopping websites in China mainly relies on the purchase history, which can be divided into three categories [1]:

1* Collaborative filtering [2]. It explores the adjacent customers first and recommends what they love to the target customer. It dose not consider the product attributes which are pivotal in the system.

2* Content-based Recommendation [3,4]. It extracts the product features and generates feature vectors. Then it views customer shopping history and recommends similar product they had bought based on distances between feature vectors. It dose not consider customers' attitude towards the product, which limits the accuracy of recommendation.

3* Knowledge-based Recommendation [5]. It requires customers propose the demand first and the whole process is strongly interactive. Obviously, it can't obtain new customers initiatively and it's time consuming.

To solve the problems, more and more scholars began to research in E-commerce recommender system from various perspectives. Some of them are from the perspective of the users' opinion mining. Such a system based on opinion mining can be divided into three parts: Sentiment expression recognition, sentiment analysis and recommendation algorithm.

The process of sentiment expression recognition is based on Chinese Natural Language Processing (NLP) tools which can achieve word segmentation, POS tagging, named entity recognition and anaphora resolution. To extract sentiment expressions, key words related a certain theme have to be identified. Although there are many methods that focus on Chinese key words extraction, almost none of them consider the character of topic relativity. Jianlin Zhang and Qianli Shen [6] apply Dunning's [7] possibility algorithm to identified credit related key words, but the process is too complicated and time-consuming. Wenhua Wang and Yanhui Zhu try to extract the product attribute words and opinion words with machine learning method.

Sentiment analysis aims at recognizing words, sentences or document's sentimental polarity. Benefiting from the development and maturity of the technology in natural language processing and machine learning, it becomes possible to widely employ sentiment analysis on Chinese texts. Existing studies on sentiment analysis are mainly focusing on the task of determining word-level and sentence-level. Yanlan

Zhu and Jing Min use the semantic similarity between two words in HowNet library to distinguish sentence polarity [8]. Zhenyu Wang and Zeheng Wu combine point mutual information with semantic similarity to improve performance [9]. Hongwei Wang and Lijuan Zheng apply machine learning to get the sentimental contribution degree of sentence [10]. These methods rely on the accuracy of library, but there is no library directed to E-commerce.

The studies on recommendation algorithm are mainly focus on the three methods mentioned above.

3 Methodology

Our system mainly covers two parts—reviews data reproducing and recommendation algorithm. With the use of Fudannlp, a famous open source tool in the field of Chinese language processing, we apply SVM to extract attribute words and their opinion words (Sec.3.1). Then we calculate the sentimental strength of sentiment expressions (Sec.3.2). After that, we select products through two layers (Sec.3.3). More specific process is shown in the flow chart as follow:

