

Mining Feature-Opinion from Reviews Based on Dependency Parsing

Rui Hao, Tieke He, Hang Qi, Jia Liu*

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

*liujia@nju.edu.cn

Abstract

Manually reading all the product reviews to find a satisfying item is not only labor-intensive, but also tedious for the consumers. In this paper, we propose a feature-opinion mining approach to automatically summarize the reviews. Specifically, in our approach we first utilize a regression model to generate sentiment word, including phrase and its sentiment weight, then extract feature based on the dependency relationship between feature word and sentiment word, and finally we assign score to feature according to the dependency relationship. The experimental results demonstrate that our approach can effectively mine the feature-opinion from reviews.

Keywords: E-commerce, opinion analysis, dependency parsing

1. Introduction

In the information age, electronic commerce has been widely accepted by the public. According to the Economic Daily¹, there have been more than 0.3 billion netizens and 83 million online markets contributing to electronic commerce in China.

However, there is a serious problem in e-commerce, it is the intermingling of good, mediocre and bad products. Consumers usually judge whether a product is good or not by its online reviews, but too many reviews bring them trouble in choosing a suitable product within 10 minutes [1, 7], even though they clearly know what they want.

Facing enormous comments, people are desperate for a tool to scan and summary the comments automatically. Researchers have proposed many methods [2, 6, 9] for extracting product virtues and defects. In this paper, we propose feature-opinion mining approach, which is based on dependency parsing. In summary, there are two main contributions of our work:

¹http://paper.ce.cn/jjrb/html/2014-05/31/content_202681.htm

Feature extraction: Previous works usually identify features according to the lexical categories of the word, for example, M. Hu [6] used the CBA association rule miner [8] to pick out nouns or noun phrases in the review as possible features, but some non-noun features would be omitted in their method. In this paper we extract feature in a new way that we regard the word or phrase which is described or modified most frequently by sentiment words as feature candidates. The main idea behind this approach is that when people talk about feature, they usually use some opinion words on the feature, in other words, if a word or phrase is related to many opinion words, it is probably a feature.

Feature scoring: In the work of C. Scaffidi, et al [9], they assigned the semantic orientation of the whole sentence to features in the sentences as their orientation, which is not suitable for our work, as the reviewer may give both terrible and fine features in a single sentence. In that case, terrible features should be assigned negative orientation while fine features should be assigned positive orientation. Therefore in this article we compute semantic score for each feature in a sentence. What's more, we compute the score using the orientation sum of all opinion words which have relationship with the feature. Different from previous works [3], which defined the relationship as text distance between feature and opinion word, our work prefer to use syntax dependency between feature and opinion word, which has been proved more accurate [10].

The rest of this paper is organized as follows. We first present our approach in detail in next section, then we evaluate our method in Section 3, and finally we conclude this paper and outline some future work in Section 4.

2. Approach

2.1 Sentiment Word Generation

We collected product information and consumer reviews in December 2015 from JingDong website², which is one of

²<http://www.jd.com>

the largest Chinese online shopping platforms. We crawled 1,909 cellphone products, 1,089 digital single lens reflex camera(DSLR) products and 3,160 tablet products, whose review count is 357,126, 53,980 and 308,574 respectively.

We utilized a linear regression model [5] to generate sentiment words, it models the relationship between review text and rating, and output weights for each word, which is crucial for us to calculate the score of features. We got the items with outstanding positive and negative weights as our sentiment words.

Instead of dumping data into model directly, we first segmented the review text using *jieba* tool³, and then we removed stop words, transformed the English words to lowercase, and finally filtered out the words whose frequency of occurrence is less than 1,000. After the pre-processing steps, a corpus, which would later be formalized into a $m \times n$ matrix X , was obtained. Here, m is the number of comments, and n is the size of corpus. We also need a vector Y with m items to stand the ratings. Then we trained the following regression model, X_{ij} means the frequency of the j -th word in the i -th comment, Y_i is the rating of the i -th comment.

$$L(W) = \sum_{i=1}^m (Y_i - (w_0 + \sum_{j=1}^n X_{ij}w_j))^2 + \gamma P(W) \quad (1)$$

$$P(W) = \sum_{i=1}^n ((1 - \alpha)w_i^2 + \alpha|w_i|) \quad (2)$$

Through the model above we could produce a set of parameters $W = w_0, w_1, \dots, w_n$ as the words' weight.