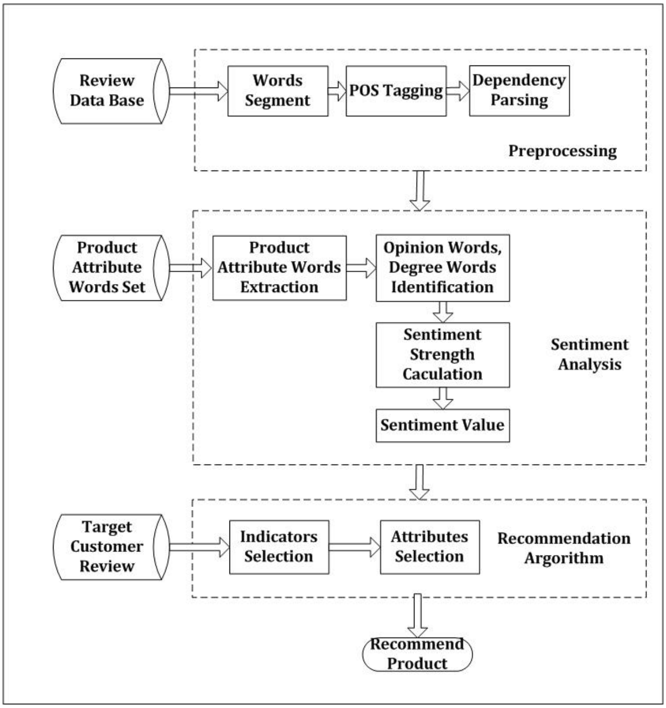

Fig. 1. The flowchart of the recommender system

3.1 Key Words Extraction

For every product attribute, the key words include attribute word, related opinion words, negative adverbs, and degree adverbs. The attribute word should be one of the features of the product. Recently, the accuracy and recall of attribute words extraction are relatively low [6]. To improve the performance, we refine the process and divide it into several steps:

Candidate Words Selection: Although Chinese sentences are complex and diverse, they show some statistic characteristics when limited to product reviews. We randomly crawled 4,000 customer reviews from Jingdong which is one of the chief online shopping malls in China. After text data preprocesses including Chinese word segment, POS tagging, and dependency parsing, we count 90% of attribute words are noun and 75% of them act as the subjects of the sentences. This is highly in accordance with the Chinese daily way of speaking. Therefore, we mainly extract the words with the POS tagging "noun" and parsing feature "subject" as candidate attribute words.

Product Attribute Words Extraction: Based on candidate attribute words, we go further more to choose to credit indicator related words as attribute words by matching the product attribute words set. More details, we establish word sets $W_1 \dots W_n$, and each of them contains the common used words of one aspect of product. To prune the candidate attribute words, we calculate every candidate's semantic distance with every element w in $W_1 \dots W_n$ using the method based on HowNet and normalize it as Equation (1), whose values are between 0 and 1.

$$Sem(w,s) = \frac{d(-\frac{1}{2}S(w,s)+1)}{S(w,s)+d} \tag{1}$$

Where $S(w, s)$ is the space distance of Space Vector Model between w and s, d is a controllable parameter, and s is a candidate attribute word. $if\, \exists i \in \{1,2,L,n\},\ s.t.\ Sem(s,Wi) \geq t$, where t is a threshold, then s is a chosen attribute word.

Opinion Words, Degree Adverbs, and Negative Adverbs Identifying: for every product attribute word, we identified its opinion words, negative adverbs, and degree adverbs with SVM. With *one versus rest*, one of the most widely used methods of SVM multi-classifier, we classify the words excluding the attribute words in one piece of review to opinion words, negative adverbs, degree adverbs and other words. For the convenience of description, we noted one attribute word's opinion words, degree adverbs, and negative adverbs as its related words. Furthermore, to facilitate classification, we quantify the relationship between one attribute word and its related words. Under the comprehensive analysis of the grammar features, the orders and the distances between attribute words and their related words, we make **feature selection rules** as follow:

1* Part of speech of the related words. According to the 39 kinds of outputs of POS Tagging with FudanNLP-1.6.1, we number the different outputs from 1 to 39.

2* Dependency parsing's result of the related words. There are 22 kinds of syntactic relationships of Dependency Parsing in FudanNLP-1.6.1, we number them from 1 to 22. E.g. we label the *Subject* to 1.

3* the distance between an attribute word and its related word. We calculate the number of characters excluding spaces between the attribute words and their related words.

4* the order of an attribute word and its related word. If the attribute word is preceded with its related word, we label this dimension of feature as 1, else 0.

5* is there a punctuation. We will label the feature as 0, if there is no punctuation between the attribute word and its related word. Else, if there is a period, label 1; if there is a comma, label 2; if there is a apace, label 3; else, label 4.

6* is there another attribute word. If there is another attribute word between the attribute word and its related word, we label this feature as 1, else as 0.

7* is there an opinion word. If the attribute word has opinion words, the dimension of feature will be labeled as 1; else, labeled as 0.

8* is there a degree adverb. If the attribute word has degree adverbs, the dimension of feature will be labeled as 1; else, labeled as 0.