We used *glmnet* package in R [4] to do the linear analysis, and the best γ for category cellphone, DSLR, tablet is 0.00041, 0.00390 and 0.00061 respectively. The α did not need to be optimized and was set to 1.

Finally, we listed the words with the largest positive weights and negative weights to constitute our sentiment word set. The top 5 sentiment words for each category are listed in Figure 1.

2.2 Feature Extraction

We have obtained the sentiment words for each category in above-mentioned steps, and we will present how to extract features using those sentiment words in this section.

2.2.1 Dependency parsing

Before describing the feature extraction method, we first need to introduce the dependency parsing, which reveals the syntactic structure by analyzing the dependency relationship between different components of a sentence. That

³<https://github.com/fxsjy/jieba>

is, the dependency parsing labels grammatical constituents such as “subject-verb”, “verb-object” and analyzes the relations between them.

To do the parsing job, we utilized analysis service provided by HIT-SCIR⁴ in this article. A simplified example is illustrated in Figure 2. From the figure we can see that the head predicate of the sentence is “great”, the subject is “phone”, “so” is degree adverbial of “great” and “this” modifies “phone”. The whole list of annotation of dependency relationships used in this article is listed in the Table 1.

Table 1. Dependency relationship definition

Tag	Description	Example
HED	head	the core part of the sentence
SBV	subject-verb	I gave her a camera (I ← gave)
VOB	verb-object	I gave her a camera (gave → camera)
ATT	attribute	great camera (great ← camera)
ADV	adverbial	so great (so ← great)

2.2.2 Extracting feature

For each sentence in a review, we captured its syntactic structure based on dependency parsing, which has been mentioned above, and saved the parse result to a .xml file. Then we defined rules to extract feature candidates from the result files.

In this paper, we define five extraction rules that are listed in Table 2. s and f refers to sentiment word and feature candidate respectively, tmp means any word, S (or F) represents sentiment words (or feature candidates) which have been extracted before and ref stands for the dependency relationships.

After feature extraction, we applied feature combination to make the final feature set more accurate. We merged two features according to their similarity, whose calculating formula is listed in the following. f_i, f_j stands for different features, and $union(f_i, f_j)$ means the length of common parts in f_i and f_j , $max(f_i, f_j)$ means the max of f_i 's length and f_j 's length. For feature whose length was 2, 3, 4 and 5, we chose 1.0, 0.6, 0.5 and 0.6 as their threshold respectively.

$$sim(f_i, f_j) = \frac{union(f_i, f_j)}{max(f_i, f_j)} \quad (3)$$

Based on the frequency of occurrences, we list the top 20 hottest features of cellphone, DSLR and tablet in Figure 3. In particular, we take a descriptive word for each feature, which is written in Chinese, to express the main idea of it.

2.3 Feature Scoring

For a feature, we summed weights of all sentiment words that had dependency relations with it as its score. Assume

⁴<http://www.ltp-cloud.com>

	Cellphone				DSLR				Tablet			
	word	description	weight	frequency	word	description	weight	frequency	word	description	weight	frequency
Positive	差不多	fine	0.543	1132	不错	nice	0.168	21504	差不多	almost	0.502	1351
	国货	domestic	0.291	2002	行货	authentic	0.109	1405	不卡	smooth	0.482	2853
	放心	assured	0.217	1085	实用	practical	0.088	1440	正品	certified	0.276	11058
	玩游戏	games	0.201	1085	正品	certified	0.071	8915	价钱	price	0.27	4317
	不错	good	0.191	111010	耐心	patient	0.07	1010	一天	a day	0.267	2079
Negative	失望	disappointed	-0.505	1458	不	no	-0.14	3622	后悔	regret	-0.583	1360
	坏	broken	-0.453	3061	客服	custom-service	-0.139	2871	太差	poor	-0.53	1081
	垃圾	rubbish	-0.427	3455	货	product	-0.097	1017	垃圾	rubbish	-0.502	4561
	死机	crash	-0.419	1796	发票	receipt	-0.094	1143	不好	bad	-0.5	6353
	经常	frequent	-0.344	1342	京东	JingDong	-0.091	3488	一个月	a month	-0.495	1368

Figure 1. Top 5 sentiment words of cellphone, DSLR and tablet

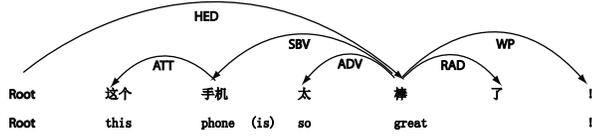


Figure 2. Dependency parsing example

that there is a review named “great cellphone”, from the dependency result we know that word “great” modifies word “cellphone”, and the orientation weight of “great”, which is calculated in Section 2.1, is supposed to be 0.5, so we say that the score of feature “cellphone” in review “great cellphone” is 0.5.

We obtained feature list for each product category in Section 2.2, then for each feature f on the list, we calculated its score on product p according to the equation:

$$score(f, p) = \frac{1}{n} \sum_{r \in R_p} \sum_{s_i \in S_r^f} weight(s_i) \quad (4)$$

R_p is the set of all reviews of product p , S_r^f is the set of sentiment words of feature f in review r , s_i and $weight(s_i)$ represents the sentiment word and its orientation weight, n means the size of set R_p .

Getting the set S_r^f is very important for calculating $score(f, p)$, and we also defined rules to detect sentiment word in a review, which has been listed in Table 2.

3. Evaluation

ZOL⁵ is a specialized information provider of digital products, it offers lots of information, including expert comments, customer comments, user forums and market news. In particular, ZOL extracts some features as standard features for each category manually and requires reviewers to rate them as long as they want to publish a review on the website. The consumers’ ratings are the capable criterion

⁵<http://www.zol.com.cn>

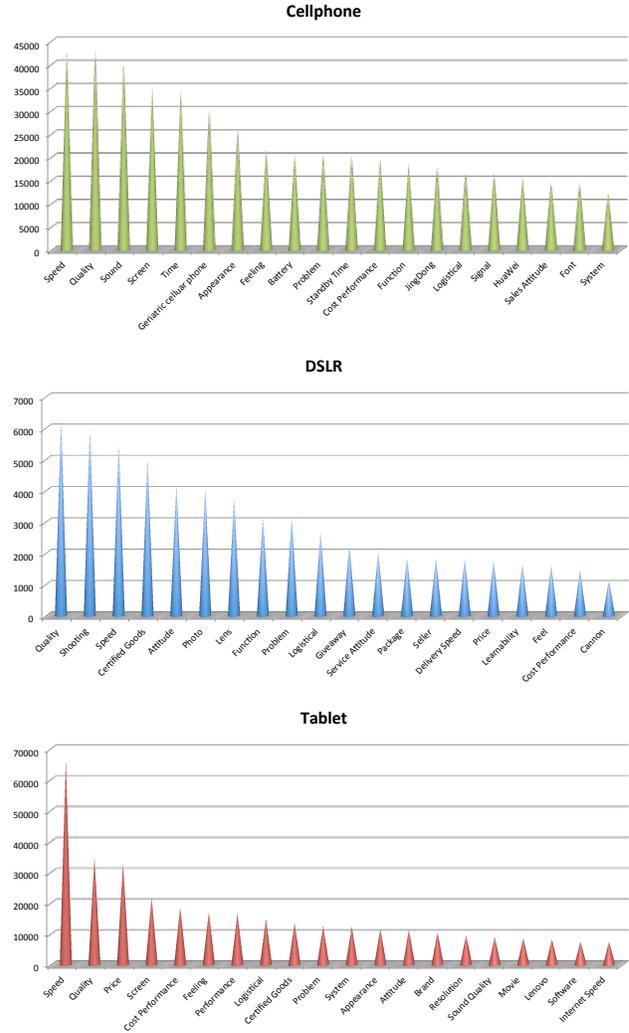


Figure 3. Top 20 hottest features

to evaluate our scoring result, so we crawled ratings for category cellphone, DSLR and tablet from ZOL in December 2015, and then matched product of ZOL with product of JingDong manually. After matching process, we obtained a date set with 120, 75 and 126 products for category cellphone, DSLR and tablet, respectively.