9* is there a negative adverb. If the attribute word has negative adverbs, the dimension of feature will be labeled as 1; else, labeled as 0.

Training Classifiers. We choose the reviews from www.jd.com as training set after manually annotating. Specifically, we manually extract the product attribute words and identified their related words firstly, after that we will get the set of evaluation unit $G=\{(k1,d1,p1,o1), \ldots,(kn,dn,pn,on)\}$ for every piece of review, where n is the number of attribute words in the review, *ki, di, pi, oi* represent attribute words, degree adverb, negative adverb, and opinion word.. Then we quantify the relation features between attribute the attribute word and its related words. For example, there is an evaluation unit *(k,d,p,o)=(*布料, 非常, 不, 好*)* extracted from the review—这件衣服的布料非常不好, which means *the fabric of this dress is very bad*. The manual annotation is as follow:

Table 1. Manual annotation

Match or not	Attribute word	Related words	feature
1		*d*	2 4 0 1 0 0 0 0 0
1	*k*	*p*	3 5 4 1 0 0 0 1 0
1		*o*	3 6 6 1 0 0 0 1 1

3.2 Sentiment Analysis

We analyze every opinion word's semantic orientation with the method of word's similarity computing based on Chinese Thesaurus -*Tongyici Cilin*, which not only contains one word's synonyms, but also a certain number of its similar words. This dictionary is compiled in 1983, and has not been updated from then on. At the same time, the *Tongyici Cilin extension* keeps the original edition's three layer classification system, and adds two layers to be further sub-classes. With the final five layers classification, the words in the dictionary shows the good hierarchical relationships, which can be expressed as shown below, L1 means the first layer and the remains are in the same way.

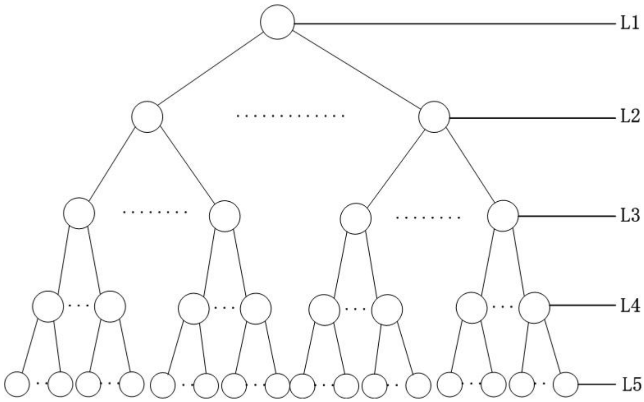

Fig. 2. Hierarchical relationships in *Tongyici Cilin extension*

Now the first layer will be C, the second will be b, the third will be 02 the fourth will be A, and the fifth will be 01. We definite *S (w1, w2)* as the similarity between *w1* and *w2* and its value belong to 0 to 1. The bigger the value is, the higher the similarity is. Let n represents the number of the layers, and the k is the distance of the two branches. The value of *S (w1, w2)* can be discussed in the following circumstances:

If *w1* and *w2* are not in the same search tree:

$$S\,(w1,\,w2) = f$$

If *w1* and *w2* are both in the 2nd layer of a same search tree:

$$S(w1, w2) = a\frac{n - k + 1}{n}\cos(\frac{n\pi}{180})$$

If *w1* and *w2* are both in the 3rd layer of a same search tree:

$$S(w1, w2) = b\frac{n - k + 1}{n}\cos(\frac{n\pi}{180})$$

If *w1* and *w2* are both in the 4th layer of a same search tree:

$$S(w1, w2) = c\frac{n - k + 1}{n}\cos(\frac{n\pi}{180})$$

If *w1* and *w2* are both in the 5th layer of a same search tree:

$$S(w1, w2) = d\frac{n - k + 1}{n}\cos(\frac{n\pi}{180})$$

Parameter *a, b, c, d, e, f* will be determined from the experimental result. One word may have several code, we choose the biggest value of the similarity. For example, the codes of word “骄傲（pride）” are “Da13A01” and “Ee34D01”. And the codes of the word “仔细（careful）” are “Ee26A01” and “Ee28A01”. There will 4 values of the similarity between the two words, there are 0.1, 0.1, 0.483920, and 0.51007, but we final choose the biggest one 0.51007. What is more, we get the similarity between word “美丽（beautiful）” and other words in Tab.2:

Table 2. The similarity between word “美丽（beautiful）” and other words

Word	Similarity
漂亮(beautiful)	1.000000
丑陋(ugly)	0.161666
可爱(cute)	0.582177
灿烂(splendid)	0.478922

For one evaluation unit *(k, d, p, o),* we get the sentiment of the opinion word *o* according to Formula (2) as follow:

$$Ori(o) = w_n(p) \cdot w_{\mathrm{d}}(d) \cdot \left(\frac{\sum_{i=1}^{n} S(wp_i, o)}{n} - \frac{\sum_{j=1}^{m} S(wn_j, o)}{m}\right) \tag{2}$$

Where wp_i and wn_j separately are basic words of positive and negative word set, n and m separately are the number of basic words of positive and negative word set. W_n is the weight of the negative adverb, and W_d is of the degree adverb. If there is no negative adverb n, W_n is equal to 1. Else W_n can be obtained by Formula (3):

$$W_n = \max_{i \le m}(S(wn_i, n)) \tag{3}$$

Where wn_i is the word of negative adverb set, and m is the number of the words in the set.

If there is no degree adverb n, W_d is equal to 1. Else W_d can be obtained by Formula (4):

$$W_d = 1.1 \max_{i \le m}(S(wdi, d)) \tag{4}$$

Where *wdi* is the word of highest degree adverb set, and n is the number of the number of the words in the set. In fact, degree word can be classified to three types — high, intermediate, and low. We mainly build the highest degree adverb set, with which degree adverb d is compared.

Since the value of w_n, w_{d} and $S(w1, w2)$ are belong to [0, 1], the value of the sentimental orientation of opinion words should range from -1 to +1, and the bigger the value is, the stronger the sentimental orientation is and the higher the score of the indicator corresponding to the attribute word is.

3.3 Recommendation Algorithm

So far we have obtained the subjective feedback from customers. Next, we propose a two layers matching algorithm. We establish an indicator system as the upper layer with clustering method. Then in the sublayer, for every single review we select attributes with high customer scores as feature attributes. After that, we search for products through these two layers according to customers' opinion.

3.3.1 Attributes Clustering

Customer review scores can reflect the quality of products, but the amount of attribute words is huge and dimensionality reduction should be done. In order to simplify the search complexity, similar attributes should be combined into an indicator and the number of indicators should be appropriate. We use K-means clustering method to obtain indicators from attribute words. We try *1* to *25* as K value which represents the number of class centers. It shows that when K=*12*, the classes have best performance on the within class scatter and the between class scatter.

So we select one word that summarizes the class as the indicator and set the indicator system with *12* indicators which are shown below.

Indicators: *Quality, Performance, Function, Price, Advertising authenticity, Online service, After-sale, Logistics service, Transaction security, Transaction convenience, Transaction frequency, Transaction value.*

Every review can be quantized as a vector with 12 dimensions: $R=\left(r_1, r_2 \cdots r_{12}\right)$. And every product can be quantized as: $\overline{R}=\left(\overline{r}_1, \overline{r}_2 \cdots \overline{r}_{12}\right)$, where $\overline{R}$ is the mean of all reviews of the product.

3.3.2 Attributes Selection

As a product to be recommended, we select its advantageous attributes as feature attributes and use them to represent the product. Feature attributes should be satisfied with following conditions:

1* High customer score. The scores of feature attributes should be higher than a threshold σ_s which can be set manually.

2* Enough review rate. There should be enough customers in favour of it when setting a feature attributes. Attributes with review rate under σ_r are not in the area of concern.