Table 2. Rules for feature candidate extraction

	Pattern	Constraint	Conclusion
R1	$f \rightarrow ref \rightarrow f'$	$ref \in \{ATT\}, f' \in F$	$f \in F$
R2	$f' \rightarrow ref \rightarrow f$	$ref \in \{ATT\}, f' \in F, f \notin S$	$f \in F$
R3	$f \rightarrow ref \rightarrow s$	$ref \in \{ADV, ATT\}, s \in S(f \in F)$	$f \in F(s \in S)$
R4	$s \rightarrow ref \rightarrow f$	$ref \in \{SBV, VOB, ATT\}(ref \in \{VOB, ATT\}), s \in S(f \in F)$	$f \in F(s \in S)$
R5	$f \leftarrow ref \leftarrow tmp \rightarrow ref' \rightarrow s$	$ref \in \{SBV\}, ref' \in \{VOB\}, s \in S(f \in F)$	$f \in F(s \in S)$

Then we matched standard features in ZOL with features we extracted before, but some standard features in ZOL are too general to match, for example there is a standard feature named “Entertainment”, but entertainment contains many aspects, such as movie, game, song, internet, etc. So we filtered out these features and got 5 cellphone features, 4 DSLR features and 5 tablet features as our criterion finally.

We utilized a generalized precision metric[9], which is listed in the following, to measure the difference between system score set S_f^c of category c and feature f with human score set H_f^c , n is the size of S_f^c , which is equal to the size of H_f^c . The evaluation result is illustrated in Figure 4.

$$precision(c, f) = 1 - \frac{1}{n} \sum_{i=1}^n \frac{|s_i - h_i|}{MAX(s_i, h_i)} \quad (5)$$

where $s_i \in S_f^c, h_i \in H_f^c$

4. Conclusion and Future Work

In this paper, we proposed a method to summarize user reviews based on dependency parsing. Our approach performed well during evaluation but there is still room for improvement. In our future work, we plan to add new feature extraction and feature scoring rules to refine the result, and we also intend to improve our feature combination using synonyms or semantic analysis.

References

[1] L. Barnard and J. L. Wesson. Usability issues for e-commerce in south africa: an empirical investigation. In *SAICSIT*, 2003.

[2] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

[3] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM*, pages 231–240, 2008.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *JSS*, 33(1):1, 2010.

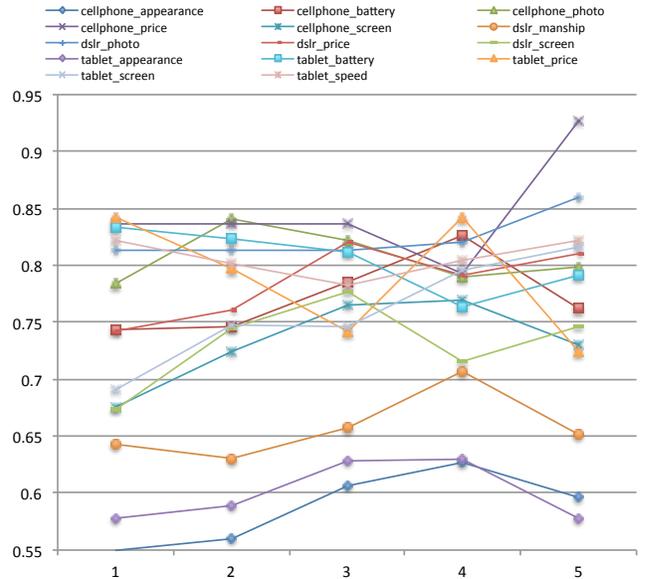


Figure 4. Feature scoring evaluation result

[5] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh. Why people hate your app: Making sense of user feedback in a mobile app store. In *KDD*, pages 1276–1284, 2013.

[6] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[7] R. Kohavi and R. Parekh. Ten supplementary analyses to improve e-commerce web sites. *Proceedings of the Fifth Webkdd Workshop*, 2003.

[8] B. L. W. H. Y. Ma. Integrating classification and association rule mining. In *KDD*, 1998.

[9] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin. Red opal: Product-feature scoring from reviews. In *EC*, pages 182–191, 2007.

[10] L. Zhuang, F. Jing, and X. Y. Zhu. Movie review mining and summarization. In *CIKM*, pages 43–50, 2006.