For every product, we establish a set of attributes $A=\left\{a_1, a_2 \cdots a_n\right\}$ which can be summarized as an indicator set $I=\left\{i_1, i_2 \cdots i_n\right\}(\mathrm{n} \leq 12)$.

3.3.3 Product Matching

We assume that customers express what they care about by comments. When a customer writes a review which contains several attributes $A'=\left\{a'_1, a'_2 \cdots a'_n\right\}$ in certain indicators $I' = \{i'_1, i'_2 \cdots i'_n\}$ (n $\leq$ 12). We first search for the same kinds of products with the same indicator set, which means $I = I'$. After that we search for products with $A' \subseteq A$ in the candidate products. The more details the customer review contains, the more accurate the recommendation will be.

3.4 Experimental Analysis

Text Mining Performance Test. In the text mining, if the related word is matched, we label 1 as matching information, if not, label 0. Then we get rid of the text information and set the matching information and quantified feature characters as input to train the classifiers. We randomly choose 70% of the manually annotated data, around 700 pieces of reviews, as training set, the remains as testing set. To evaluate the performance of the classifier, we use the test data and calculate the accuracy, recall, and F1-means, and the result is shown in Tab. 3.

Table 3. Evaluation-unit extracting result

Precision	Recall	F1-means
85.05%	80.47%	82.69%

It is obvious that text mining performance is improved significantly.

Recommendation Performance Test. Due to the particularity of the system mechanism, it is difficult to directly test its performance. We assume that when customers review with sale slips, they are satisfied with the product. We randomly select 100 reviews with slips and compared their products and products our system recommended, the result is shown in Tab.4.

Table 4. Evaluation-unit extracting result

Same brand	Other brands with close scores	Other brands with far different scores
34%	61%	5%

It shows that 95% of customers get effective recommendation.

4 Conclusions and Future Work

We present an efficient but surprisingly simple e-business recommender model based on opinion mining. Text mining performance is improved significantly and the refined data provides a good foundation of other related researches. We design every detail with actual mechanism and in the whole process of modeling, we maintain a good objectivity.

Limitations. The theme related key words' extracting depends on word sets, which affects the portability of the model. The recommendation algorithm may be combined with existing algorithms to improve the performance one step further.

Future Works. Deep learning combined by POS and parsing character can be used in theme related words extracting to improve its efficiency.

References

[1] Adomavicius, G., Tuzhilin, A.: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)

[2] Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital libraries, DL 2000, New York, pp. 195–204 (2000)

[3] Zhou, J., Luo, T.: Towards an Introduction to Collaborative Filtering. In: Computational Science and Engineering, pp. 576–581. IEEE, Los Alamtios (2009)

[4] Si, L., Jin, R.: Unified Filtering by Combining Collaborative Filtering and Contend-Based Filtering via Mixture Model and Exponential Model. In: Knowledge Manage, pp. 156–157. ACM, New York (2004)

[5] Jannach, D., Zanker, M., Felfering, A., et al.: Recommender Systems: An Introduction. Cambridge University Press (2011)

[6] Zhang, J.L., Shen, Q.L., Wu, J.Y.: E-Businessmen credibility mining based on web reviews. In: Proceedings of 2nd International Conference on E-Business and E-Government, pp. 5955–5961. ICEE, Shanghai (2011)

[7] Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1993)

[8] Zhu, Y.L., Min, J., Zhou, Y.Q., et al.: Semantic Orientation Computing Based on HowNet. Journal of Chinese Information Processing 20(1), 14–20 (2006)

[9] Wang, Z.Y., Wu, Z.H., Hu, F.T.: Words Sentiment Polarity Calculation Based on HowNet and PMI. Computer Engineering 38(15), 187–189 (2012)

[10] Wang, H.W., Zheng, L.J., Yin, P., et al.: Chinese Web Reviews Sentiment Polarity Classification Based on Sentence Level Sentiment. Journal of Management Sciences in China 16(9), 64–74 (2013)